

Vakatseite zur korrekten Darstellung der 2-seitigen PDF-Ansicht.

Falls Sie die Arbeit ausdrucken, dann schließen Sie diese Seite
vom Druck aus.

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Betriebswissenschaften und Montagetechnik am
Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*)

Steigerung der Rekonfigurationsfähigkeit von Montageanlagen durch Cyber-physische Feldgeräte

Veit Georg Josef Hammerstingl

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen
Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. dr. ir. Daniel J. Rixen

Prüfer der Dissertation: 1. Prof. Dr.-Ing. Gunther Reinhart
2. Prof. Dr.-Ing. Rainer Müller

Die Dissertation wurde am 16.10.2019 bei der Technischen Universität München
eingereicht und durch die Fakultät für Maschinenwesen am 28.04.2020 angenommen.

Geleitwort der Herausgeber

Die Produktionstechnik ist für die Weiterentwicklung unserer Industriegesellschaft von zentraler Bedeutung, denn die Leistungsfähigkeit eines Industriebetriebes hängt entscheidend von den eingesetzten Produktionsmitteln, den angewandten Produktionsverfahren und der eingeführten Produktionsorganisation ab. Erst das optimale Zusammenspiel von Mensch, Organisation und Technik erlaubt es, alle Potentiale für den Unternehmenserfolg auszuschöpfen.

Um in dem Spannungsfeld Komplexität, Kosten, Zeit und Qualität bestehen zu können, müssen Produktionsstrukturen ständig neu überdacht und weiterentwickelt werden. Dabei ist es notwendig, die Komplexität von Produkten, Produktionsabläufen und -systemen einerseits zu verringern und andererseits besser zu beherrschen.

Ziel der Forschungsarbeiten des *iwb* ist die ständige Verbesserung von Produktentwicklungs- und Planungssystemen, von Herstellverfahren sowie von Produktionsanlagen. Betriebsorganisation, Produktions- und Arbeitsstrukturen sowie Systeme zur Auftragsabwicklung werden unter besonderer Berücksichtigung mitarbeiterorientierter Anforderungen entwickelt. Die dabei notwendige Steigerung des Automatisierungsgrades darf jedoch nicht zu einer Verfestigung arbeitsteiliger Strukturen führen. Fragen der optimalen Einbindung des Menschen in den Produktentstehungsprozess spielen deshalb eine sehr wichtige Rolle.

Die im Rahmen dieser Buchreihe erscheinenden Bände stammen thematisch aus den Forschungsbereichen des *iwb*. Diese reichen von der Entwicklung von Produktionssystemen über deren Planung bis hin zu den eingesetzten Technologien in den Bereichen Fertigung und Montage. Steuerung und Betrieb von Produktionssystemen, Qualitätssicherung, Verfügbarkeit und Autonomie sind Querschnittsthemen hierfür. In den *iwb* Forschungsberichten werden neue Ergebnisse und Erkenntnisse aus der praxisnahen Forschung des *iwb* veröffentlicht. Diese Buchreihe soll dazu beitragen, den Wissenstransfer zwischen dem Hochschulbereich und dem Anwender in der Praxis zu verbessern.

Gunther Reinhart

Michael Zäh

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) der Technischen Universität München (TUM).

Mein Dank gilt den Leitern des Institutes, Herrn Prof. Gunther Reinhart und Herrn Prof. Michael F. Zäh, für die Möglichkeit an dieser herausragenden Einrichtung promovieren zu können. Besonders bedanke ich mich bei Herrn Prof. Reinhart für die Betreuung und Unterstützung meiner Arbeit. Herrn Professor Rainer Müller (Universität des Saarlandes) danke ich sehr für die Übernahme des Korreferates und die aufmerksame Durchsicht der Arbeit.

Darüber hinaus bedanke ich mich bei all meinen Institutskollegen*innen, durch die ich viele neue Erfahrungen und Erkenntnisse gewinnen konnte. Euer Einsatz und Humor hat diese intensive Zeit zu etwas ganz Besonderem gemacht. Dies gilt besonders für Gregor Lux-Gruenberg, Marco Ulrich, Joachim Michniewicz, Cosima Stocker, Daria Leiber und Georgios Theodossiadis, die mich nicht nur mit großartigen Ideen befeuert, sondern mir auch zahlreiche unvergessliche Erinnerungen geschenkt haben.

Sehr herzlich bedanke ich mich bei meinem **AK@MI**-Studententeam. Ohne euren Einsatz wäre es nicht möglich gewesen, diese Vision auch in gegebener Zeit Realität werden zu lassen. Hierbei möchte ich das außerordentliche Engagement von Sarah Delgado, Bernhard Gerl, Anna Kollenda, Carina Micheler, Matthias Noller, Eva Paul, Ismael Prieto, Tobias Rupert und Patrick Zimmermann hervorheben. Besonders bedanke ich mich bei Markus Weber, der durch seine intergalaktischen Programmierkenntnisse mehr Probleme gelöst hat, als ich zählen kann. Ich wünsche euch für die weitere berufliche und wissenschaftliche Karriere alles Gute.

Weiterhin gilt ein großer Dank all meinen Freunden und Studienkollegen sowie meinen beiden Mentoren, Herrn Dr.-Ing. Bernd Spiegelberger und Herrn Dr.-Ing. Stefan Krug.

In ganz besonderer Weise bedanke ich mich bei Eva Katur, die mich stets unterstützt und motiviert hat, diese anspruchsvolle Aufgabe erfolgreich zu bewältigen.

München, im Dezember 2018

Veit Hammerstingl

Für meinen Vater, den wahrscheinlich größten Maschinenbauer der Welt

Inhaltsverzeichnis

Akronyme und Abkürzungen	V
Formelzeichen	IX
1 Einleitung	1
1.1 Vision der rekonfigurierbaren Produktionssysteme	2
1.2 Cyber-physische Systeme als Enabling Technology	3
1.3 Problemstellung und Betrachtungsraum der Arbeit	4
1.4 Forschungsmethodischer Rahmen und Aufbau der Arbeit	7
2 Grundlagen	9
2.1 Montagetechnik	9
2.1.1 Prozessschritte der Montage	10
2.1.2 Lebenszyklus von Montageanlagen	12
2.1.3 Anpassungsdimensionen und -befähiger	16
2.1.4 Komponenten automatisierter Montageanlagen	17
2.2 Automatisierungstechnik	18
2.2.1 Industrielle Kommunikation	20
2.2.2 Steuerungsgeräte	29
2.3 Cyber-physische Systeme	34
3 Stand der Erkenntnisse	37
3.1 Ansätze zur automatisierten Vernetzung von Betriebsmitteln	38
3.1.1 Methoden zur automatisierten Konfiguration des Bus-Masters	38
3.1.2 Dienstorientierte Architekturen und Middlewares	40
3.2 Digitale Modelle zur Erhöhung der Interoperabilität	43
3.2.1 Modellierung von Produkten	44
3.2.2 Modellierung von Prozessen	46
3.2.3 Modellierung von Betriebsmitteln	50

3.3	Ansätze zur aufgabenorientierten Programmierung von Montageanlagen	53
3.4	Ansätze zur automatisierten Ressourcenauswahl in Montageanlagen	61
3.5	Bewertung	65
4	Zielsetzung und Lösungsansatz	71
4.1	Handlungsbedarf	71
4.2	Zielsetzung	72
4.3	Vorstellung des Lösungsansatzes aus Anwendersicht	75
4.4	Methode zur Beschreibung der Softwarekonzepte	76
5	Automatisierte Vernetzung von Feldgeräten	77
5.1	Anforderungen	77
5.1.1	Referenzarchitektur	77
5.1.2	Anforderungen an das Konzept	80
5.2	Konzept zur automatisierten Vernetzung von Feldgeräten	81
5.2.1	Gesamtarchitektur	81
5.2.2	Architektur der Plug&Produce-Software	82
5.2.3	Ansteuerung von Geräten in Kommunikationshierarchien	87
5.2.4	Synchronisation des Digital Twins	88
5.2.5	Bestandteile des Ressourcentreibers	90
5.2.6	Besonderheiten in den Domänen Robotik und Bildverarbeitung	90
5.3	Hierarchie des Ressourcen-Informationsmodells	92
6	Fähigkeiten zur Prozessmodellierung	93
6.1	Anforderungen	93
6.2	Fähigkeiten in der Montage	95
6.2.1	Begriffsklärung	95
6.2.2	Vorgehensweise zur Herleitung	96
6.2.3	Elementare Fähigkeiten	96
6.2.4	Zusammengesetzte Fähigkeiten	99
6.2.5	Parameter von Fähigkeiten	101
6.3	Fähigkeiten im Ressourcen-Informationsmodell	108
7	Aufgabenorientierte Ressourcenauswahl und Programmierung	111
7.1	Anforderungen	111
7.1.1	Anforderungen an die Ressourcenauswahl	111
7.1.2	Anforderungen an die Programmierung	112

7.2	Merkmale im Ressourcen-Informationsmodell	113
7.3	Konzept zur aufgabenorientierten Ressourcenauswahl und Programmierung	115
7.3.1	Gesamtablauf	115
7.3.2	Architektur des TOMAP-Frameworks	116
7.3.3	Modelling Framework	119
7.3.4	Blackboard-System	120
7.3.5	Beschreibung ausgewählter Experten für die Montagetechnik	131
7.3.6	Codegenerator	142
8	Evaluation	145
8.1	Technische Bewertung	145
8.1.1	Demonstrator	145
8.1.2	Softwareimplementierung und Ausgangskonfiguration	146
8.1.3	Rekonfigurationsszenarien	155
8.1.4	Fazit anhand der Veränderungsbefähiger	158
8.2	Quantitative Bewertung	159
8.2.1	Probandenstudie	159
8.2.2	Wirtschaftlichkeitsrechnung	162
8.3	Weitere Potentiale der Lösung	165
9	Zusammenfassung	169
	Literatur	171
	Anhang	201

Akronyme und Abkürzungen

AoP	Aufgabenorientierte Programmierung
AP	Anforderung Programmierung
API	Application Programming Interface
AR	Anforderung Ressourcenauswahl
AT	Automatisierungstechnik
AV	Anforderung Vernetzung
BVS	Bildverarbeitungssystem
CAD	Computer-aided Design
CAEX	Computer Aided Engineering Exchange
COLLADA	COLLABorative Design Activity
CPPS	Cyber-physisches Produktionssystem
CPS	Cyber-physisches System
CPU	Central Processing Unit
DIN	Deutsches Institut für Normung
DOF	Degrees-of-Freedom
DT	Digital Twin
eng.	englisch
ETH	Ethernet
FB	Förderband
FDT	Field Device Tool
FTP	File Transfer Protocol
FU	Frequenzumrichter
gepl.	geplant
Getr	Getriebe
GSD	General Station Description
GUI	Graphical User Interface
HDD	Hard disk drive

HKKT	Hostspezifischer Kommunikationskanaltreiber (Plural: HKKTs)
HT	Host-Treiber
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IO	Input-Output
IODD	IO Device Description
IP	Internet Protocol
IPC	Industrie-PC
IRS	Industrierobotersteuerung
ISO	International Standards Organization
IT	Information Technology
KK	Kommunikationskanal
KKT	Kommunikationskanaltreiber (Plural: KKTs)
lat.	lateinisch
LED	Light-Emitting Diode
MAC	Media Access Control
MES	Manufacturing Execution System
MMI	Mensch-Maschine-Interaktion
Mot	Motor
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
MVG	Montagevorranggraph
OPC UA	OPC Unified Architecture
OSI	Open Systems Interconnection
OT	Operations Technology
OTE	Oberteil
P&P	Plug&Produce
p.a.	per annum
PC	Personal Computer
PDM	Produktdatenmanagement
PLM	Product Lifecycle Management
PPR	Produkt-Prozess-Ressource
PPRS	Produkt-Prozess-Ressource-Skill
PSP	Product-Shape-Pose

PSW	Pressschweißen
RAM	Random-access memory
RMS	Reconfigurable Manufacturing System
ROI	Region-of-Interest
ROS	Robot Operating System
RTE	Real-time Ethernet
SFC	Sequential Function Chart (Ablaufsprache)
SFG	Smartes Feldgerät (Plural: SFGs)
SOA	Service-orientierte Architektur
SPS	Speicherprogrammierbare Steuerung
SPSen	Speicherprogrammierbare Steuerungen
SSW	Schmelzschweißen
STEP	Standard for the exchange of product model data
Strg	Steuerung (z. B. SPS)
TCP	Transmission Control Protocol
UC	Use Case
UDP	User Datagram Protocol
ungepl.	ungeplant
USB	Universal Serial Bus
UTE	Unterteil
VDI	Verein Deutscher Ingenieure
WST	Werkstückträger

Formelzeichen

Bezeichner	Einheit	Beschreibung
Anz_h	–	Anzahl pro Stunde
$E_{\text{Stück}}$	€	Erlös pro Stück
K_{Betrieb}	€	Rechnerbetriebskosten
$K_{\text{InitialModellierung}}, K_{\text{IM}}$	€	Kosten der initialen Softwaremodellierung
$K_{\text{InitialProgrammierung}}, K_{\text{IP}}$	€	Kosten der initialen Softwareprogrammierung
$K_{\text{Invest}}, K_{\text{Inv}}$	€	Investkosten
$K_{\text{Komp.tausch}}$	€	Kosten des Komponententauschs
K_{Pers}	€	Personalkosten
K_{Rechner}	€	Kosten für den Rechnerbetrieb
K_{Rekonfig}	€	Rekonfigurationskosten
$K_{\text{Stillstand}}$	€	Stillstandskosten
$K_{\text{Sys,IT,LCC}}$	€	IT-Lebenszykluskosten des Gesamtsystems
$K_{\text{SysIntegration}}$	€	Kosten der Systemintegration
$K_{\text{TreiberLizenzen}}$	€	Kosten der Treiberlizenzen
K_{Wartung}	€	Wartungskosten
$K_{\text{Zins,kal}}$	€	Kalkulatorische Zinskosten
$K_{\text{zykl,fix}}$	€	Fixe zyklische Kosten
$K_{\text{zykl,var}}$	€	Variable zyklische Kosten
K_{zyklisch}	€	Zyklische Kosten
$t_{\text{Komp.tausch}}$	h	Zeitdauer des Komponententauschs
t_{Konfig}	h	Zeitdauer zur Konfiguration der Ressourcen
t_{Planung}	h	Zeitdauer zur Planung der Rekonfiguration
$t_{\text{Programmierung}}$	h	Zeitdauer zur Programmierung der Ressourcen
t_{Rekonfig}	h	Zeitdauer der Rekonfiguration
t_{Takt}	s	Taktzeit

1 Einleitung

Πάντα χωρεῖ καὶ οὐδὲν μένει

*Alles bewegt sich und nichts steht still*¹

Heraklit postulierte mit seiner Theorie der „Flussfragmente“, dass eine fortwährende Veränderung ein Grundgesetz aller Dinge sei (STÖRIG 1997, S. 136). So kann ein Fluss in seiner Einheit nur existieren, solange zu jedem Zeitpunkt neue Wasser durch ihn fließen (GRAHAM 2015).

Nach 2500 Jahren ist die Idee des „ewigen Wandels zur Wahrung des Ganzen“ heute aktueller denn je. Der Mensch als Teil eines sozio-ökonomischen Systems unterliegt einerseits immer häufiger auftretenden Veränderungen, ist andererseits durch sein (Konsum-)Verhalten aber auch Verursacher dieses Wandels. Als Dienstleister des gesellschaftlichen Konsums ist die Produktionstechnik von diesen Veränderungen besonders betroffen, die Adaption an geänderte Produkthanforderungen oder Prozessvorgaben ist mittlerweile zu einer fast täglichen Aufgabe geworden (WIENDAHL ET AL. 2002, S. 650). So geben 67 % der Unternehmen in einer Studie der Technischen Universität München an, mehr als 120 Anpassungen pro Jahr in der Produktion durchzuführen (KOCH ET AL. 2015, S. 31).

Hervorgerufen werden diese Änderungen durch verschiedene globale Trends: Seit vielen Jahren ist eine zunehmende Individualisierung von großindustriell hergestellten Endkundenprodukten zu erkennen („Mass Customization“) (KRÜGER ET AL. 2017, S. 707; WIENDAHL ET AL. 2007, S. 783). Hierbei nutzen Unternehmen Diversifizierungsstrategien, um immer kleinere Marktnischen besetzen zu können und somit dem Kundenwunsch nach individuellen Produktlösungen zu entsprechen (WIENDAHL ET AL. 2004, S. 9). Dies führt zu einer Steigerung der Variantenzahl und damit zu einer Erhöhung der

¹PLATON, Zusammenfassung der Ideenlehre des HERAKLIT (PLATON & KRAPINGER 2014, S. 71).

erforderlichen Rüst- und Umbauaufwände (BAUERNHANSL ET AL. 2014, S. 13; LOTTER & WIENDAHL 2012, S. 4).

Als zweiter Trend ist eine Verkürzung und Dynamisierung von Produktlebenszyklen erkennbar (ELMARAGHY & ELMARAGHY 2016; SCHENK ET AL. 2014, S. 500). Dies wird sowohl durch rasch wechselnde Modeströmungen eines anspruchsvollen Kundenmarkts hervorgerufen als auch herstellerseitig durch kürzer werdende Innovations sprünge (ABELE & REINHART 2011, S. 15; SENDLER ET AL. 2013, S. 22 ff.) Die Globalisierung der Weltwirtschaft stellt einen weiteren Einflusstrend dar (NYHUIS ET AL. 2008, S. 1; WIENDAHL ET AL. 2004, S. 15). So ist das weltweite Außenhandelsvolumen in den letzten 60 Jahren um mehr als das Zwanzigfache gestiegen (SENDER ET AL. 2013, S. 22). Diese zunehmende Internationalisierung schafft eine erhebliche Verschärfung des Wettbewerbs (VDMA 2013, S. 22). Insgesamt wird häufig von einem *turbulenten Umfeld* für Unternehmen gesprochen, in dem sich Produktionsparameter sprunghaft ändern können – bei gleichzeitig abnehmender Vorhersagbarkeit von Veränderungen (WIENDAHL 2014, S. 21 f.; BAUERNHANSL ET AL. 2014, S. 13 f.).

Um unter diesen Bedingungen wettbewerbsfähig zu bleiben, ergibt sich für Unternehmen die Notwendigkeit effizienter als bisher auf geplante sowie ungeplante Änderungen reagieren können (NYHUIS ET AL. 2008; KOREN & SHPITALNI 2010). Das Vermögen einer Fabrik, „aktiv strukturelle Veränderungen auf allen Ebenen bei geringem Aufwand durchführen zu können“ beschreibt WIENDAHL (2002, S. 123) als *Wandlungsfähigkeit*. NYHUIS ET AL. (2008) präzisieren den Begriff als „Potential, (...) jenseits der vorgehaltenen Korridore organisatorische und technische Veränderungen bei Bedarf reaktiv oder sogar proaktiv durchführen zu können“. Dies unterscheidet sich von der *Flexibilität*, bei der Anpassungen nur innerhalb eines (vorab) geplanten Korridors geschehen können (NYHUIS ET AL. 2008, S. 24). Beide Begriffe werden auch als *Veränderungsfähigkeit* zusammengefasst und können auf unterschiedlichen Ebenen in der Fabrik angewandt werden – mit jeweils eigenem Betrachtungsfokus und Maßnahmenbündel (WIENDAHL 2002).

1.1 Vision der rekonfigurierbaren Produktionssysteme

Als wirtschaftlich orientierte Institutionen streben Unternehmen nach Gewinnmaximierung. Zur Erreichung dieses Ziels gilt die Rationalisierung (lat. „vernünftig machen“) als ein wichtiges Mittel (BRONNER 2003, S. 2). Kenngröße der Rationalisierung ist die

Produktivitätssteigerung, welche als Verbesserung des Quotienten zwischen der erbrachten Produktionsleistung und den dafür eingesetzten Mitteln gilt (FISCHER 1975, S. 15; BRONNER 2003, S. 2 f.). Produktivitätssteigerungen werden dabei häufig auch durch die Automatisierung von bisher manuell durchgeführten Tätigkeiten erreicht (BRONNER 2003, S. 2).

Besonders Unternehmen in Erste-Welt-Ländern besitzen einen hohen Automatisierungsgrad, um wirtschaftlich trotz hoher Lohnkosten produzieren zu können (AUTOMATIONSPRAXIS 2013; NAUMANN ET AL. 2010, S. 631). WIENDAHL ET AL. (2002, S. 652) stellen jedoch fest, dass ein hoher Automatisierungsgrad die Komplexität und damit den Aufwand einer Erst- und Wiederinbetriebnahme erhöht. Hierdurch steigen die Fix- und somit auch die Stückkosten, weshalb Automatisierungssysteme bevorzugt bei hohen Stückzahlen eingesetzt werden (LOTTER & WIENDAHL 2012, S. 192 f.; FELDMANN & SPUR 2014, S. 437). Dies läuft jedoch den vorgestellten Trends entgegen, so sehen ABELE & REINHART (2011, S. 99) den zukünftigen Automatisierungseinsatz vor allem bei geringen bis mittleren Stückzahlen.

Zur Lösung dieses Dilemmas entwickelte sich um die Jahrtausendwende das Konzept der rekonfigurierbaren Produktionssysteme. *Rekonfigurierbarkeit* beschreibt nach WIENDAHL (2002, S. 126) eine *Veränderungsfähigkeit* auf technischer Ebene, wie etwa das Umrüsten von Arbeitsstationen. KOREN ET AL. (1999, S. 529) definieren rekonfigurierbare Produktionssysteme (engl. *Reconfigurable Manufacturing Systems, RMS*) hierin, dass sie inhärent „für eine schnelle Anpassung ihrer Struktur, Hardware- und Softwarekomponenten ausgelegt sind, um effizient Änderungen der Produktionskapazität und Funktionalität innerhalb einer herzustellenden Teilefamilie herbeirufen zu können“. Eine Rekonfiguration kann dabei unterschiedliche Eigenschaften des Systems betreffen, wie etwa die Produktionskapazität, die Funktionalität der Anlage oder dessen Automatisierungsgrad (WIENDAHL ET AL. 2007, S. 786 ff.; KOREN ET AL. 1999, S. 529 f.).

1.2 Cyber-physische Systeme als Enabling Technology

In den letzten Jahren ist es zu einem starken Anstieg beim Einsatz von Informationstechnologie innerhalb der Produktionstechnik gekommen (WGP 2016 S. 7; ACATECH 2011 S. 11). Als Folge wurde in Deutschland im Jahr 2011 der Begriff „Industrie 4.0“ geschaffen, welcher den Wandel hin zu einer umfassend digitalisierten Produktion

beschreibt (WGP 2016). Die veränderungsfähige Produktion spielt bei Industrie 4.0 eine zentrale Rolle (WGP 2016). Laut der durch das Bundeswirtschaftsministerium geschaffenen PLATTFORM INDUSTRIE 4.0 soll dieses Ziel durch die Entwicklung von „intelligenten, [...] interoperablen Modulen“ erreicht werden, welche sich „weitgehend selbstständig anpassen“ (PLATTFORM I4.0 2016, S. 18).

In diesem Zusammenhang wurde international der Begriff der *Cyber-physischen Systeme* (CPS) geprägt. CPS sind eine Weiterentwicklung traditioneller eingebetteter Systeme, welche um globale Kommunikationsschnittstellen (z. B. Internet) erweitert wurden. Diese Systeme interagieren über Sensoren und Aktoren mit ihrer Umwelt (physisch) und können Informationen und Dienstleistungen verteilter externer Systeme nutzen, auf deren Basis sie Entscheidungen treffen (cyber). (ACATECH 2011; BROY 2010)

Überträgt man CPS in die Produktionswelt, so entstehen *Cyber-physische Produktionssysteme* (CPPS), die aus „flexiblen, [...] sich selbstkonfigurierenden Produktionsressourcen“ bestehen. Die WISSENSCHAFTLICHE GESELLSCHAFT FÜR PRODUKTIONSTECHNIK (WGP) sieht in CPPS eine Chance die industrielle Produktion für die neuen Anforderungen des Marktes fit zu machen. (WGP 2016, S. 11 f.)

CPS können somit als eine Enabling Technology gesehen werden (ACATECH 2011, S. 11), die technische Potentiale für eine Realisierung von rekonfigurierbaren Produktionssystemen liefern.

1.3 Problemstellung und Betrachtungsraum der Arbeit

Obwohl die Vision von RMS seit mehreren Jahrzehnten existiert, wurde das gewünschte Maß an Umsetzung im Status quo der Industrie noch nicht realisiert (SELIG 2010, S. 19). Hieraus ergibt sich ein Forschungsbedarf zur Reduktion heutiger Rekonfigurationsaufwände in Produktionssystemen.

Im Rahmen der zu wählenden Betrachtungsebene zeigen NYHUIS ET AL. (2008, S. 37 ff.), dass sich ein Großteil der Forschungsarbeiten mit Wandlungsfähigkeit aus einer ganzheitlichen Fabriksicht beschäftigt. Die Autoren empfehlen stärker auf *technische Lösungen* innerhalb der *Komponenten-* und *Arbeitsstationsebene* einzugehen, um vorhandene Forschungslücken zu schließen.

Betrachtet man die Rekonfiguration auf Komponentenebene, befindet man sich im Bereich der Integration von Automatisierungstechnik, auch als „Systemintegration“

bezeichnet. Diese ist noch immer als Problemstellung anzusehen, da sie nach GRIF-FEN (2014, S. 13 f.) sowie REINHART & KRUG (2010) nur mit hohem Zeitaufwand und mit dem Einsatz von Experten durchzuführen ist. Gründe hierfür sind heterogene Systemlandschaften, welche sich in proprietären Herstellerschnittstellen, unterschiedlichen Kommunikationsprotokollen sowie fehlenden Beschreibungsstandards äußern (WEINLÄNDER 2016, S. 19; PLATTFORM I4.0 2015, S. 45 f.). Hinzu kommt eine hohe Systemkomplexität der Produktionsanlagen aufgrund steigender Funktionsumfänge und verdichteter Kommunikationsbeziehungen (KLEIN 2016, S. 36; SENDLER ET AL. 2013, S. 107). Generell kann von einem Kostenanteil der Geräteintegration von bis zu 50 % im Verhältnis zu den Gesamtkosten einer Automatisierungsanlage ausgegangen werden (HAMMERSTINGL & REINHART 2015a, S. 17; STATISTA 2015; JAMMES & SMIT 2005, S. 62). BRECHER ET AL. (2010) sehen deshalb große Potentiale zur Kosteneinsparung bei der Integration von Automatisierungskomponenten verborgen. *Hierdurch rückt die Systemintegration in den zentralen Betrachtungsfokus dieser Arbeit.*

Wird die Systemintegration anhand ihrer domänenspezifischen Zeitanteile analysiert, so ist zu erkennen, dass der informationstechnische Anteil (Konfiguration, Programmierung etc.) dominierend ist. FELDMANN & SPUR (2014, S. 638) beschreiben, dass bis zu 90 % der Inbetriebnahmezeit für Steuerungstechnik und Elektrik aufgewandt wird. Von diesem Anteil sind wiederum bis zu 70 % der Aufwände auf Fehler im Steuerungscode zurückzuführen. Die vorliegende Arbeit wird daher auf den Bereich der *Informationstechnik* eingegrenzt, die Domänen der Elektrik und Mechanik stehen nicht im Fokus.

Für die Untersuchung vorherrschender Prozesse und Technologien wird auf den Bereich der *Montage* eingegrenzt. Nach ELMARAGHY & ELMARAGHY (2016, S. 4) und FELDMANN & SPUR (2014, S. 3) kommt der Montage aufgrund der realisierten Wertschöpfung und Beschäftigungszahlen eine gehobene volkswirtschaftliche Bedeutung zu. MÜLLER ET AL. (2009)² geben einen Wertschöpfungsanteil der Montage am gesamten Produktionsprozess von bis zu 70 % an, jedoch auch einen Kostenanteil von 44 %. Bedingt durch einen sehr niedrigen Automatisierungsgrad von 6 % bis 20 % (FOCUS 2016, S. 80) ist die Montage zu dem teuersten Produktionsprozess geworden (LOTTER & WIENDAHL 2012, S. 4). Dies rührt daher, dass die Montage durch die Auswirkungen des turbulenten Marktumfelds besonders stark betroffen ist (FELDMANN & SPUR 2014,

²Zitiert nach ABELE & REINHART 2011, S. 99.

S. 6), aufgrund der hohen Integrationsaufwände die geforderte Veränderungsfähigkeit jedoch nur bei manueller Arbeitskraft erreicht werden kann (HU ET AL. 2011, S. 726).

Für die Gestaltung zukünftiger Montagesysteme sehen LOTTER & WIENDAHL (2012, S. 4), WIENDAHL ET AL. (2004, S. 15) als Zielstellung eine Steigerung der *Anpassbarkeit* und der *Wiederverwendung eingesetzter Betriebsmittel*. HU ET AL. (2011, S. 724) ergänzen, dass die notwendige Zeit zur Planung und dem Entwurf von Montagesystemen *beträchtlich reduziert* werden muss. Darüber hinaus werden von ABELE & REINHART (2011, S. 100 f.) die *Reduktion des Programmieraufwands* genannt. Für ELMARAGHY & ELMARAGHY (2016, S. 7 ff.) stellt die Entwicklung von rekonfigurierbaren Montagesystemen ein Schlüsselement zur Umsetzung von RMS dar. Auch sie unterstreichen die Notwendigkeit einer verbesserten Modularisierung, Systemintegration und eingebetteten Intelligenz in Montageanlagen.

Damit korrelieren die Forschungsbedarfe in der Montage mit den Defiziten im Bereich der Rekonfiguration. Die vorgestellten Eigenschaften von CPS, wie etwa Selbstvernetzung und -konfiguration, können Lösungsmöglichkeiten für die genannten Forschungsbedarfe bieten. *Als Forschungsthese wird deshalb postuliert, dass die Entwicklung Cyber-physischer Automatisierungskomponenten eine Steigerung der Rekonfigurationsfähigkeit von Produktionsanlagen bieten kann.* Dies gilt es im weiteren Verlauf der Arbeit näher zu untersuchen.

Abbildung 1.1 zeigt den definierten Betrachtungsraum in der Zusammenfassung.

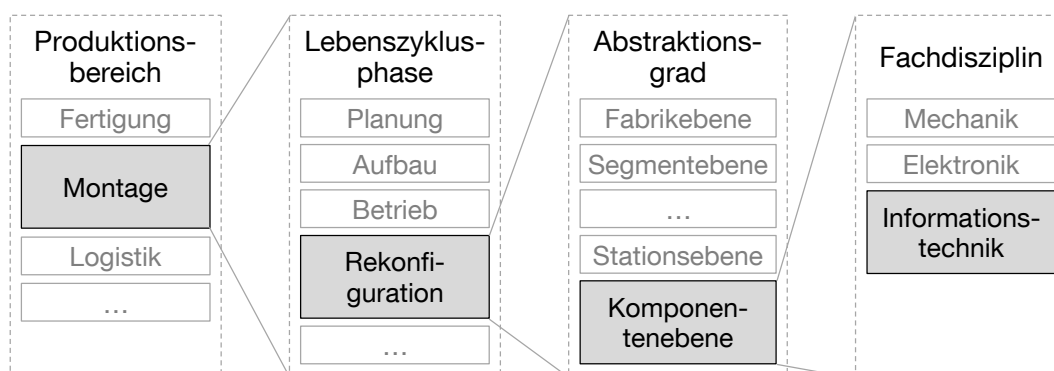


Abbildung 1.1: Der Betrachtungsraum der Arbeit bezieht sich auf die Rekonfiguration von Automatisierungskomponenten in der Montage aus informationstechnischer Sicht.

1.4 Forschungsmethodischer Rahmen und Aufbau der Arbeit

Die vorliegende Dissertation ist dem Gebiet der Ingenieurwissenschaften zuzuordnen. Diese sind wissenschaftssystematisch dem Bereich der Realwissenschaften zugehörig. Realwissenschaften beschäftigen sich mit der „Beschreibung, Erklärung und Gestaltung empirisch wahrnehmbarer Wirklichkeitsausschnitte“. (ULRICH & HILL 1976, S. 305 f.) Ergebnis dieser konstruktiven Dissertation ist eine Methodik zur Steigerung der Rekonfigurationsfähigkeit von Montageanlagen durch Cyber-physische Feldgeräte.

Als forschungsmethodisches Vorgehen wird die allgemeine *Design Research Methodology (DRM)* nach BLESSING & CHAKRABARTI (2009) gewählt. Diese definiert vier Phasen (vgl. Abbildung 1.2):

In der Phase *Research Clarification* erfolgt die Beschreibung des Ausgangs- sowie Zielzustands. Kapitel 1 identifiziert hier deskriptiv auf Basis von Literaturrecherchen die relevante Problemstellung hoher Rekonfigurationsaufwände und die zu erreichende Vision rekonfigurierbarer Produktionssysteme (RMS).

Die nachfolgende Phase *Descriptive Study I* dient zur Exploration bedeutender Einflussfaktoren auf die Problemstellung und den Zielzustand. Hierzu werden im Grundlagenteil (Kapitel 2) relevante Bestandteile die Montage- und Automatisierungstechnik beschrieben sowie deren Wirkungen auf die Rekonfigurationsfähigkeit erläutert. Als Ergebnis erfolgt die induktive Ableitung notwendiger Handlungsfelder. Im Hinblick auf diese wird in Kapitel 3 auf Basis von Literaturrecherchen der Stand der Erkenntnisse analysiert. Während Kapitel 1 und 2 vorrangig terminologisch-deskriptiv geprägt sind, ist das Ziel von Kapitel 3 die empirisch-induktive Identifikation von relevanten, aber noch ungelösten Faktoren zur Steigerung der Rekonfigurationsfähigkeit.

Mit Übergang in die präskriptive Phase (*Prescriptive Study*) wird aus den bisher erarbeiteten Ergebnissen in Kapitel 4 eine übergreifende Zielstellung definiert, für die kreativ-synthetisch ein Gesamtkonzept auf Basis dreier Lösungsbausteine entwickelt wird. Diese werden in den anschließenden Kapiteln beschrieben: In Kapitel 5 wird eine Methodik zur automatisierten Vernetzung von Feldgeräten erstellt und in Kapitel 6 ein Modell zur fähigkeitsbasierten Prozessbeschreibung formuliert. Beide Bestandteile werden in Kapitel 7 für den Aufbau einer Architektur zur aufgabenorientierten Ressourcenauswahl und Programmierung genutzt. Das Vorgehen innerhalb der Kapitel besteht aus terminologisch-deskriptiven Tätigkeiten zur Definition der domänenspezifischen Begriffswelten und empirisch-induktiven Schlussfolgerungen zum Aufbau

der Methoden. Für die Lösung wohl definierter und bekannter Teilprobleme (z. B. Wegoptimierung) werden analytisch-deduktive Ansätze genutzt.

Während die vorangegangenen Phasen vorrangig explorativer Natur sind, ist die Zielstellung der letzten Phase *Descriptive Study II* konfirmatorischer Art. Sie dient der Bewertung von Zielerreichung und Generalisierbarkeit der Lösung. In Kapitel 8 wird die Methodik prototypisch anhand eines Referenzszenarios implementiert. Die Einzelbausteine werden hierauf explikativ und instrumentell validiert (TÖPFER 2012, S. 150 ff.), das Vorgehen ist dabei sowohl empirisch als auch analytisch-deduktiv. Zusätzlich erfolgt eine wirtschaftliche und nutzerbasierte Bewertung der Methodik. In Kapitel 9 wird eine abschließende Zusammenfassung der Erkenntnisse und Ergebnisse dieser Arbeit gegeben.

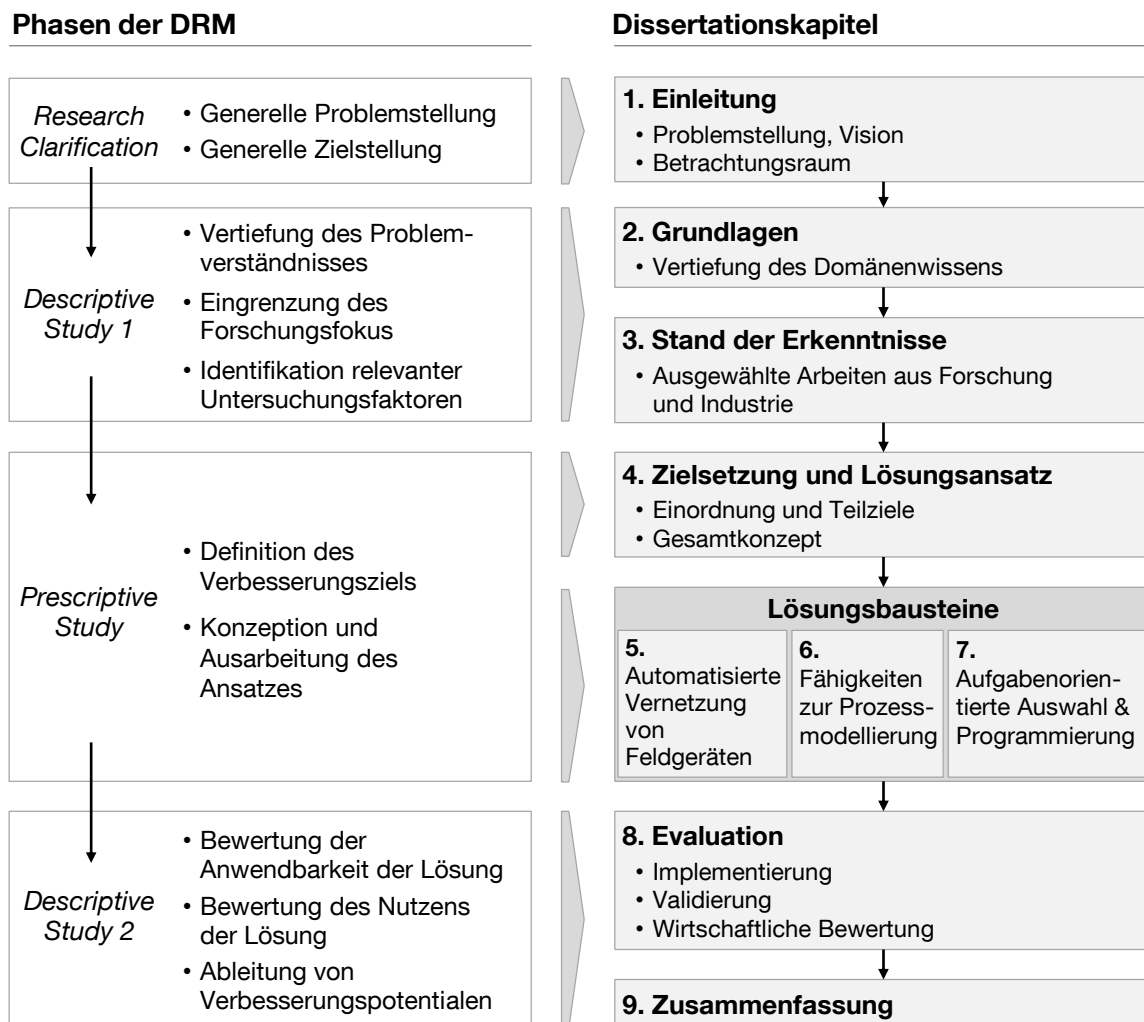


Abbildung 1.2: Der Aufbau der Arbeit orientiert sich an den Phasen und Ergebnissen der Design-Research-Methodology (DRM), BLESSING & CHAKRABARTI (2009).

2 Grundlagen

Es wird angenommen, dass Leser dieser Arbeit ein grundlegendes Verständnis der Ingenieurwissenschaften, Produktionstechnik und Softwareentwicklung besitzen. Weitere Grundlagen, den Betrachtungsraum betreffend, werden in diesem Kapitel behandelt.

Abschnitt 2.1 gibt einen Überblick über die Domäne der Montagetechnik. Es werden die heterogene Strukturen und Aktivitäten innerhalb des Lebenszyklus von Montageanlagen aufgezeigt, ebenso wie mögliche Rekonfigurationsarten und deren Befähiger. Abschnitt 2.2 widmet sich der Automatisierungstechnik und zeigt deren Hürden im Rahmen der Vernetzung, Konfiguration und Programmierung. Abschnitt 2.3 definiert die Eigenschaften Cyber-physischer Systeme und spannt einen Bogen zu Ansätzen im nachfolgenden Stand der Erkenntnisse.

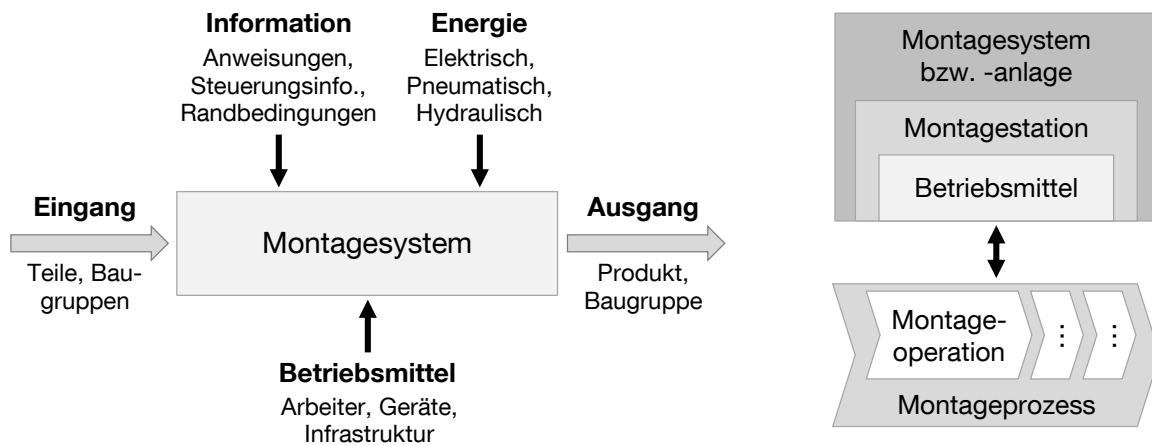
2.1 Montagetechnik

In der *Montage* werden gefertigte Einzelteile zu einem angestrebten Produkt höherer Komplexität zusammengesetzt. Hierzu zählen alle Vorgänge, die dem Zusammenbau des Produkts dienlich sind, wie bspw. Fügeoperationen, die Handhabung oder Funktionsprüfungen. (CIRP 2011 S. 2; LOTTER & WIENDAHL 2012, S. 1 f.)

Montagesysteme (auch *Montageanlagen*) sind Einrichtungen zur Realisierung der Montage. Eingangsgrößen eines Montagesystems sind Bauteile oder Baugruppen. Gesteuert durch Informationen (z. B. Produktionsaufträge) montieren Produktionsmittel (Arbeiter und Maschinen) die Einzelteile unter Zuhilfenahme von Energie zu einem Produkt oder einer höherwertigen Baugruppe (vgl. Abbildung 2.1a). (CIRP 2011 S. 2)

Montagesysteme bestehen aus mehreren verketteten *Montagestationen*. In einer Montagestation erfüllen Betriebsmittel meist eine dezidierte *Montageoperation* innerhalb eines Montageprozesses (vgl. Abbildung 2.1b, CIRP 2011). *Betriebsmittel* sind hierbei

Anlagen, Geräte und Werkzeuge, die zur Erzeugung der betrieblichen Leistung dienen, ohne dabei selbst in dem Prozess mit aufzugehen. (CIRP 2011 S. 6 ff.)



(a) Systembetrachtung (erweiterte Darstellung nach CIRP 2015 S. 2)

(b) Strukturelle Ebenen

Abbildung 2.1: Montagesysteme

Montagesysteme können aus manuellen oder automatisierten Montagestationen bestehen. *Automatisierung* beschreibt hierbei den Zustand, wenn die in der Montagestation eingesetzten Betriebsmittel durch ein programmierbares Steuerungssystem kontrolliert werden (CIRP 2011 S. 16). Besteht ein Montagesystem aus einer Kombination von automatisierten und manuellen Stationen, wird von einem *hybridem Montagesystem* gesprochen (CIRP 2011 S. 186). Zwischen Montagestationen wird das Produkt oder die Baugruppe meist durch Verkettungseinrichtungen transportiert, wie etwa Runddrehtische oder Gurtförderbänder (CIRP 2011 S. 270 ff.).

2.1.1 Prozessschritte der Montage

LOTTER & WIENDAHL (2012, S. 2) unterteilen die Montage in fünf Prozesskategorien, das Fügen, Handhaben, Kontrollieren, Justieren sowie die Sonderoperationen. Für jede Kategorie lassen sich weitere Subprozesse definieren (vgl. Abbildung 2.2), die durch Montagestationen oder einzelne Betriebsmittel in der Montage ausgeführt werden.

Fügen ist hierbei durch die DIN 8593 als das „[...] auf Dauer angelegte Verbinden oder sonstige Zusammenbringen von zwei oder mehr Werkstücken geometrisch bestimmter Form [...]“ definiert. Neben weiteren Hauptgruppen wie etwa dem Urformen oder Trennen, zählt Fügen zu den Fertigungsverfahren. Im Bereich des Fügens existieren

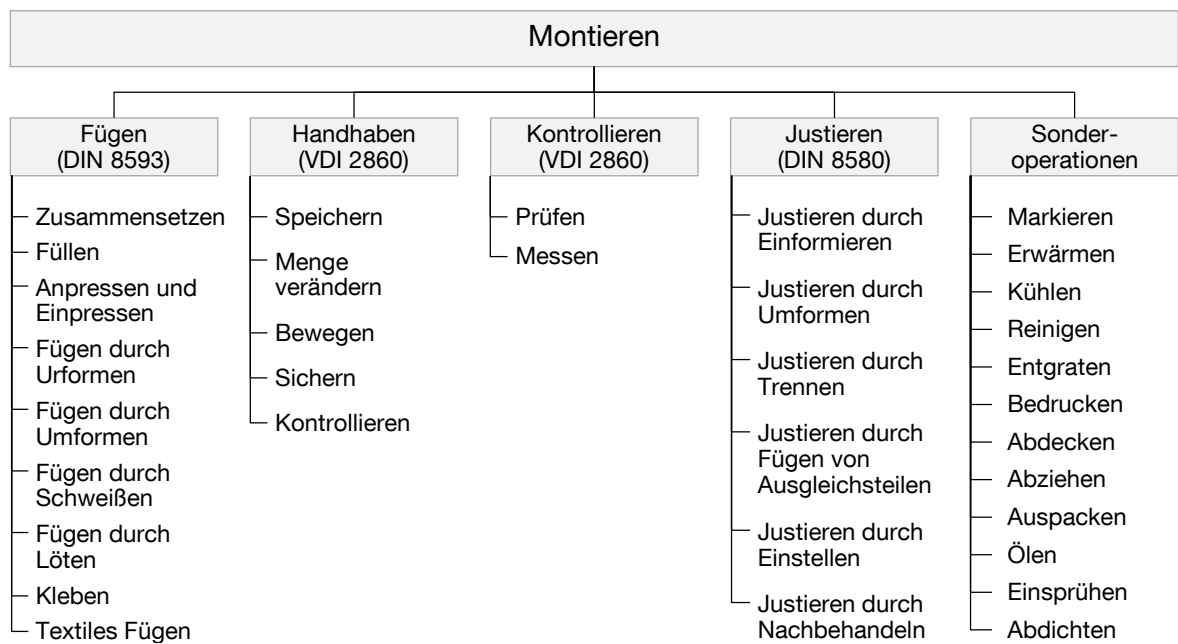


Abbildung 2.2: Übersicht der Funktionen in der Montage, in Anlehnung an LOTTER & WIENDAHL (2012, S. 2). In der Darstellung fällt der geringe Detaillierungsgrad der Kontrollfunktionen auf.

eine Vielzahl von untergeordneten Gruppen, bspw. das *Fügen durch Zusammensetzen* oder das *Fügen durch Kleben*. (DIN 8593, S. 5)

Das *Handhaben* beschreibt das „Schaffen, definierte Verändern oder vorübergehende Aufrechterhalten einer vorgegebenen räumlichen Anordnung von geometrisch bestimmten Körpern [...]“ (VDI 2860, S. 2). Fügeoperationen setzen vor und nach der Prozessausführung Handhabungsoperationen voraus. Als Abgrenzung zur Handhabung beginnt das Fügen, „sobald ein minimaler Zusammenhalt auch bei Wegfall der Handhabungsfunktion sichergestellt ist. Der [Füge-]Vorgang endet, wenn die geforderte Festigkeit der Verbindung bei gewünschter Position und räumlicher Lage der zu fügenden Partner erreicht ist“ (FELDMANN & SPUR 2014, S. 8)¹.

Kontrollieren dient dem Feststellen geforderter Eigenschaften (Prüfen) oder dem Vergleich mit gegebenen Bezugsgrößen (Messen). Es tritt als Teilprozess in allen Fertigungsfolgen und Handhabungsoperationen auf. Beispiele sind das Messen einer Orientierung oder Prüfen eines Gewichts. (VDI 2860, S. 7 f.)

¹Sekundärzitat. Originalquelle STÖFERLE et al. (1974) nicht auffindbar.

Als *Justieren* werden Tätigkeiten bezeichnet, die Bauteile entsprechend getroffener Vorgaben (meist mechanisch) einstellen, bspw. die Festlegung einer Achsposition zwischen zwei Lagern (LOTTER & WIENDAHL 2012). Als letzte Kategorie werden *Sonderoperationen* definiert, unter die alle Funktionen fallen, welche notwendiger Bestandteil der Montage sind, sich aber nicht anderweitig zuordnen lassen (CIRP 2011 S. 134). Beispiele hierfür sind das Erwärmen oder Reinigen von Gegenständen.

2.1.2 Lebenszyklus von Montageanlagen

Rekonfiguration als Bestandteil des Wandlungsregelkreises

Montageanlagen sind eine Untergattung der Produktionssysteme und verfügen dementsprechend auch über deren Lebenszyklus. Bei Produktionsanlagen besteht dieser aus den Phasen der *Planung*, der *Realisierung*, des *Betriebs* sowie der *Redistribution* (WIENDAHL ET AL. 2002, S. 651).

Die Planungsphase von Montagesystemen unterteilt BULLINGER (1986) in die drei Bereiche der *Ablaufplanung*, des *Systementwurfs* und der *Ausarbeitung*. Basierend auf der Produktstruktur und Stückliste, definiert die *Ablaufplanung* notwendige Prozessfolgen und deren Randbedingungen, um das gewünschte Ergebnis zu erzielen. Während die Ablaufplanung das Montagesystem noch auf einer abstrakt funktionalen Ebene beschreibt, spezifiziert der *Systementwurf* die zugehörigen Betriebsmittelarten, deren Dimensionierung und Anordnung. In der Phase der *Ausarbeitung* werden die generischen Betriebsmittel durch konkrete Varianten ersetzt und deren Prozesse in den Montagestationen weiter spezifiziert.

Die Realisierungsphase beginnt mit der *Konstruktion*, gefolgt von der *Fertigung* bzw. *Beschaffung* der zu nutzenden Betriebsmittel. KRUG (2013, S. 28 f.) und HODEK (2013, S. 12 f.) definieren in der Realisierungsphase die *Systemintegration* als zentrale Tätigkeit. Diese besteht aus dem *Aufbau*, der *Konfiguration* sowie der *Programmierung* der Betriebsmittel. Die Realisierungsphase gipfelt in der *Inbetriebnahme*, bei der schrittweise die einzelnen Komponenten der Anlage aktiviert und deren Zusammenspiel getestet werden. Dem schließt sich ein schrittweiser *Anlauf* der Anlage an, welcher mit dem Erreichen der Soll-Ausbringungsmenge fließend in den *Betrieb* übergeht. (KRUG 2013, S. 28 f.; HODEK 2013, S. 11 ff.)

Am Ende des Anlagen-Lebenszyklus stehen der *Rückbau* oder eine *Redistribution* an. Abbildung 2.3 zeigt die vorgestellten Phasen in der Übersicht.

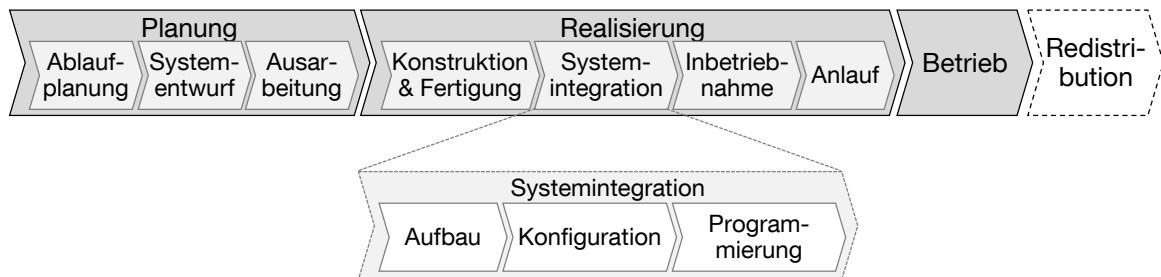


Abbildung 2.3: Phasen des Anlagenlebenszyklus. Vereinheitlichte Darstellung der Quellen BULLINGER (1986), WIENDAHL ET AL. (2002, S. 651) und KRUG (2013, S. 28).

Bedingt durch die Megatrends, werden zunehmend häufigere Anpassungen an Produktionsmitteln notwendig (ELMARAGHY & ELMARAGHY 2016, S. 4). Die in Abbildung 2.3 gezeigten Phasen werden dadurch nicht mehr einmalig durchlaufen, sondern zu einem Bestandteil eines Regelkreises, bei dem auftretende Anforderungen mit den vorherrschenden Produktionsparametern abgeglichen und bei Bedarf angepasst werden (KOCH 2017, S. 27 f.; PAWELLEK 2008, S. 17; ELMARAGHY 2006, S. 263 f.) (vgl. Abbildung 2.4). Der Aufwand der Anpassung hängt dabei von der Veränderungsfähigkeit des Systems ab. Wird diese auf einer Betriebsmittel- und Stationsebene durchgeführt, spricht man von einer *Rekonfiguration* des Systems (WIENDAHL 2002, S. 127). Das erneute Durchlaufen der Lebenszyklusphasen ähnelt dabei dem initialen Durchgang bei einem Fabrikstart (ELMARAGHY 2006, S. 20 f.).

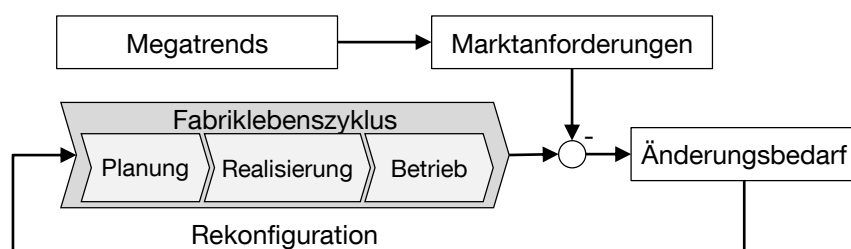


Abbildung 2.4: Wiederkehrender Ablauf des Anlagenlebenszyklus bedingt durch Änderungsbedarfe aufgrund wechselnder Marktanforderungen.

Aktivitäten zur Durchführung der Rekonfiguration

Die während einer Rekonfiguration durchzuführenden Aktivitäten können aus Quellen der Montage- und Fabrikplanung entnommen werden². Die Bezeichnung und Zuordnung zu einzelnen Phasen im Fabriklebenszyklus sind dabei nicht immer eindeutig, es lassen sich jedoch grundlegende Gemeinsamkeiten ausmachen, welche nachfolgend beschrieben werden.

Unter der *Prozessplanung* werden Aktivitäten zusammengefasst, die auf abstrakter, funktionaler Ebene das zu realisierende Montagesystem spezifizieren. Ausgehend von einem zu montierenden Produktspektrum werden der prinzipielle Montageprozess und dessen Stationen definiert. Die Festlegung des Automatisierungsgrads und der einzusetzenden Operationsverfahren fallen unter die Aktivität der Technologieplanung. In der Montageplanung werden die durchzuführenden Einzelprozesse (vgl. Abschnitt 2.1.1) für einen Zusammenbau des Produkts sowie deren Reihenfolge festgelegt, die Prüfplanung beschreibt selbige für die Untersuchung relevanter Qualitätsmerkmale.

Der *Systementwurf* beschreibt die technische Ausgestaltung des Montagesystems. Auf Basis der ermittelten Produkt- und Prozessanforderungen müssen geeignete Betriebs- und Prüfmittel ausgewählt und beschafft werden. Parallel hierzu geschieht die Festlegung der Arbeitsinhalte pro Montagestation und deren geometrische Ausgestaltung. Die Materialflüsse zwischen den Montagestationen werden innerhalb der Transportplanung spezifiziert, dies beinhaltet auch die Auswahl geeigneter Transportmittel. Existieren mehrere Konzepte, welche die gegebenen Anforderungen und Randbedingungen erfüllen, muss abschließend eine Alternativenbewertung durchgeführt werden. Die geschieht meist auf Basis monetärer Kennzahlen,

Innerhalb der *Systemintegration* wird die Anlage aufgebaut, indem einzelne Geräte und Vorrichtungen zu Montagestationen kombiniert werden. Dies umfasst einerseits den mechanischen und elektrischen Aufbau als auch eine softwareseitige Installation. Unter Letztere fallen die Konfiguration der Kommunikationsbeziehungen sowie die Parametrierung auf Basis der definierten Anforderungen. Daraufhin können die jeweils erstellten Subsysteme (wie etwa Bildverarbeitung, Steuerungstechnik) programmiert

²SCHENK ET AL. (2014, S. 294 ff.), PAWELLEK (2008, S. 137 f.), ELMARAGHY (2006, S. 207), EVERSHEIM (2002, S. 6 ff.), BULLINGER (1986) sowie HODEK (2013, S. 12 f.), KRUG (2013) und GROSSMANN (2008, S. 15 f.)

und zu einem Gesamtablauf verschaltet werden. Hierzu zählt auch die Einrichtung von Schnittstellen zu übergeordneten IT-Systemen, wie etwa Manufacturing Execution Systems (MES). Den Abschluss stellt die Inbetriebnahme dar, bei der schrittweise die Komponenten eingeschaltet und getestet werden. Eine weiterführende technische Beschreibung dieser Aktivitäten wird in Abschnitt 2.2 gegeben.

Für den aufgestellten Betrachtungsraum stehen die Aktivitäten des *Systementwurfs* und der *Systemintegration* verstärkt im Fokus. Aktivitäten der Betriebsmittelkonstruktion, deren mechanischer Aufbau und die Strukturanordnung werden nicht betrachtet. Hierdurch ergibt sich das in Abbildung 2.5 dargestellte Aktivitätenfeld.

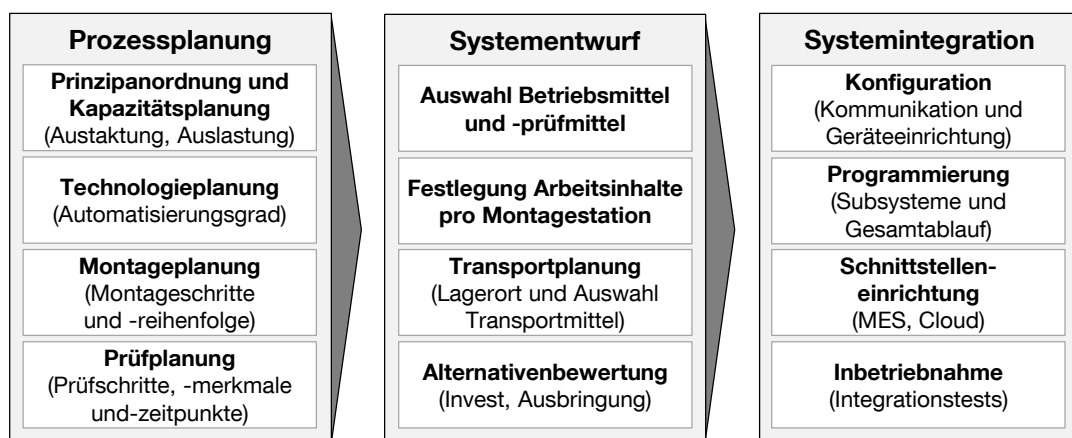


Abbildung 2.5: Durchzuführende Tätigkeiten bei der Planung und Realisierung von Montagesystemen aus Sicht der Automatisierungstechnik im Betrachtungsraum.

Im Falle von Rekonfigurationen existiert bereits ein Produktspektrum samt zugehörigem Montagesystem, welches in Teilen auf neue Randbedingungen angepasst wird. Die vorgestellten Aktivitäten werden somit selektiv für betroffene Prozesse bzw. Betriebsmittel durchgeführt. Die Ausführung übernehmen heutzutage Planungsingenieure und Systemintegratoren (BIRKHOFER 2011, S. 16 ff.). Hierdurch ergibt sich ein hohes Maß an Investitionskosten, hervorgerufen durch einen hohen Personaleinsatz gepaart mit notwendigem Expertenwissen.

2.1.3 Anpassungsdimensionen und -befähiger

EVERSHEIM (1983, S. 27) definiert fünf Flexibilitätsdimensionen von Montagesystemen, innerhalb derer Anpassungen vorgenommen werden können:

Umsteuer-Flexibilität wird benötigt, um zu spezifischen Zeitpunkten Materialflüsse zu verändern, bspw. bei unterschiedlichen Produktderivaten. *Umstell*-Flexibilität beschreibt die zeitpunktorientierte Anpassungsfähigkeit innerhalb einer Montagestation, bspw. durch den Einsatz von automatisierten Greiferwechselsystemen. Auf einen längeren Zeithorizont beziehen sich die *Umrüst*- und *Umbau*-Flexibilität, die etwa bei der Umstellung zwischen zwei Chargen zur Geltung kommen. Erstere bezieht sich auf Anpassungsmöglichkeiten innerhalb einer Station (z. B. Aufhängungsmöglichkeiten von Behältern), Letztere beschreibt den Austausch ganzer Montagestationen durch neue Module (z. B. manuelle Arbeitsstation mit automatisierter Montagezelle). Die letzte Flexibilitätsart, *Störungs*-Flexibilität, ist ereignisgesteuert und beschreibt die Möglichkeit, innerhalb des Montagesystems auf Ausfälle reagieren zu können (z. B. durch effizienten Ersatzteilaustausch oder veränderte Materialflüsse).

Um diese Flexibilitätsarten ausgestalten zu können, wurden durch WIENDAHL (2005, S. 25 ff.) fünf Eigenschaften charakterisiert, die Produktionsanlagen zu einer effizienten Rekonfiguration befähigen (vgl. ELMARAGHY 2009, S. 17):

- *Mobilität* – Örtliche Beweglichkeit von Fabrikelementen (z. B. Betriebsmittel, Montagestationen).
- *Universalität* – Eigenschaft eines Fabrikelements, für verschiedene Aufgaben und Anforderungen einsetzbar zu sein.
- *Skalierbarkeit* – Erweiter- und Reduzierbarkeit eines Fabrikelements, realisiert durch technische, räumliche und organisatorische Anpassungen.
- *Kompatibilität* – (Passende) Vernetzungsfähigkeit von Fabrikelementen in den Bereichen Material, Energie und Informationstechnik.
- *Modularität* – Aufbau von Fabrikelementen aus standardisierten, autark funktionsfähigen Einheiten mit standardisierten Schnittstellen („Plug&Produce-Module“).

WIENDAHL (2005, S. 26 f.) sieht die *Modularität* als bedeutendsten Befähiger. Für den Bereich der Montage ergänzt ELMARAGHY (2009, S. 17) die Wichtigkeit der Möglichkeit zur *stufenweisen Automatisierbarkeit* – bedingt durch den Umstand, dass viele Montageoperationen sowohl manuell als auch automatisiert durchgeführt werden können.

2.1.4 Komponenten automatisierter Montageanlagen

Elemente einer Montagestation, welche die unter Abschnitt 2.1.1 beschriebenen Prozesse zur Erfüllung einer Montageaufgabe umsetzen, werden synonym als *Systeme*, *Komponenten*, *Ressourcen* oder *Betriebsmittel* bezeichnet (vgl. BACKHAUS 2016, S. 15 f.). Aus Sicht der Automatisierungstechnik wird auch von *Feldgeräten* („Geräte im Feld“ bzw. „nah am Prozess“) gesprochen (FRÜH & AHRENS 2009; DIN 62390).

Die Vielzahl und Vielfalt an durchzuführenden Operationen in der Montage spiegelt sich in der Heterogenität der eingesetzten Komponenten wider. Je nach Sichtweise können diese unterschiedlich strukturiert werden, wie etwa nach Hauptfunktion, Wirkprinzip oder Komplexitätsgrad (vgl. SECKNER 2008, S. 49 ff.; SCHMIDT 1992, S. 21).

In der Literatur existiert keine einheitliche Unterteilung, weshalb in Tabelle 2.1 bekannte Klassifizierungen vereinigt und erweitert wurden. Generell bestehen Montageanlagen aus Aktoren (Prozesseingriff), Sensoren (Prozessaufnahme), Steuerungen (Prozessablauf) und Benutzerschnittstellen zur Mensch-Maschine-Interaktion (MMI).

Im Bereich der Aktorik setzen *Bewegungseinrichtungen* vorrangig Füge- und Handhabungsprozesse um. Diese können manuell geführt oder automatisiert gesteuert werden, jeweils mit fester oder freier Bewegungsbahn (VDI 2860, S. 14). Bekanntestes Beispiel sind Industrieroboter, die aufgrund der freien Programmierbarkeit ihrer mechanischen Gelenke eine hohe Universalität bez. der Bewegungsbahn und somit in ihrem Einsatzzweck aufweisen (VDI 2861). Bewegungseinrichtungen werden über einen verbundenen *Endeffektor* auf ihre Aufgabe spezialisiert (DIN EN 8373), bspw. mit einem Greifer oder Schweißbrenner.

Werden für einen Fügevorgang Hilfsmittel benötigt (z. B. Schrauben, Nieten), so werden diese über *Zuführsysteme*, wie etwa Vibrationswendelförderer, bereitgestellt. Zwischen Montagestationen transportieren *Verkettungsmittel* Bauteile oder Baugruppen. Um die zu transportierenden Gegenstände nicht zu beschädigen und eine definierte Fixierung zur Durchführung der Füge- und Kontrollprozesse innerhalb der Montagestationen zu erreichen, wird dabei auf *Transporthilfsmittel* (z. B. Werkstückträger) zurückgegriffen (HESSE & MALISA 2009, S. 498). Hinzu kommen weitere aktorische Systeme, welche im Montageprozess sehr spezifische Aufgaben durchführen und hierfür oftmals speziell konstruiert wurden, bspw. ein Kontaktiersystem zur Durchführung von elektrischen Funktionstests.

Sensoren messen auftretende Prozessinformationen in der Montageanlage. Eine generelle Unterscheidung kann zwischen *merkmalskontrollierenden* Komponenten und *Sicherheitstechnik* getroffen werden. Erstere nehmen eine Vielzahl an auftretenden Größen im Montageprozess auf, etwa die Anwesenheit von Objekten oder deren Position. Messobjekt können hierbei das Bauteil sein oder den Prozess ausführende Komponenten, etwa die aktuelle Einpresskraft einer Bewegungseinrichtung.

Die vorrangige Aufgabe von Sicherheitskomponenten ist das Verhindern von Schäden an Mensch und Material. Unterschied zu merkmalskontrollierenden Sensoren stellt eine sehr ausfallsichere, meist redundante Hardware dar, die über eigene Signalkreise überwacht wird.

Zur Datenverarbeitung und *Prozesssteuerung* werden typischerweise speicherprogrammierbare Steuerungen (SPS) in Montageanlagen eingesetzt. Diese stellen nach FELDMANN & SPUR (2014, S. 624) die am weitesten verbreiteten Automatisierungsgeräte dar und werden in Abschnitt 2.2.2 näher beschrieben. Hinzu kommen *Leitsysteme*, die eine Produktionssteuerung zwischen Montageanlagen und angrenzenden Bereichen, wie Logistiksystemen, übernehmen.

Die vierte Kategorie stellen Komponenten dar, mit denen der Mensch in Interaktion mit dem Montagesystem treten kann (*MMI*). Neben der Kenntlichmachung des Anlagenzustands zählen hierzu Stellelemente, über die der Anwender den Anlagenstatus oder dessen Prozesse manipulieren kann.

Typische automatisierte Montagesysteme sind aus einem Großteil der hier vorgestellten Komponentenarten aufgebaut (vgl. HEINRICH ET AL. 2015, S. 13; FELDMANN & SPUR 2014, S. 622 f.). Von jeder Komponentenart sind meist mehrere bis dutzende Entitäten im Einsatz, oftmals von unterschiedlichen Herstellern.

2.2 Automatisierungstechnik

Die *Automatisierung* beschreibt die Übertragung menschlicher Arbeitsschritte auf selbstständig arbeitende künstliche Systeme, sog. „Automaten“ (VOIGT 2015; IEC 60050, S. 31). *Automatisierte* Prozesse benötigen nur noch teilweise menschliche Arbeitskraft und Entscheidungsfindung, *automatische* Prozesse laufen ohne menschliches Zutun ab („voll-automatisiert“). In der Produktionstechnik werden Automaten durch Automatisierungskomponenten dargestellt, deren Entscheidungen auf Basis eines zuvor

Tabelle 2.1: Komponenten von Montagesystemen. Zusammenfassung der Aufzählungen von HEINRICH ET AL. (2015, S. 13), LAPERRIÈRE & REINHART (2014, S. 1097), FELDMANN & SPUR (2014, S. 623), HESSE & SCHNELL (2014, S. 18), KRUG (2013, S. 52 f.), LOTTER & WIENDAHL (2012, S. 138), CIRP 2011, KLUGE (2011, S. 81 ff.), SECKNER (2008, S. 49).

Hauptklasse	Funktionsklasse	Sub-Funktionsklasse	Beispiel primitives Gerät	Beispiel komplexes Gerät
Aktoren	Bewegungseinrichtungen	-	Pneumatische Zylinder	Industrieroboter
	Endeffektoren	-	Greifer	Schweißbrenner
	Verkettungsmittel	Fördersysteme	Rollenförderer	Gurtsystem mit Haltevorrichtungen
		Transporthilfsmittel	Werkstückträger	-
	Bereitstellungs- und Zuführsysteme	-	Magazine	Vibrationswendelförderer
	Spezifische Systeme	-	Leuchten	Prüfsystem
Sensoren	Merkmalskontrolle	Anwesenheitskontrolle	Induktionssensor	Bildverarbeitungssystem
		Bauteil Identifikation	-	RFID, Barcodescanner
		Eigenschaftenprüfung	Helligkeitssensor	Mikrowellensensor
		Form- und Lageprüfung	Taster	2-D-Kamerasysteme
		Kraft- und Momentenmessung	Dehnungsmessstreifen	Kraft-Momenten-Sensor
Sicherheitstechnik	-	Not-Aus-Schalter	Lasersensor	
Steuerungen	Prozesssteuerungen	-	-	SPS, Bildverarbeitungsrechner
	Leitsysteme	-	-	PCs, Rechenzentren
MMI	Anzeigen	-	Signallampen	Bildschirm
	Steller	-	Schalter	Touchscreen

erstellten Programms die Steuerung und Regelung von Prozessen zum Ziel haben (WELLENREUTHER & ZASTROW 2008, S. 1).

Historisch bedingt lassen sich zwei Arten von Automatisierungskomponenten unterscheiden: *Feldgeräte* (Sensoren und Aktoren), die direkt mit dem Prozess in Verbindung stehen, jedoch nur auf externe Anweisung ihre Aufgabe durchführen. Ihr Gegenstück sind dezidierte *Steuerungsrechner*, die rein zur Datenverarbeitung eingesetzt werden. Dies beinhaltet auch die Kommunikation mit Feldgeräten und überlagerten IT-Systemen (vgl. KLASSEN ET AL. 2010, S. 206).

2.2.1 Industrielle Kommunikation

Kommunikationsarten

Die Grundlage für einen Informationsaustausch bildet die Kommunikation. Im einfachsten Fall tauschen zwei Teilnehmer Daten über eine direkte Verbindung untereinander aus. *Daten* stellen hierbei die real übertragenen Größen dar (bspw. Wörter, Messwerte), *Informationen* sind eine durch Semantik und Kontext beeinflusste Interpretation der Daten (vgl. GORECKY ET AL. 2017, S. 558 f.). *Kommunikation* ist somit der Ablauf und die Interpretation der Datenübertragung.

Daten besitzen unterschiedliche *Typen*: binär (z. B. an/aus), numerisch (z. B. 3,14) oder alphanumerisch (z. B. „grün“). Die *Datenübertragung* kann hierbei analog oder digital erfolgen. *Analog*signale sind wertkontinuierlich, d. h., jeder Wert innerhalb eines Wertebereichs kann angenommen werden, in der Automatisierungstechnik typischerweise zwischen 0 – 24 V. *Digital*signale sind wertdiskret und damit prinzipiell ungenauer, jedoch auch robuster gegenüber äußeren Störungen. Sollen analoge Werte digital übertragen werden, nimmt ein Analog-Digital-Wandler die Umrechnung vor. (HESSE & SCHNELL 2014, S. 5 ff.; UFFELMANN & WIENZEK 2010, S. 34)

Werden mehrere zusammengehörige Daten übertragen, so muss mit der Gegenseite Einigkeit über die Formalitäten herrschen (bspw. Länge des Datensatzes, Reihenfolge der Einträge). Dieser „Vertrag“ wird in der Kommunikationstechnik als *Protokoll* bezeichnet. Bedingt durch die (heutige) binäre Grundlage der Datenverarbeitung, erfolgt auch die Kommunikation in der Automatisierungstechnik oftmals binär. Hierbei werden die zu übertragenden Daten binarisiert und als Binärwerte (z. B. 0 V oder 24 V) innerhalb eines Protokolls übermittelt.

Werden mehr als zwei Teilnehmer direkt miteinander verbunden, so entsteht ein vermaschtes Netz. Bei n vollständig untereinander verbundenen Teilnehmern muss jeder Teilnehmer $(n - 1)$ Schnittstellen bereitstellen und es werden $\frac{n(n-1)}{2}$ Kommunikationsleitungen benötigt. Bei 20 Geräten entspricht dies 19 Schnittstellen pro Gerät sowie 190 Kommunikationsleitungen. Das Beispiel macht den Nachteil direkter Verbindungen deutlich: erhebliche Investitionskosten, bedingt durch große Installationsaufwände und Leitungslängen. Vorteile sind eine gleichmäßige Kommunikationslastverteilung sowie eine hohe Ausfallsicherheit, falls Nachrichten von Teilnehmern weitergeleitet werden können. (SCHNELL & WIEDEMANN 2006, S. 1 ff.)

Mit dem Ansatz der *Bus-Kommunikation* wurde ab den 1970er-Jahren eine Alternative geschaffen, bei der Teilnehmer nunmehr über eine gemeinsame Leitung kommunizieren. Im Falle einer sequentiellen Verschaltung (Linienstruktur) werden bei n Teilnehmern $n - 1$ Leitungen benötigt. Neue Teilnehmer können unabhängig von den bisherigen Teilnehmern integriert oder entfernt werden. Hierdurch sinkt der physische Installationsaufwand und es steigt die Wandelbarkeit des Netzwerks. (REISSENWEBER 2009, S. 25)

Nachdem sich mehrere Teilnehmer eine Leitung teilen, steigt bei Bussystemen allerdings die Komplexität der Kommunikation: Damit Nachrichten dem gewünschten Adressaten übermittelt werden können, müssen sich Teilnehmer eindeutig identifizieren können. Um Kollisionen oder Verzögerungen zu vermeiden, muss außerdem der Zugriff auf das gemeinsame Busmedium geregelt werden. (SCHNELL & WIEDEMANN 2006, S. 3 ff.)

Der Buszugriff kann in kontrollierte oder zufällige Verfahren unterschieden werden. Bei *kontrollierten Verfahren* wird vorab bestimmt, welcher Teilnehmer Daten senden darf. Hierzu existieren Master/Slave-Ansätze, bei denen ein Kommunikationsverantwortlicher (Master, z. B. SPS) den Teilnehmern (Slaves, z. B. Sensoren) die Sendeaufforderung zuteilt, oder Token-Verfahren, bei denen die Sendeaufforderung als Marke zwischen den Teilnehmern wandert. Durch die Festlegung wiederkehrender Sendezeiträume kann eine maximale Übertragungs- und Antwortzeit berechnet werden (vgl. SCHNELL & WIEDEMANN 2006, S. 97 ff.). Hierdurch verhält sich die Kommunikation der Teilnehmer *deterministisch*, was für viele Prozesse in der Automatisierungstechnik unabdingbar ist, etwa für Sicherheitstechnik oder synchronisierte Antriebe. Deterministische Kommunikation wird auch als *echtzeitfähige* Kommunikation bezeichnet.

Bei *zufälligen Verfahren* entscheiden die Teilnehmer selbst, wann sie senden möchten. Hierbei muss sichergestellt sein, dass im Falle von zwei gleichzeitig sendenden Teilnehmern Kollisionen erkannt und verhindert werden: In Variante 1 warten beide Teilnehmer jeweils eine zufällige Zeitspanne und senden dann erneut, in Variante 2 existiert eine manuell festgelegte Priorisierung, bei der der Unterlegene die Kommunikation bis zum Freiwerden des Übertragungsmediums einstellt. Bei beiden Varianten ist nicht sichergestellt, wann ein Teilnehmer seine Nachricht absetzen kann. *Ethernet* als bekanntester Vertreter von Variante 1 ist deshalb nicht echtzeitfähig und nur bedingt für zeitbestimmte Produktionsprozesse einsetzbar. (SCHNELL & WIEDEMANN 2006, S. 19 ff.)

Kommunikationsprotokolle

In den Siebzigerjahren wurde durch die International Standards Organization (ISO) das Referenzmodell zur Rechnerkommunikation *Open Systems Interconnection (OSI)* spezifiziert, das bis heute Grundlage der bedeutendsten Kommunikationsstandards ist. Das Referenzmodell definiert sieben abstrakte Schichten mit jeweils eigenen Aufgaben (vgl. Abbildung 2.6a). Jede Schicht nutzt dabei angebotene Funktionalitäten unterlagerter Schichten und bietet ihrerseits Dienste an höhere Schichten an. Kommunizieren zwei Teilnehmer untereinander, so nutzt die Sender-Applikation die Dienste der Anwendungsschicht (7). Diese verwendet wiederum Dienste der unterlagerten Schicht und so fort. Jede Schicht fügt hierbei protokollspezifische Informationen hinzu (z. B. Prüfsummen) und gibt das kombinierte Paket an die unterlagerte Schicht weiter, bis es auf Schicht 1 physikalisch übertragen wird. Auf Empfängerseite werden die Schichten in umgekehrter Reihenfolge durchlaufen und das Paket entsprechend wieder zerlegt, bis es von der Empfängerapplikation entgegengenommen wird. Für eine korrekte Kommunikation müssen Teilnehmer auf allen Schichten identische Protokolle nutzen. (REISSENWEBER 2009, S. 37 ff.; SCHNELL & WIEDEMANN 2006, S. 8 ff.)

Als eines der ersten Protokolle wurde *Ethernet*³ Ende der Siebzigerjahre entwickelt. In Kombination mit *Transmission Control Protocol (TCP)* sowie dem *Internet Protocol (IP)* stellt dieser Protokollverbund die Grundlage heutiger digitaler Kommunikation dar (vgl. Abbildung 2.6b, links).

Durch den nicht-deterministischen Aufbau des Ethernet (vgl. Abschnitt 2.2.1) entstanden in den Neunzigerjahren verschiedene Feldbus-Standards, die mit eigenen Protokollen zur echtzeitfähigen Kommunikation den Anforderungen der Fabrikautomation besser gerecht wurden. Im ISO-OSI-Modell besitzen diese nur die Schichten 1, 2 und 7, Aufgaben der Vermittlung (Schicht 3) oder Transportsteuerung (Schicht 4) müssen innerhalb der Anwendungsschicht gelöst werden oder entfallen gänzlich (vgl. Abbildung 2.6b, Mitte) (REISSENWEBER 2009, S. 42 ff.). Feldbusse sind demnach inkompatibel zu Ethernet bzw. TCP/IP. Im Zuge der stetig wachsenden Unternehmensvernetzung wurde dies zu einem zunehmenden Nachteil, welchem mit der Entwicklung von echtzeitfähigen Ethernet-Protokollen (engl. Real-time Ethernet, RTE) Abhilfe geschaffen werden sollte. Hierbei entwickelten sich unterschiedliche Verfahren, die Modifikationen

³Name abgeleitet von „Äther“, dem omnipräsenten Trägermedium (METCALFE 1999).

OSI-Schicht	Aufgabe	Internet-Protokolle		Feldbusse	Real-time-Ethernet (RTE)			
		TCP/IP	Beispiel		Powerlink	Profinet	EtherCAT	
(7) Anwendungen	• Schnittstelle für Applikationen	Anwendungen	HTTP, FTP, OPC UA	(7) Anwendung	modifiziert	modifiziert	modifiziert	
(6) Darstellung	• Zeichencodierung • Optische Darstellung		Transport					TCP, UDP
(5) Sitzung	• Nutzung der Transportkanäle							
(4) Transport	• Flusskontrolle • Zerlegung Daten	Internet	IP	(2)	(2)	(2)	modifiziert	
(3) Netzwerk	• Adressierung • Routing	Netz-zugang	Ethernet	(1)	(1)	(1)		
(2) Verbindung	• Datenübertragung • Buszugriff							
(1) Übertragung	• Phys. Übertragung							

(a) ISO-OSI-Modell

(b) Bekannte Beispiele (EPSG 2016; SCHNELL & WIEDEMANN 2006, S. 8 ff.; REISSENWEBER 2009, S. 328)

Abbildung 2.6: Schichtenaufbau von Protokollen nach ISO/IEC 7498

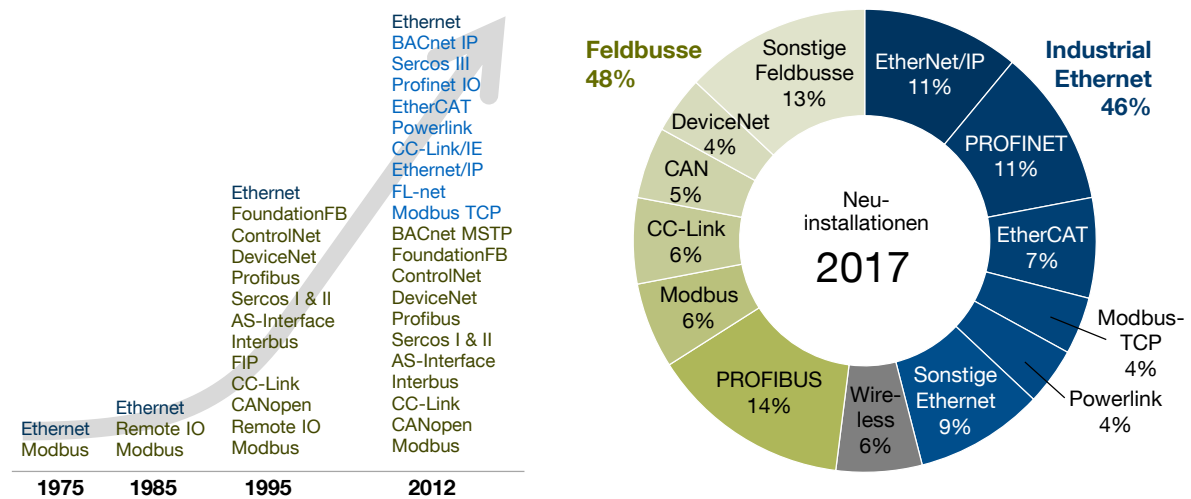
innerhalb der einzelnen Schichten des TCP/IP-Modells einführen (vgl. Abbildung 2.6b, rechts). Die hieraus entstandenen Varianten unterscheiden sich u. a. in der minimalen Antwortzeit und dem möglichen Datendurchsatz.

Gegenüber traditionellen Feldbussen besitzen RTE-Protokolle den Vorteil, dass parallel zu den deterministisch zu verschickenden Prozessdaten auch nicht-deterministische Daten über die Standard IP-Protokolle verschickt werden können (KLASEN ET AL. 2010; REISSENWEBER 2009). Das ermöglicht eine leichtere Inbetriebnahme im Feld sowie eine einfachere Anbindung von Automatisierungsgeräten an die Unternehmensnetzwerke (z. B. MES). Erreicht wird dies entweder durch eine priorisierte Behandlung der echtzeitfähigen Daten oder durch ein Zeitschlitzverfahren, bei dem echtzeitfähige und nicht-echtzeitfähige Daten abwechselnd gesendet werden (KLASEN ET AL. 2010; REISSENWEBER 2009). Einen weiteren Vorteil stellt die wesentlich höhere Übertragungsrates dar (SCHNELL & WIEDEMANN 2006, S. 315), welche mit zunehmendem Datenaustausch stetig an Bedeutung gewinnt.

Innerhalb der deterministischen Datenübertragung sind sämtliche automatisierungstechnischen Kommunikationsprotokolle untereinander inkompatibel. Hinter den einzelnen Kommunikationsstandards stehen meist Unternehmen (bspw. SIEMENS bei PROFIBUS und PROFINET), welche ihre Lösungen bevorzugt vermarkten. Die Entscheidung für eine Kommunikationslösung geht demnach oft mit der Entscheidung für ein Unternehmen oder einen Hersteller und dessen Komponentenportfolio einher. Hierdurch entstehen Insellösungen, welche eine Interoperabilität zwischen Automatisierungsgerä-

ten hemmen. Dies senkt die mögliche Rekonfigurationsfähigkeit von Montagesystemen. (KLEINEBERG ET AL. 2016, S. 8; EPSG 2016; BMBF 2015)

Aufgrund der mangelnden Konsolidierung existieren mittlerweile über 20 kabelgebundene Standards am Markt (Abbildung 2.7a). Diese Fragmentierung zeigt sich noch immer bei Neuinstallationen: Im Jahr 2017 ergeben sich Marktanteile von meist einstelligen Prozentbereichen (Abbildung 2.7b). Nach SPINNARKE (2016) legt dies die Schlussfolgerung nahe, dass es aufgrund zu heterogener industrieller Anforderungen *den einen* Idealbus nicht gibt. Bedingt durch oben genannte Vorteile stellen RTE-Protokolle die bevorzugte Wahl für neu zu installierende Kommunikationsnetzwerke dar, was sich in einem höheren jährlichen Wachstum (22 % gegenüber 4 %) bemerkbar macht (HERKOMMER 2017). Dennoch weisen traditionelle Feldbusse nach wie vor eine größere Gesamtinstallationsbasis auf, bedingt durch die lange Lebensdauer von Industrieanlagen sowie die schwierige Aufrüstung existierender Kommunikationstechnologien (BMBF 2015; REISSENWEBER 2009, S. 53). Wird die Initialschätzung von LYDON (2013) mit den Studien von HMS Industrial Networks (HERKOMMER 2017; VOLZ 2016) extrapoliert, so existieren heute über 200 Millionen per Bus-vernetzte Automatisierungskomponenten.



(a) Zeitliche Entwicklung (in Anlehnung an KLASSEN ET AL. 2010, S. 9) (b) Marktanteile der Neuinstallationen 2017, Schätzung von HMS Industrial Networks (HERKOMMER 2017)

Abbildung 2.7: Industrielle Busprotokolle

Hierarchieebenen

Aufgrund der in Abschnitt 2.2.1 beschriebenen Historie haben sich unterschiedliche Protokolle auf verschiedenen Hierarchieebenen in Unternehmen etabliert. Die bereichsübergreifende Vernetzung basiert auf Ethernet und wird als *IT-Ebene* (eng. Information Technology) abgekürzt. Die prozessnahe Kommunikation – meist deterministisch ausgelegt – wird als *OT-Ebene* (Operations Technology) bezeichnet (SACHSE 2017, S. 10).

Prozesseitig werden eine Vielzahl von Sensoren und Aktoren mit deren steuernden Komponenten (SPS, Industrie-PC) vernetzt. Die Gruppierung von Steuerungsaufgaben geschieht hierbei meist auf der Stations- oder Arbeitssystemebene. Zur Abbildung von Produktvarianten werden die Steuerungsprogramme parametrisiert erstellt oder pro Variante ein eigenes Programm hinterlegt. Die Programmwahl geschieht über einzelne Leitreechner des MES, welche wiederum deren Aufträge durch ein zentrales Unternehmensplanungssystem (ERP, Enterprise Resource Planning) beziehen. (FRIEDL 2014, S. 622)

Wird die Kommunikationstopologie betrachtet, so ergibt sich eine nach oben hin schmaler werdende Dreiecksstruktur, welche im Allgemeinen als *Automatisierungspyramide* bezeichnet wird (Abbildung 2.8a).

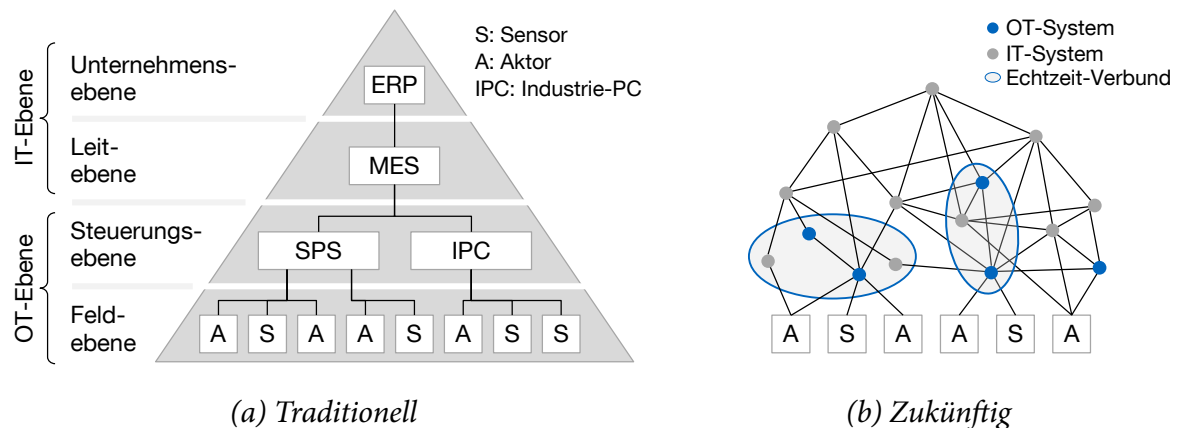


Abbildung 2.8: Kommunikationshierarchien in der Automatisierung (erweiterte Darstellung in Anlehnung an GORECKY ET AL. 2017, S. 556 und WGP 2016 S. 11)

Durch den aktuell steigenden Vernetzungsgrad nimmt der Kommunikations- und Datenverwaltungsumfang kontinuierlich zu. Aus diesen Grund existiert ein Trend weg von monolithischen Steuerungsarchitekturen hin zu dezentraleren Systemen, bei denen mehr Steuerungsverbünde existieren, diese jedoch kleinere Aufgaben übernehmen.

Hinzu kommen verstärkt Komponenten, die ihre Daten – parallel zur Steuerung – direkt an IT-Systeme schicken, etwa zur zentralen Datenanalyse. Hierdurch müssen Informationen nicht mehr streng hierarchisch ausgetauscht werden, wodurch sich die klassische Automatisierungspyramide hin zu einem Cluster von Subnetzen wandelt, welche jeweils bestimmte Eigenschaften und Funktionalitäten besitzen (Abbildung 2.8b). (WGP 2016 S. 11 f.)

Datenaustausch zwischen der OT- und IT-Ebene

Die Automatisierungstechnik ist von proprietären Standards geprägt (BMBF 2015), so dass nur wenig herstellerübergreifende Lösungen zum Datenaustausch zwischen der OT- und IT-Ebene existieren. Abhilfe schafft das 2009 spezifizierte OPC UA⁴-Protokoll (IEC 62541), welches – beflügelt durch den Trend zur steigenden Vernetzung – mittlerweile zu einem De-facto-Standard des Datenaustauschs von IT-Systemen mit automatisierungstechnischen Komponenten geworden ist. (WGP 2016, S. 264; PLATTFORM I4.0 2016b, S. 6)

Im ISO-OSI-Modell stellt OPC UA eine Anwendungsschicht dar (Abschnitt 2.2.1) und ist somit prinzipiell unabhängig von einem Transportprotokoll spezifiziert. Der Großteil am Markt verfügbarer Komponenten basiert jedoch auf Ethernet/TCP-Implementierungen. (LANGE ET AL. 2010; OPC 2017)

Die Kernidee von OPC UA lautet, dass Geräte ihren Zustand und Attribute (z. B. Prozessgrößen) über einen OPC UA-Server im Netzwerk publizieren. Die Informationen stehen hierbei in einem sog. *Informationsmodell* zur Verfügung. Dieses erlaubt eine objektorientierte Strukturierung von Informationen, d. h., es können Entitäten mit zugehörigen Eigenschaften und Methoden angelegt werden. Klienten können das Informationsmodell „erkunden“ und bei Bedarf Informationen lesen, schreiben oder sich bei Veränderungen benachrichtigen lassen. Hierfür stehen definierte OPC UA-Funktionalitäten zur Verfügung.

OPC UA wird von der „OPC Foundation“ getrieben, welche auch ein Meta-Modell zur Definition eigener Informationsmodelle bereitstellt. Abbildung 2.9 zeigt hierzu das exemplarisch erstellte Modell eines Industrieroboters.

⁴Open Platform Communications - Unified Architecture

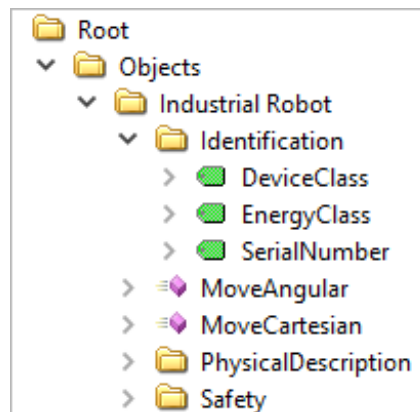


Abbildung 2.9: Beispielhafte Datenpublikation eines OPC UA-Servers. Grüne Symbole stellen Attribute dar, purpur-farbige Methoden.

Darüber hinaus können sog. „Begleitstandards“ (engl. „Companion Specifications“) geschaffen werden, welche das Informationsmodell von einer bestimmten Gerätekategorie einheitlich definieren. Unternehmen, die diese Spezifikation unterstützen, schaffen so eine interoperable Schnittstelle zu dem jeweiligen Gerät und reduzieren somit den Entwicklungsaufwand für verbundene Softwareapplikation, wie etwa MES-Applikationen. Ein bekanntes Beispiel ist der Begleitstandard zur Darstellung von IEC 61131-3 Steuerungssystemen. (BAUERNHANSL ET AL. 2014, S. 329 ff.; ENSTE & MAHNKE 2011, S. 402)

Welche Variablen und Funktionen durch den OPC UA-Server publiziert werden, legt der Anwender im Rahmen des Steuerungsengeinerings fest. Hinzu kommen weitere Server-Einstellungen, wie etwa die zu erfüllenden Autorisierungsmechanismen.

Ablauf der Geräteintegration

Während Geräte im Consumerbereich seit vielen Jahrzehnten mittels Plug&Play-Technologien automatisiert vernetzt werden (beispielsweise über den Universal Serial Bus, USB), erfolgt die Verbindung von Automatisierungskomponenten bis heute nur teil-automatisiert. Der Prozess umfasst nach HAMMERSTINGL & REINHART (2015b, S. 1957), HODEK (2013, S. 12 ff.) und REINHART & KRUG (2012, S. 149 f.) mehrere Schritte, welche nachfolgend konsolidiert beschrieben werden (Abbildung 2.10).

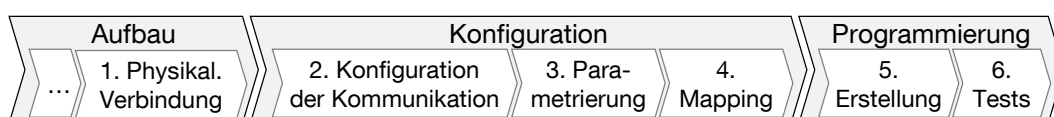


Abbildung 2.10: Ablauf der Geräteintegration in Steuerungssysteme

1. *Physische Verbindung* – Zur Datenübertragung (Schicht 1 im ISO/OSI-Modell, vgl. Abbildung 2.6a) muss eine entsprechende Verbindung zwischen den Kommunikationsteilnehmern geschaffen werden. Für einfache Geräte kann dies eine direkte elektrische Verdrahtung sein, komplexere Geräte besitzen hierzu modulare Steckverbindungen, wie etwa RJ45-Buchsen bei ethernetbasierten Protokollen.

2. *Konfiguration der Kommunikation* – In diesem Schritt wird die Netzwerkkonfiguration festgelegt. Diese besteht aus drei Bestandteilen:

- *Vorbereitung des Kommunikationsmasters*: Initial müssen mit dem Kommunikationsmaster eine Verbindung hergestellt und dessen grundlegenden Hardwarespezifika eingerichtet werden. Dies geschieht über umfangreiche Engineeringtools, die Steuerungshersteller zusammen mit ihren Geräten vertreiben. Bekannte Beispiele sind etwa das „TIA-Portal“ (SIEMENS) oder „Automation Studio“ (B&R).
- Innerhalb der *Geräteerkennung* wird dem Master mitgeteilt, welche abzufragenden Slaves existieren. Hierfür wird im Engineeringtool die real vorliegende Netzwerktopologie nachmodelliert. Hauptaktivitäten sind die Auswahl des Kommunikationsprotokolls, dessen Grundkonfiguration (z. B. Übertragungsrate) und die Festlegung der Busadressen abzufragender Slaves.
- Innerhalb der *Signalcodierung* wird definiert, wie die zu übertragende Datenstruktur zwischen Master und Slave aussieht. Festzulegende Charakteristika sind etwa der Name, der Datentyp, die Bitaufteilung und das Zeitverhalten. Um den Vorgang zu erleichtern, stellen Hersteller von Automatisierungskomponenten umfangreiche Gerätebeschreibungsdateien zur Verfügung, in denen die Struktur der zu übertragenden Informationen bereits gespeichert ist. Diese sind hardware- und protokollabhängig, sodass der Anwender die richtige Beschreibungsdatei für die aktuell verwendete Technologiekombination auswählen und in das jeweilige Engineeringtool importieren muss. Abbildung 2.11 zeigt hierzu ein Beispiel.

```
1 ; Allgemeine Angaben
2 Vendor_Name       = "XYZ"
3 Model_Name       = "ABC123"
4 Ident_Number     = 0x059E
5
6 ; Slave-Eigenschaften
7 Sync_Mode_supp   = 1
8 Min_Slave_Intervall = 0x0005
9 Modular_Station = 1; Gerät hat Submodule
10 Max_Module      = 0x30
11 Max_Output_Len  = 0x40
12 ; Definition der Submoduldaten
13 Module = "VTSA [8D0]" 0x10
14 1
15 Ext_User_Prm_Data(0) = 0x02, 0x03
16 Ext_User_Prm_Data_Ref(2) = 12
17 EndModule
18 Module = "VTSB [4D0]" 0x20
19 2
20 Ext_User_Prm_Data(0) = 0x45, 0x81
21 Ext_User_Prm_Data_Ref(2) = 17
22 EndModule
```

Abbildung 2.11: Exemplarische Gerätebeschreibungsdatei für PROFIBUS (Auszug)

3. *Parametrierung*: Grundsätzlich besitzen Automatisierungsgeräte eine bestimmte Universalität in ihrem Einsatzbereich (z. B. Antriebe), weshalb sie für die Anforderungen des aktuellen Einsatzzwecks eingerichtet werden müssen. Bei Sensoren ist dies beispielsweise das Ansprechverhalten, bei Aktoren der Verfahrweg oder die Verfahrgeschwindigkeit. Die Parametrierung wird entweder direkt am Gerät vorgenommen (z. B. über physische Schalter) oder über einen Rechner eingestellt. Dies geschieht entweder über das Engineeringtool der Steuerung, spezifische Software des Automatisierungstechnik-Herstellers oder übergreifende Parametrierungsstandards wie FDT (Field Device Tool, vgl. IEC 62453).

4. *Mapping*: In dieser Phase verknüpft der Programmiersteller die Prozessdaten, welche an den physischen Ein-/Ausgängen der Steuerung anliegen, mit den erstellten Variablen im Steuerungsprogramm. Ändert sich eine Variable, so wird diese Änderung mittels der in Schritt 3 angelegten Prozessvariablen auf das Gerät übertragen – und umgekehrt.

5. *Programmierstellung*: Auf Basis von logischen Anweisungen wird der Prozessablauf des Automatisierungssystems erstellt (vgl. Abschnitt 2.2.2). Je nach Digitalisierungsgrad des Unternehmens kann die Programmierung auch bereits in früheren Projektphasen stattfinden. Dann wird auf Simulationsmodelle des aufzubauenden Systems zurückgegriffen.

6. *Tests*: Im letzten Schritt wird der erstellte Softwareablauf im Zusammenspiel mit den realen Komponenten getestet. Für Programmierung, Kompilierung, Upload und Fehlersuche nutzt der Anwender erneut das Engineeringtool des jeweiligen Steuerungsherstellers.

2.2.2 Steuerungsgeräte

Neben SPSen als Standardkomponenten zur Prozesssteuerung entwickelten sich für die komplexen Anwendungsbereiche der Robotik und Bildverarbeitung spezialisiertere Systeme, sodass heute drei große Steuerungsdomänen in der Montage existieren. Alle drei zeichnet eine gewisse Universalität aus, bedingt durch Hardware-Modularität und „freie“ Programmierbarkeit, wodurch sich ein breites Spektrum von Produktionsprozessen automatisieren lässt.

Für die vorliegende Arbeit sind ein grundlegendes Verständnis für Aufbau und Programmierung der Systeme notwendig, weshalb diese hier kurz vorgestellt werden. Eine

umfassende Erläuterung ist der jeweiligen Fachliteratur zu entnehmen (z. B. JOHN & TIEGELKAMP 2009, WEBER 2009 und DEMANT ET AL. 2011).

Speicherprogrammierbare Steuerungen (SPS)

Speicherprogrammierbare Steuerungen besitzen einen ähnlichen Hardware-Aufbau wie herkömmliche Computer. Im Unterschied zu Letzteren ist die Hardware wesentlich langlebiger und robuster gegenüber Umgebungseinflüssen ausgelegt. Für den Anschluss einfacher Peripheriegeräte (z. B. Lichtschranken) existiert ein modulares Steckkartensystem, über das sich Analog- oder Digitalsignale übertragen lassen (Abschnitt 2.2.1). Sollen weiter entfernte Peripheriegeräte angesteuert werden, können diese mit sogenannten „dezentralen IO-Baugruppen“ (auch „Buskoppler“) verbunden werden, welche das Peripheriesignal in eine Bus-Nachricht übersetzen. Dezentrale Baugruppen werden zusammen mit weiteren Feldgeräten über die Feldbus- oder RTE-Schnittstelle der SPS angesteuert. Darüber hinaus besitzen SPSen einen oder mehrere Ethernet-Anschlüsse zur nicht-deterministischen Kommunikation mit Geräten aus der OT- oder IT-Ebene, bspw. mit Leitrechnern (vgl. Abbildung 2.12). Die aktuelle Zusammenstellung der Hardware und Schnittstellenmodule muss je nach Hersteller in dessen Engineeringtool durch den Anwender spezifiziert werden oder kann teilautomatisiert durch die Software identifiziert werden. (WELLENREUTHER & ZASTROW 2008)

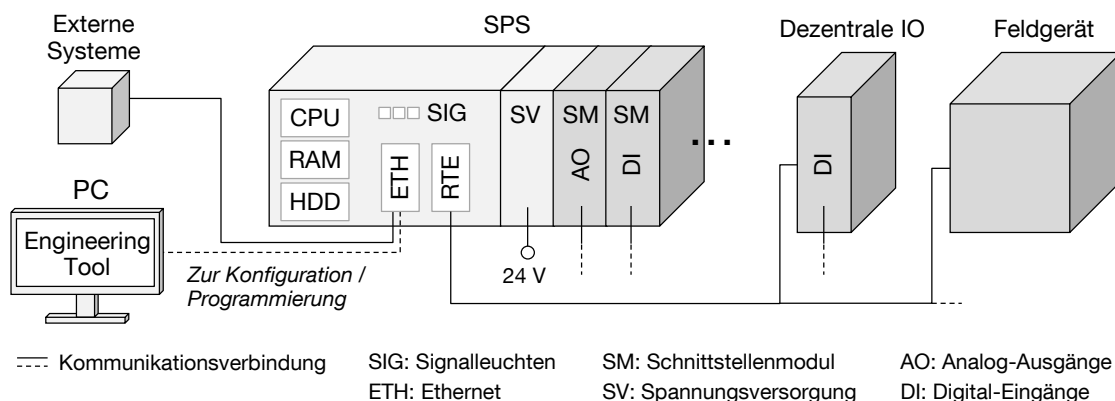


Abbildung 2.12: Generelle Hardwarestruktur von SPSen

Ein wichtiger Aspekt bei SPSen ist deren Echtzeitfähigkeit, d. h. die deterministische Verarbeitung von Programmen. Nur in Kombination mit deterministischen Kommunikationsprotokollen lassen sich synchronisierte oder sicherheitskritische Aktoren steuern. Programme laufen bei SPSen grundsätzlich wiederholend (zyklisch) ab, weshalb wäh-

rend der Erstellung eine Zykluszeit zugewiesen werden muss (z. B. 50 ms). (FRÜH & AHRENS 2009, S. 205 ff.)

Die Programmierung von SPSen ist in der DIN 61131-3 standardisiert. Diese beschreibt fünf Programmiersprachen: Historisch bedingt existieren mit der *Anweisungsliste (AWL)* eine Assembler-nahe Sprache sowie zwei grafische Programmiersprachen, welche elektrischen Schaltplänen ähneln: Der *Kontaktplan (KOP)* und die *Funktionsbausteinsprache (FBS)*. Im Zuge der großen Verbreitung von C-basierten Hochsprachen (z. B. C++) finden diese jedoch immer seltener Anwendung. Häufiger wird *Strukturierter Text (ST)* verwendet, welcher von seiner Syntax und seinem Umfang der Sprache C ähnelt. Zur Modellierung von Prozessabläufen mit definierten Weiterschaltbedingungen (Zustandsautomaten) wird auf die *Ablaufsprache (AS)* zurückgegriffen, welche einzelne Programmbausteine grafisch verknüpft. Abbildung 2.13 zeigt die vorgestellten Programmiersprachen in der Übersicht. (JOHN & TIEGELKAMP 2009)

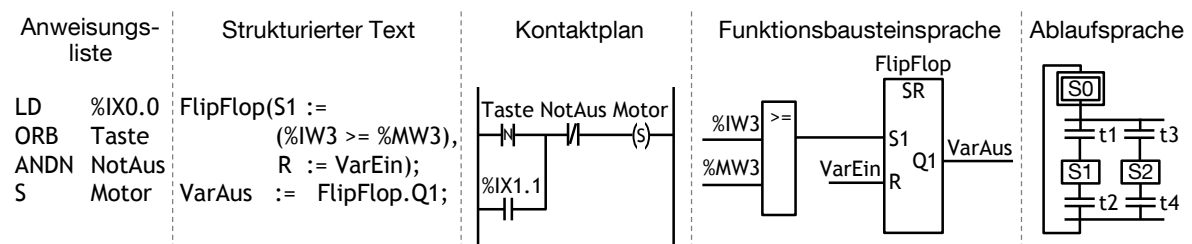


Abbildung 2.13: SPS-Programmiersprachen der DIN 61131-3, in Anlehnung an JOHN & TIEGELKAMP (2009, S. 24f.)

Industrieroboter-Steuerungen (IRS)

Der Hardwareaufbau von Industrieroboter-Steuerungen ähnelt denen von SPSen (vgl. SICILIANO & KHATIB 2008, S. 982). Steckkartensysteme sind weniger ausgeprägt, weshalb die Ansteuerung einfacher Geräte über dezentrale IOs gelöst wird, bspw. bei Druckluftgreifern. Die Kommunikation mit komplexeren Feldgeräten geschieht über Feldbus bzw. RTE-Schnittstellen, zusätzlich ist meist ein Ethernet-Anschluss verbaut, der den Datenaustausch mit IT-Systemen ermöglicht (Abbildung 2.14). Industrieroboter besitzen herstellereigenspezifische Handgeräte (sog. „Teach-Panels“) und Betriebssysteme zur Konfiguration und Programmierung. Hierdurch ist eine Einarbeitung in das jeweils verwendete System notwendig.

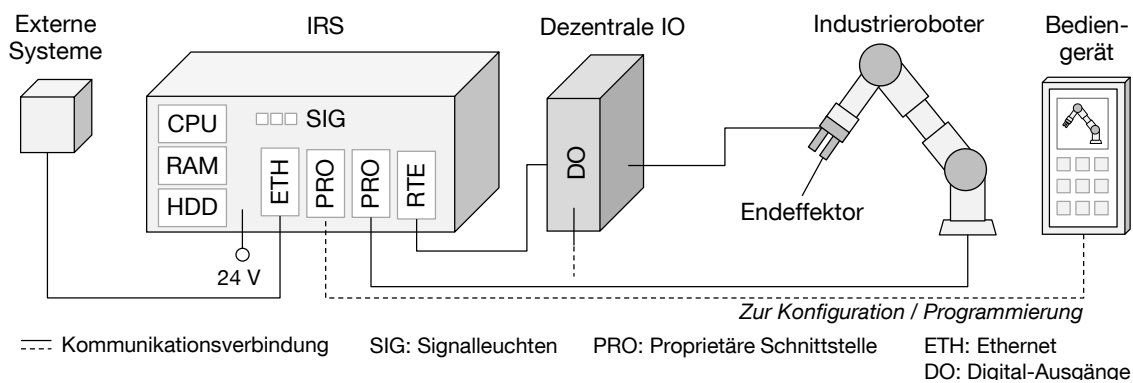


Abbildung 2.14: Generelle Hardwarestruktur von Industrierobotern

Die Programmstruktur von Industrierobotersteuerungen ähnelt denen anderer Softwareapplikationen. Eine Besonderheit sind jedoch spezielle Verfahrbefehle, um den Roboter in eine spezifische Pose in dessen Arbeitsbereich zu bewegen. Eine *Pose* stellt die Kombination aus Position und Orientierung im Raum dar (DIN EN 8373). Die jeweiligen Verfahrparameter und deren Bezugskoordinatensysteme müssen durch den Anwender spezifiziert werden, u. a. Geschwindigkeit, Beschleunigung, Bremsverhalten und Bewegungsart (z. B. linear, zirkular, zeit-/energieoptimal). Innerhalb des Robotersystems berechnet eine Bewegungssteuerung aus den gegebenen Größen die Sollverläufe der Gelenkbewegungen. (WEBER 2009, S. 21 ff.)

Allgemein existieren fünf Grundarten zur Programmierung von Industrierobotern, die in Tabelle 2.2 aufgelistet sind.

Tabelle 2.2: Programmierarten von Industrierobotern nach HESSE & MALISA (2009, S. 74f.)

Direkte Verfahren ⁵		Indirekte Verfahren ⁵		
<i>Teach-in</i>	<i>Playback</i>	<i>Textbasiert</i>	<i>Simulationsbasiert</i>	<i>Aufgabenorientiert</i>
Über ein Handbedien- gerät verfährt der Anwender den realen Ro- boter an die gewünschten Punkte und speichert die- se sukzessive ab.	Der An- wender berührt direkt den Ro- boterarm und führt diesen zu den Prozess- orten.	Das Programm wird in einem Textedi- tor geschrieben und anschließend in die reale Steuerung im- portiert. Die Posen des Roboters wer- den als numerische Koordinaten in das Programm eingetra- gen.	In einer 3D-Kinematik- simulation werden das Robotermodell und des- sen Umgebung nachge- bildet. Der Roboter wird vom Anwender in der Si- mulation verfahren und dessen Posen abgespei- chert. Im Nachgang wird das erstellte Programm in die IRS importiert.	Der Anwender spezifiziert in einem 3D-Simulations- tool die Aufgabe, bspw. Bauteil und gewünschte Lackdicke. Über ein hin- terlegtes Softwaremodell werden die notwendigen Posen und Verfahrwege berechnet. Der Import er- folgt analog zu simulati- onsbasierten Verfahren.

In der industriellen Anwendung können hierbei Mischformen auftreten, vgl. UNIVERSAL ROBOTS (2017, S. II-39). So müssen simulationsbasiert erstellte Programme etwa häufig im realen Prozess über Teach-in-Verfahren „nachbearbeitet“ werden.

Es existiert keine übergreifend standardisierte Sprache zur Programmierung von Industrierobotern. Gepaart mit dem spezifischen Aufbau der Programmierumgebungen ist somit herstellerepezifisches Wissen beim Anwender erforderlich. Durch das *Robot Operating System (ROS)* (ROS 2018) ist zwar eine gewisse Standardisierung in der Softwareentwicklung für innovative Roboteranwendungen zu erkennen, der Markt kommerziell verfügbarer Industrieroboter ist jedoch noch stark proprietär geprägt.

Bildverarbeitungssysteme (BVS)

Aufgrund der benötigten hohen Rechenleistung werden Bildverarbeitungsoperationen in der Automatisierungstechnik meist auf industriellen PCs (IPCs) durchgeführt. Die Programmierung kann direkt auf dem IPC erfolgen oder wird auf einem externen Computer erstellt und im Anschluss übertragen. Die für eine Bildaufnahme notwendigen Kamerasysteme sind über verbreitete Schnittstellen wie USB oder Ethernet angeschlossen. Darüber hinaus existieren proprietäre Standards wie „CameraLink“, für die spezielle Schnittstellenkarten benötigt werden. Neben der Kameraeinheit mit Bildsensor wird ein auf die Aufgabe abgestimmtes Objektiv benötigt. Zusätzlich ist eine gleichbleibende Beleuchtung für eine stabile Bildauswertung wichtig, weshalb häufig externe Leuchten oder Blitzlichter zum Einsatz kommen. Diese werden meist über einen Digitalausgang der Kamera angesteuert (Abbildung 2.15). (DEMANT ET AL. 2011, S. 6 ff.)

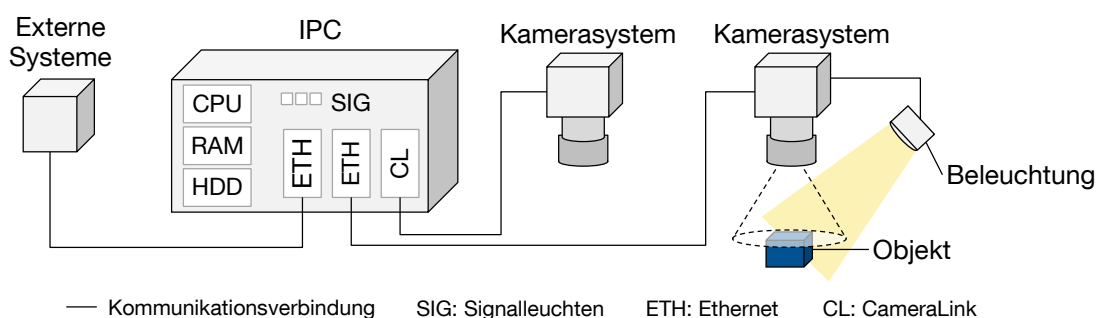


Abbildung 2.15: Genereller Aufbau von Bildverarbeitungssystemen. Erweiterte Darstellung in Anlehnung an DEMANT ET AL. (2011, S. 7).

⁵Alternative Bezeichnungen für direkte und indirekte Verfahren lauten *Online-* und *Offline-*Verfahren.

In der Bildverarbeitung sollen relevante Merkmale aus 2D- oder 3D-Bildern gelesen werden, wie etwa Lagebeziehungen oder morphologische Größen. Die Softwareentwicklung unterscheidet sich hierbei zu den vorherigen Domänen: So steht weniger eine umfangreiche Ablauflogik im Mittelpunkt, sondern vielmehr das Herausarbeiten und Interpretieren farblicher oder morphologischer Charakteristika. Dies geschieht meistens unter Zuhilfenahme von Vorschaubildern, welche in den Programmierumgebungen parallel zur Ablauflogik dargestellt werden. Die Parameterfindung zur Extraktion relevanter Merkmale erfolgt dabei experimentell in der Software.

Der Abstraktionsgrad der Softwareumgebungen ist höher als in den anderen Steuerungsdomänen. So existieren häufig Assistenten, welche den Anwender durch die notwendigen Schritte leiten und Komfortfunktionen bieten, bspw. bei der Objektauswahl. Auch für die Suche nach angeschlossenen Kamerasystemen existieren Hilfsprogramme, sodass der Anwender weniger Aufwand zur Peripheriekonfiguration leisten muss (z. B. MVTEC 2018, S. 24 ff.).

Generell ist die proprietäre Verzahnung zwischen (Kamera-)Hardware und Software geringer als in den anderen beiden Domänen. So liefern bekannte Kamerahersteller zwar eigene Programme zur Bildverarbeitung aus, alternativ ist eine Ansteuerung mit Drittanbieter-Tools meist auch möglich. (DEMANT ET AL. 2011)

2.3 Cyber-physische Systeme

Der Begriff der Cyber-physischen Systeme (CPS) tritt in vielen Domänen auf, wie etwa in der Produktion, Logistik oder Medizintechnik. Er ist als nächste Stufe der Technologieintegration zu sehen, und wird deshalb, je nach Anwendungsbereich, unterschiedlich interpretiert. (WGP 2016, S. 10 ff.)

Der Einsatz von Cyber-physischen Produktionssystemen (CPPS) führt nach VOGELHEUSER ET AL. (2012, S. 6) zu einer *Smart Factory*, deren Eigenschaften in Bezug auf Qualität, Zeit und Kosten denen klassischer Produktionssysteme überlegen ist (ACATECH 2011, S. 35). WGP (2016) konkretisiert die Eigenschaften von CPS im Produktionsumfeld u. a. auf folgende Ausprägungen:

- Kommunikation sowohl zwischen CPPS als auch mit „smarten Produkten“
- Inhärente Selbstbeschreibung⁶
- Funktionen des CPS sind nicht zwingend an dessen Physis gebunden, sondern können über Software-Plattformen verteilt und genutzt werden.
- Selbstkonfiguration und Treffen situationspezifischer Entscheidungen
- Erhebung und Nutzung der Daten unabhängig von der Automatisierungspyramide (vgl. Abschnitt 2.2.1)
- Erkennung fehlender Dienste und Daten, Suche und Einbindung dieser
- Garantie von Qualitätsanforderungen (Quality-of-Service, QoS)
- Zugangskontrolle zu systemeigenen Daten und Diensten

CPS bzw. CPPS stellen ein Zukunftsbild dar, weshalb hier noch ein umfangreicher Forschungsbedarf besteht. Als vorrangig wird die *Standardisierung von Schnittstellen* genannt, um eine bessere Interoperabilität der eingesetzten Komponenten zu erreichen (und damit Veränderungsfähigkeit im produktionstechnischen Umfeld) (BITKOM 2015, S. 7; PLATTFORM I4.0 2015, S. 44 f.).

Eng damit verknüpft ist eine semantische Selbstbeschreibung der Systeme, auf deren gemeinsamer Basis Informationen zwischen den Teilnehmern ausgetauscht werden können (PLATTFORM I4.0 2015, S. 44 f.; ACATECH 2011, S. 25). PLATTFORM I4.0 (2016a, S. 17 f.) präzisiert dies weiter, indem eine Beschreibung der verfügbaren Fähigkeiten produktionstechnischer CPS als Forschungsbedarf genannt wird. Dabei heben die Autoren die Wichtigkeit von Plug&Play-Konzepten für die Vernetzung von *Sensoren* hervor, inklusive Ansätze für bereits existierende Systeme.

Nachdem CPS als Enabling Technology für RMS gesehen werden (vgl. Abschnitt 1.2), fließen die beschriebenen Forschungsbedarfe in die nachfolgende Untersuchung im Stand der Erkenntnisse (Kapitel 3) mit ein – im Rahmen einer Analyse digitaler Modelle zur Selbstbeschreibung im Automatisierungsumfeld.

⁶In diesem Zusammenhang wird auch häufig von „Digitaler Schatten“ gesprochen, welcher ein virtuelles Datenabbild des realen Systems darstellt. Die Bezeichnung „Digitaler Zwilling“ (Digital Twin) wird für ein umfassendes Echtzeitabbild verwendet, inklusive einem hinreichend genauen Simulationsmodell und einer Verhaltensbeschreibung. (WGP 2016, S. 23)

3 Stand der Erkenntnisse

In den vergangenen beiden Kapiteln wurden verschiedenartige Herausforderungen bei der (Re-)Konfiguration von Montagesystemen beschrieben. Als wiederkehrende Merkmale treten hierbei auf:

- Hohe Komplexität (Vielfalt, Vielzahl, Veränderlichkeit) der Prozesse, Aktivitäten sowie der eingesetzten Betriebsmittel in der Montagetechnik
- Fehlende Durchgängigkeit der Datennutzung über die einzelnen Phasen im Anlagenlebenszyklus hinweg
- Hohe manuelle Aufwände bei der Vernetzung und Programmierung von Automatisierungskomponenten

Mit Blick auf den aufgestellten Betrachtungsraum (vgl. Abschnitt 1) lassen sich hieraus vier Handlungsfelder ableiten, die für eine *zukünftige Verbesserung der Rekonfigurationsfähigkeit* von Montageanlagen erfüllt sein müssen:

- H1 Der bisherige Aufwand zur kommunikationsseitigen Vernetzung von Automatisierungskomponenten muss durch geeignete Softwaremechanismen stark reduziert werden (→Abschnitt 3.1).
- H2 Digitale Beschreibungssystematiken von Prozessen und Betriebsmitteln müssen stärker vereinheitlicht werden, um eine bessere Einsetzbarkeit und Wiederverwendbarkeit zu erreichen (→Abschnitt 3.2).
- H3 Die Programmierung von Montageanlagen muss effizienter geschehen. Dies erfordert eine stärkere Abstraktion vom jeweils eingesetzten Softwaresystem als auch von der anzusteuern Hardware (→Abschnitt 3.3).

Während H1 – H3 vorrangig technische Aspekte adressieren, ist es außerdem notwendig, die Technik betreffende Entscheidungsprozesse zu unterstützen. Diese Aktivitäten sind hauptsächlich in der Entwurfsphase von Montagesystemen (vgl. Abschnitt 2.1.2) auszuüben. Die vorliegende Arbeit fokussiert sich in diesem Bereich auf die Tätigkeit

der *Auswahl von Betriebs- und Prüfmitteln*. Andere Aktivitäten wie bspw. die *Betriebsmittelkonstruktion* sind außerhalb des Betrachtungsraums oder werden wie im Falle der *Alternativenbewertung* nur am Rande behandelt. Das vierte Handlungsfeld lautet somit:

H4 Durchzuführende Entscheidungsprozesse im Rahmen der Ressourcenauswahl müssen durch geeignete Softwaremechanismen stärker automatisiert werden (→Abschnitt 3.4).

Auf Basis der Punkte H1 – H4 wird der nachfolgende Stand der Erkenntnisse gegliedert, in dem Konzepte aus Industrie und Forschung zur Lösung der jeweiligen Anforderungen vorgestellt werden. Eine Verbesserung zum Status quo wird definiert als Reduktion der notwendigen Ressourcen zur Durchführung der jeweiligen Aktivitäten. Als Metriken hierfür dienen das erforderliche Experten-Know-how und die notwendige Zeit zur Durchführung.

3.1 Ansätze zur automatisierten Vernetzung von Betriebsmitteln

3.1.1 Methoden zur automatisierten Konfiguration des Bus-Masters

GRUNWALD ET AL. (2008) entwickelten im Rahmen des Forschungsprojekts „Plug And Play Antriebs- und Steuerungskonzepte für die Produktion von morgen“ (PAPAS) Lösungen zur einfacheren Systemintegration von Industrierobotern. Hierzu wurde eine 3-stufige Schichtenarchitektur realisiert: Die *Kommunikationsschicht* stellt einen protokollunabhängigen Buszugriff sicher, die *Geräteschicht* kümmert sich um die automatische Integration gefundener Geräte in die IRS, während die *Funktionsschicht* eine generische Schnittstelle zur Applikationsebene bereitstellt. Die Kommunikationsschicht adressiert hierbei die Schichten 1 - 5 des ISO/OSI-Modells (vgl. Abschnitt 2.2.1). Zur Kommunikation setzt der Master sog. „Device Requests“ ab, die durch das jeweilige Gerät beantwortet werden, z. B. Get_Status. Hierfür wird eine Grundmenge an Device Requests spezifiziert. Das PAPAS-Kommunikationsmodell erlaubt eine automatische und echtzeitfähige Integration von Komponenten, Voraussetzung ist jedoch eine Unterstützung des Konzepts durch alle Teilnehmer. Die Umsetzung erfolgte anhand eines Prototyps für das RTE-Protokoll ETHERCAT.

ANDREN ET AL. (2012) beschreiben den Ansatz eines Buskopplers, welcher die Übersetzung zwischen verschiedenen Busstandards vornimmt. In dem Konzept wird die Kommunikation zwischen zwei Geräten einmalig auf abstrakter Ebene modelliert und im Anschluss über einen Transformator für den verwendeten Bus generiert. Für die Definition der Kommunikationsstruktur wird der Standard DIN EN 61499 verwendet. Das Konzept richtet sich an Szenarien, bei denen ein Buskoppler zwischen zwei Protokollen übersetzt und innerhalb einer Rekonfiguration das Protokoll gewechselt werden soll. Es setzt die Verwendung von „DIN EN 61499“-kompatiblen Ressourcen voraus, welche bis dato nur eine geringe Marktdurchdringung haben.

KRUG (2013) und REINHART & KRUG (2012) betrachten einen Ansatz zur automatischen Vernetzung von Industrierobotersteuerungen mit Peripheriegeräten („Plug&Produce“). In dem Konzept besitzen sämtliche Ressourcen eine inhärente Selbstbeschreibung, welche über das TCP/IP-Protokoll heruntergeladen und in ein gemeinsames „Zustandsmodell“ integriert werden. Dieses beschreibt den aktuellen gerätetechnischen Aufbau der Roboterzelle auf Basis herstellerübergreifender Standards wie AUTOMATIONML oder COLLADA. Über einen treiberbasierten Ansatz kann auf die Konfiguration von unterschiedlichen Robotersteuerungen einheitlich zugegriffen werden. Der Bus-Master der Steuerung wird automatisch konfiguriert, indem aus dem Zustandsmodell Gerätebeschreibungsdateien zur Buskonfiguration kopiert (vgl. Abschnitt 2.2.1) und über den Steuerungstreiber integriert werden. Darüber hinaus werden gekapselte Programmbausteine zur erleichterten Ansteuerung der Komponenten in die IRS importiert. Das Konzept ist voll echtzeitfähig, setzt jedoch die alleinige Existenz von RTE-Protokollen und sich selbst beschreibenden Komponenten voraus.

Die Reduktion der Inbetriebnahmeaufwände bei RTE-Netzwerken ist das Ziel der Arbeiten von DÜRKOP ET AL. (2014) und DÜRKOP ET AL. (2012). Analog zu REINHART & KRUG (2012) werden Geräte über den Ethernet-Teil eines RTE-Protokolls gesucht und deren Parameter zur Konfiguration einer Echtzeitkommunikation heruntergeladen. Hierbei wird OPC UA (Abschnitt 2.2.1) als Erkennungs- und Zugriffsmechanismus auf die einzelnen Feldgeräte genutzt. Geräte besitzen jeweils eigene OPC UA-Server, welche zusätzlich zu den Kommunikationsparametern semantisch beschriebene Prozessdaten enthalten. Die Prozessdaten werden mittels eines primitiven Algorithmus mit einem hardwareunabhängig definierten IEC 61131-3 Programm abgeglichen und bei entsprechender Eignung automatisch zugewiesen. Die Ausführung des IEC 61131-3 Programms erfolgt auf einem Windows PC, zur Umsetzung der OPC UA-Funktionalität

werden vorgeschaltete Mikrocontroller verwendet oder die Firmware der Feldgeräte modifiziert.

HODEK (2013) beschreibt eine Möglichkeit zur automatischen Integration von Feldgeräten in SPSen. Klassen von Feldgeräten stellen hierzu einheitliche Schnittstellen bereit (APIs), auf Basis derer hardwareunabhängige Steuerungsprogramme entwickelt werden können. Für die Eignung zur automatischen Konfiguration des Kommunikationssystems wurden aktuelle RTE-Protokolle und Feldbusse anhand eines umfangreichen Bewertungsschemas analysiert. HODEK (2013) gelangt zu dem Schluss, dass keines der industriellen Protokolle den Anforderungen für ein selbstkonfigurierendes Konzept gerecht wird und implementiert deshalb seinen Ansatz auf Basis von Standard-Ethernet. Die Umsetzung erlaubt den Austausch von zwei Feldgeräten im laufenden Betrieb der SPS. Jedes Feldgerät besitzt dabei einen vorgeschalteten Mikrocontroller, welcher die notwendigen Vernetzungsfunktionalitäten anbietet.

REGULIN ET AL. (2015) untersuchen zwei Ansätze zur automatisierten Konfiguration des RTE-Protokolls ETHERCAT. Beiden Konzepten ist gemein, dass die Erkundungsmechanismen des Protokolls genutzt werden, um geänderte Teilnehmer im Netzwerk zu finden. In der ersten Konfigurationsvariante („Single Master“) verwenden Geräte eine eigens definierte Gerätebeschreibung, welche durch einen modifizierten Bus-Master automatisiert angefragt und integriert wird. Dies läuft ohne Zutun des Anwenders ab, benötigt jedoch modifizierte Hardware. In der zweiten Variante („Multi Master“) werden alle möglichen Gerätekonfigurationen bereits vorab auf dem Bus-Master hinterlegt und bei Bedarf aktiviert. Hierdurch können konventionelle Feldgeräte verwendet werden, nachteilig ist jedoch der hohe Modellierungsaufwand und der dafür notwendige Speicherplatz auf der Steuerung. Evaluiert wurde das Konzept mit dem „Multi Master“-Ansatz auf Basis einer Open-Source-Lösung.

3.1.2 Dienstorientierte Architekturen und Middlewares

Bei einer dienstbasierten Architektur (auch „serviceorientierte Architektur“, SOA) handelt es sich um ein Entwurfsmuster für IT-Architekturen. Zielstellung ist eine flexible Strukturierung von verteilten Softwaresystemen. Ursprünglich aus dem Bereich der Geschäftsprozesse stammend, hält dieser Ansatz zunehmend Einzug in die Automatisierungstechnik. (EVERTZ & EPPLE 2014; MATHAS 2008)

Grundlegende Idee einer SOA liegt im Angebot von Funktionalitäten, sog. *Dienste* (engl. *Services*), durch Softwaresysteme. Diese können lokal oder im Netzwerk publiziert und durch andere Teilnehmer aufgerufen bzw. genutzt werden. Die technische Umsetzung eines Diensts bleibt für andere Teilnehmer verborgen. Dies führt zu einer hohen Wiederverwendbarkeit existierender Dienste, außerdem entsteht eine hohe Flexibilität auf Prozessebene, während die Komplexität zur Implementierung eines Diensts gering bleibt. (TAKAI 2017; MATHAS 2008)

Durch eine Kombination von Diensten lassen sich Sequenzen erstellen („Orchestrierung“), mit denen sich ausführbare Prozesse abbilden lassen. Die Modularität der Dienste schafft eine nur sehr lose Kopplung zwischen den beteiligten Softwarepartnern, wodurch eine Modifikation oder Rekombination existierender Abläufe aufwandsarm realisiert werden kann. Viele Internet-Anwendungen basieren technisch auf „Web Diensten“. Hierfür werden bekannte Standards wie „SOAP“-Schnittstellen verwendet, welche auf TCP mit HTTP aufbauen (LIEBHART 2007, S. 14 f.). Auch OPC UA (vgl. Abschnitt 2.2.1) setzt in einer möglichen Technologieabbildung auf SOAP auf, um dessen Informationsmodelle in Ethernet-Netzwerken anbieten zu können (ENSTE & MAHNKE 2011, S. 398).

Die genannten Vorteile von SOA in IT-Architekturen lassen sich auch auf die OT-Ebene übertragen. Der grundlegende Ansatz besteht hierbei in einem aus Dienstaufrufen basierenden Ablaufprogramm, für das im Netzwerk geeignete Feldgerätedienste zur Ausführung gesucht werden¹.

BOHN ET AL. (2006) und JAMMES & SMIT (2005) nutzen für dieses Konzept das „Device Profile for Web Services“ (DPWS). DPWS setzt auf SOAP auf und ermöglicht eine automatisierte Entdeckung von Diensten eingebetteter Systeme. In dem Konzept wird pro Geräteklasse ein Satz an Diensten definiert, z. B. Starte Motor (Dauer) für Antriebe. Diese können von einer Ablauflogik angesteuert werden. Auch die Kombination von Geräten auf Dienstebene ist möglich, um so komplexere Produktionseinheiten zu erstellen. Eine genauere Erläuterung wird diesbezüglich nicht gegeben. Die Evaluation erfolgt mit mehreren primitiven Geräten, die sich über DPWS automatisiert erkennen.

¹Die Zuordnung von geeigneten Ressourcen für eine definierte Aufgabe wird auch als „Matchmaking“ bezeichnet.

NAUMANN & VERL (2008) definieren ein SOA-Konzept auf Basis des XIRP-Protokolls (XML Interface for Robots and Peripherals, VDMA 66430-1). Darin lassen sich angeschlossene Geräte automatisiert entdecken und eine Gerätebeschreibung herunterladen. Diese wird von einer Plug&Produce-Software für die aufgabenorientierte Programmierung einer Roboterzelle genutzt (Abschnitt 3.3). Applikationsseitig werden mittels der in XIRP standardisierten Befehle Informationen zwischen Server (Gerät) und Client (Plug&Produce-Software) ausgetauscht. Hierfür können Geräteprofile angelegt werden, welche die Informationen einer Geräteklasse standardisieren. Die Daten liegen im XML-Format vor und werden über Ethernet/TCP ausgetauscht (TAVARES & PAULA CAURIN 2010). Hierdurch ist eine deterministische Kommunikation zwischen den Teilnehmern nicht möglich.

LOS KYLL ET AL. (2011) und LOS KYLL ET AL. (2012) nutzen DPWS, um semantisch standardisierte Gerätedienste anbieten und abgleichen zu können (vgl. Abschnitt 3.4). Als technische Realisierung werden dezidierte Mikrocontroller verwendet und als Gateway zwischen die Feldgeräte geschaltet. Einen ähnlichen Hardware-Ansatz gehen PFROMMER ET AL. (2015) und PFROMMER ET AL. (2014). Hier wird vor jedes Betriebsmittel eine sog. „Skill Execution Engine“ (SEE) geschaltet, die für ein überlagertes Planungssystem eine standardisierte Eigenschaften- und Fähigkeitenbeschreibung anbietet. Als Übertragungsprotokoll wird OPC UA verwendet, weitere technische Details werden nicht angegeben.

Unter dem Gesichtspunkt neuer INDUSTRIE 4.0-Architekturen vergleichen EVERTZ & EPPLE (2014) verschiedene SOA-Ansätze, u. a. Webdienste, OPC UA und proprietäre Lösungen von Microsoft, IBM und SAP. Die Autoren gelangen zu dem Schluss, dass sich viele der Ansätze in ihrem Funktionsumfang ähneln. OPC UA hat hierbei den Vorteil, dass prinzipiell ein Einsatz auf sehr ressourcenbeschränkten Geräten vorgesehen ist. Problematisch ist jedoch der lösungs- und nutzerspezifische Aufbau der einzelnen Informationsmodelle. Um der Forderung nach einer größeren Interoperabilität Rechnung zu tragen, wird eine vereinheitlichte Beschreibung der Semantik und der Eigenschaften der Dienste gefordert.

Trotz der genannten Vorteile sind dienstorientierte Architekturen nach wie vor eine Ausnahme in der Automatisierungstechnik (DÜRKOP ET AL. 2014). DÜRKOP ET AL. (2014) sehen Gründe hierfür in der fehlenden Rechenkapazität einfacher Feldgeräte, harten Echtzeitanforderungen in der Kommunikation sowie ungeklärten Fragen in den Bereichen Engineeringtools, Sicherheit und Standardisierung.

3.2 Digitale Modelle zur Erhöhung der Interoperabilität

Nach DIN 62390 und VDI (2016) sind *Ressourcen* interoperabel, wenn sie zusammenarbeiten können, um ihre Rollen in einem logischen Ablauf zu erfüllen. Hierzu müssen alle Systeme sowohl auf einer Kommunikationsprotokollebene kompatibel sein als auch die übertragenen Parameter und Fähigkeiten gleich interpretieren (DIN 62390, S. 19 f.). *Ressourcenseitige Interoperabilität* schafft eine gewisse Trennung zwischen einer geforderten Gerätefunktionalität und der damit unterlagerten hardwaretechnischen Realisierung. Eine *softwareseitige Interoperabilität* beschreibt eine möglichst nahtlose Zusammenarbeit unterschiedlicher Applikationen (IWG 2015). Interoperabilität wird als wichtiger Baustein für eine Umsetzung von INDUSTRIE 4.0 gesehen (vgl. VDI 2016).

Für eine Klassifizierung verwendeter Modelle im Kontext der digitalen Fabrik hat sich eine Aufteilung in die Domänen *Produkt*, *Prozess* und *Ressource* (PPR) bewährt (DRATH 2010, S. 83 ff.). Die drei Domänen stehen hierbei in Beziehung zueinander: Produkte werden durch Prozesse hergestellt, Prozesse ihrerseits durch Ressourcen umgesetzt. Produkte laufen (physisch) über Ressourcen. (DRATH 2010, S. 83 ff.)

Interoperabilität innerhalb der drei Domänen wird meist über Standardisierung erreicht. Darüber hinaus existieren zahlreiche Konzepte um durch eine automatisierte Verknüpfung der Bereiche Aufwände bei der Planung oder Programmierung von Betriebsmitteln zu reduzieren. Hierbei wird häufig auf das Konzept der *Fähigkeiten* (engl. *Skills*) zurückgegriffen. Fähigkeiten stellen nach BACKHAUS & REINHART (2015) eine semantisch definierte und herstellerneutrale Funktionalität dar, welche von Ressourcen umgesetzt werden kann. Nachdem auch Produktionsprozesse einen funktionalen Charakter aufweisen, bilden Fähigkeiten im Sinne des PPR-Modells eine Verknüpfung zwischen der Prozess- und Ressourcenseite, vgl. Abbildung 3.1. In Anlehnung wird dies auch als *PPRS-Modell*² bezeichnet (PFROMMER ET AL. 2013).

Die nachfolgenden Abschnitte widmen sich der *strukturellen Beschreibung* der einzelnen Domänen innerhalb des Betrachtungsraums, der *Nutzen aus ihrer Verknüpfung* wird in den Abschnitten 3.3 und 3.4 gegeben.

²Product-Process-Resource-Skill

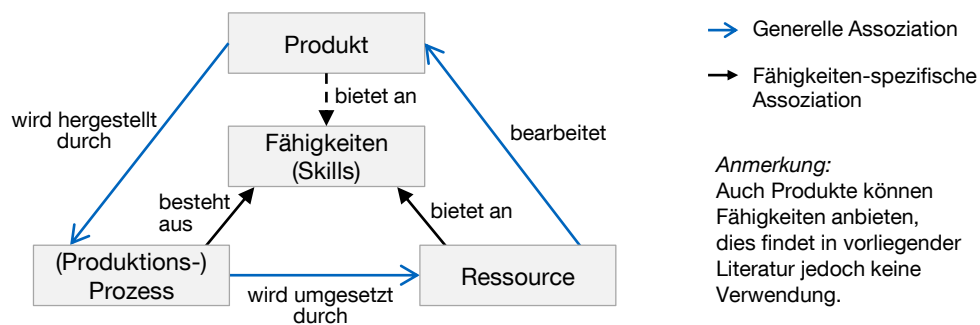


Abbildung 3.1: Beziehungen im PPRS-Modell

3.2.1 Modellierung von Produkten

Produkte besitzen *Merkmale*, über die sie beschrieben und unterschieden werden können (DIN 4002-1). Die Normenreihe der DIN 4002 gibt Anwendern eine Hilfsstellung bei der Strukturierung und Beschreibung von Merkmalen. Danach können Merkmale unterschiedlicher Natur sein, bspw. administrativ, kaufmännisch oder technisch.

Ein wesentliches Merkmal von Produkten ist nach VAJNA ET AL. (2009, S. 46) deren *Geometrie*. Obgleich sie alleine wenig Aussagekraft über die Eigenschaften eines Produkts besitzt, hängen nahezu alle Produkteigenschaften von der Geometrie ab (VAJNA ET AL. 2009, S. 46). Für die Gestaltung (engl. *Design*) von Produkt- und Bauteilgeometrien werden „*Computer-aided Design*“-Programme (CAD-Programme) verwendet. Diese nutzen verschiedene informationstechnische Ansätze zur Repräsentation des Objekts, wie etwa Flächenmodelle, Volumenmodelle oder Voxel-basierte Darstellungen (vgl. Abbildung 3.2). Eine Erklärung der Ansätze mit ihren jeweiligen Vor- und Nachteilen kann VAJNA ET AL. (2009, S. 175 ff.) entnommen werden.

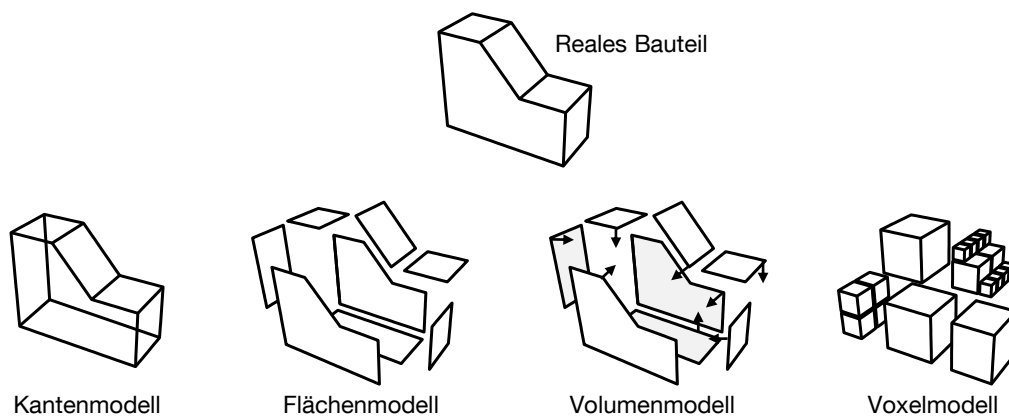


Abbildung 3.2: Repräsentationsarten von 3D-Modellen (erweiterte Darstellung in Anlehnung an VAJNA ET AL. 2009, S. 176)

Bekannte herstellerübergreifende Austauschformate bei Geometriebeschreibungen sind u. a. 3D-PDF, STEP, JT und COLLADA (HUBER 2016, S. 110; DRATH 2010, S. 95). STEP und COLLADA besitzen zusätzlich die Möglichkeit, weiterführende Produktmerkmale (z. B. kinematische Zusammenhänge) zu hinterlegen (KHRONOS GROUP 2008). Die Eigenschaften der Formate variieren in puncto Dateigröße, Genauigkeit der Modellierung, Herstellerunterstützung etc. Einen Technologievergleich bietet PROSTEP (2013).

Im Bereich der nicht-geometrischen Produktmerkmale ist die Standardisierung weniger vorangeschritten. Unternehmen implementieren meist eigene Merkmalsbeschreibungen (vgl. SENDLER ET AL. 2013, S. 115; EIGNER & STELZER 2009, S. 74). Ausnahme stellt der Klassifizierungsstandard *eCl@ss* dar, welcher eine Taxonomie von eindeutigen Produktklassen beschreibt (*eCl@ss* 2015, S. 10). Die mehr als 40.000 Einträge erstrecken sich über unterschiedlichste Bereiche wie Lebensmittel, Logistik, Medizin- und Automatisierungstechnik (*eCl@ss* 2015, S. 23; HEPP ET AL. 2007, S. 92). Zusätzlich definiert *eCl@ss* mehrere tausend produktbeschreibende Merkmale, wie etwa „Herstellername“ und „Artikelnummer“. *eCl@ss* ist vor allem im Bereich Einkauf und Beschaffung etabliert (HUBER 2016, S. 102), weshalb kaufmännische Merkmale die am häufigsten eingesetzten darstellen. Diesbezüglich merken HEPP ET AL. (2007, S. 98 ff.) an, dass nur wenige Klassen in *eCl@ss* spezialisierte Merkmale aufweisen. Im Zuge der INDUSTRIE 4.0-Initiative versucht *eCl@ss* verstärkt auch technische Merkmale aufzunehmen, sodass für einzelne Automatisierungskomponenten spezifische Merkmalslisten bereitstehen. Die Merkmalsbeschreibung bleibt hierbei auf eine Eigenschaftsebene beschränkt, Funktionalitäten und Dienste von Geräten werden nicht abgebildet.

Die Sammlung und Versionierung der anfallenden Produktdaten wird von *Produktdatenmanagement*-Systemen (*PDM*-Systemen) übernommen. Unter Produktdaten fallen neben den beschriebenen Geometriedarstellungen und Merkmalen auch die Produktstruktur, Stücklisten, Materialien, Berechnungsergebnisse u. v. m. Eine Erweiterung stellen *PLM*-Systeme (*Product Lifecycle Management*) dar, die umfassend alle anfallenden Produktinformationen integrieren. Sie aggregieren Daten zur Produktion und Fertigung (z. B. Werkzeuge), zum Kundenmanagement sowie zu den Zuliefererketten. Zur Anbindung der verschiedenen Erzeugeranwendungen (wie etwa CAD-Programme) besitzen *PLM*-Systeme umfangreiche, spezifische Schnittstellen. (EIGNER & STELZER 2009, S. 35 ff.)

Abbildung 3.3 zeigt die beschriebenen Zusammenhänge in der Übersicht.

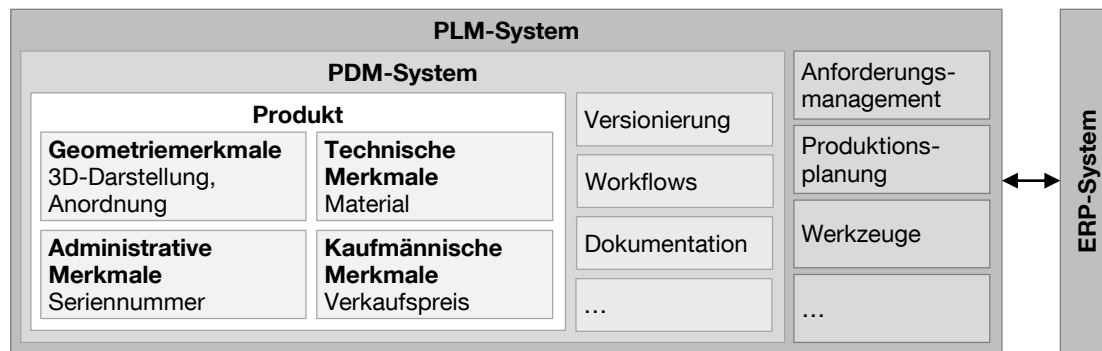


Abbildung 3.3: Produktmerkmale unterschiedlicher Art werden in PDM- oder PLM-Systemen gespeichert und verwaltet.

3.2.2 Modellierung von Prozessen

Industrielle Standards

In der Planungsphase von Montagesystemen werden funktionale ablauforientierte Modellierungen eingesetzt, um eine erste Beschreibung des zu gestaltenden Systems zu schaffen. Die Beschreibungsmittel dienen meist dem verbalen Austausch zwischen beteiligten Fachdisziplinen, weshalb bevorzugt auf eine grafische Darstellung in Form von Flussdiagrammen zurückgegriffen wird. Über meist unternehmensspezifisch definierte ikonische und textuelle Elemente erfolgt die Festlegung des Sprachumfangs. Beispiele hierfür sind in SCHENK ET AL. (2014) und TOMPKINS ET AL. (2006) für den Bereich der Fabrikplanung zu finden sowie in BULLINGER (1986) für den Bereich der Montageplanung.

Zur grafischen Darstellung von Fügereihenfolgen gilt der *Montagevorranggraph (MVG)* als bekanntester Vertreter (Abbildung 3.4a). Gebildet wird dieser auf Basis der erstellten Produktmodelle (Abschnitt 3.2.1). Wird der MVG weiter im Hinblick auf das technische Montagesystem detailliert, so entsteht ein *Montageablaufplan*, welcher die Reihenfolge durchzuführender Montage-, Transport- und Prüfprozesse sowie deren Ausführungszeiten enthält (Abbildung 3.4b) (CIRP 2011 S. 356 ff.; TOMPKINS ET AL. 2006, S. 41). Zur automatisierten Erzeugung von Montagevorranggraphen existieren zahlreiche Forschungsarbeiten, HU ET AL. (2011) und THOMAS ET AL. (2015) liefern hierzu gute Übersichten. Als häufiger Ansatz wird die virtuelle Demontage eines als 3D-CAD-Modell vorliegenden Produkts genutzt (bspw. MICHNIEWICZ ET AL. 2016 oder XU ET AL. 2012). Hierbei wird mit unterschiedlichen Strategien versucht, Trajektorien für

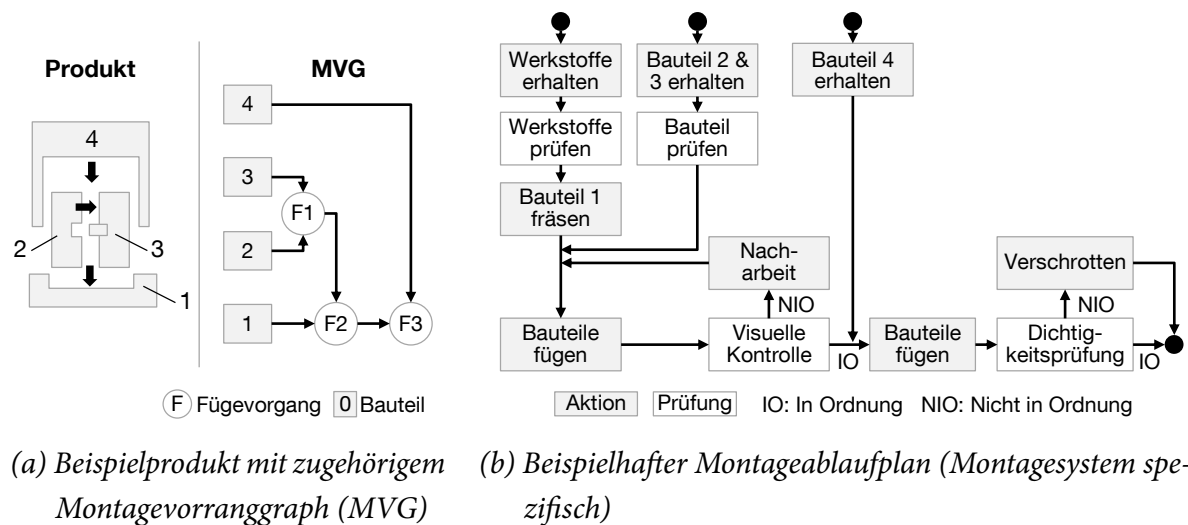
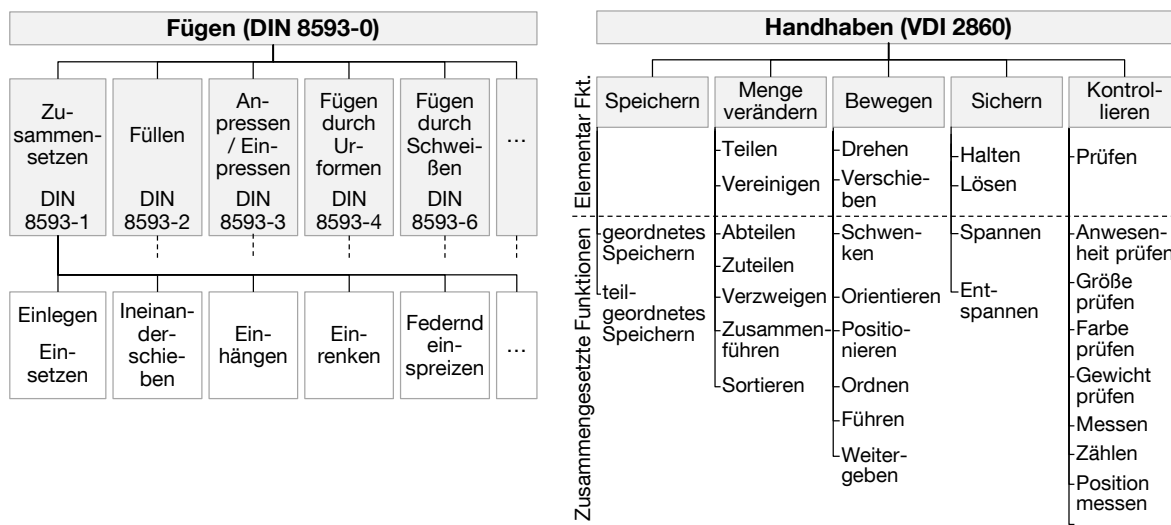


Abbildung 3.4: Grafische Darstellung von Montageabläufen

das kollisionsfreie Herauslösen von Bauteilen aus dem Produkt zu generieren. Gelingt dies, wird das Bauteil aus dem Produkt entfernt und der Prozess erneut beim nächsten Bauteil wiederholt. Abschließend wird die erzeugte Demontagesequenz umgekehrt, wodurch ein kollisionsfreier Montageablauf samt zugehöriger Bauteiltrajektorien entsteht („Assembly by Disassembly“).

Die Beschreibung von Montageablaufplänen ist durch Anwendung der DIN 8593 und VDI 2860 möglich. Während die DIN 8593 eine umfangreiche Taxonomie für Fügeoperationen liefert, listet die VDI 2860 Prozesse zur Sicherung, Transport und Kontrolle (Abbildung 3.5). Beide Standards treffen keine Aussagen zu prozessdefinierenden Parametern, wie etwa Kräften oder Geschwindigkeiten. Darüber hinaus weist die Taxonomie der VDI 2860 sich überschneidende Prozesse auf (SCHMIDT 1992, S. 29 f.) und ist vor allem im Bereich des „Kontrollierens“ unzureichend strukturiert sowie unvollständig definiert (HAMMERSTINGL ET AL. 2015, S. 51).

Die beschriebenen grafischen Prozessmodelle erreichen durch ihren funktionalen Charakter eine gewisse Interoperabilität in Bezug auf die später umzusetzende technische Lösung. Da sie vorrangig für die Kommunikation zwischen menschlichen Akteuren gedacht sind, existiert jedoch kein anerkanntes Datenaustauschformat. Als Lösung lässt sich die in der IEC 61131-3 definierte *Ablaufsprache (AS)* (engl. *Sequential Function Charts, SFC*) verwenden, deren Zielstellung die Beschreibung von Steuerungssequenzen ist (vgl. Abschnitt 2.2.2). Die Sprache basiert auf Zuständen; Aktionen, die während aktiver Zustände ausgeführt werden; sowie Zustandsübergangsbedingungen zum Schalten zwischen Zuständen. Als Austauschformat wurde ein XML-Modell unter dem Namen



(a) DIN 8593 (Auszug)

(b) VDI 2860 (Auszug)

Abbildung 3.5: Klassifizierung von Montageprozessen

„PLCopen XML“ entwickelt (PLCOPEN 2009). Der „AutomationML“-Standard (vgl. Abschnitt 3.2.3) nutzt PLCopen XML als Beschreibungsmittel für den übergreifenden Austausch von Engineeringprozessabläufen, bspw. für Gantt-Charts oder Zustandsautomaten (DRATH 2010, S. 153 ff.). Hierzu wird eine Abstraktionsschicht eingeführt, der *Intermediate Modelling Layer (IML)*. Grafisch modellierte Prozessabläufe werden zuerst in den IML übersetzt, von diesem aus ein Transformator „PLCopen XML“-Code erzeugt.

Forschungsarbeiten

Zusätzlich zu den beschriebenen industriellen Standards existieren mehrere Forschungsarbeiten, die eigene Prozessdefinitionen entwerfen. Der Fokus liegt meist auf Fähigkeiten, mit denen eine automatisierte Verknüpfung zwischen der Ressourcen- und der Prozessebene im PPRS-Sinne erreicht werden soll.

Die Zielstellung von SCHMIDT (1992, S. 28 ff.) ist die Konzeption flexibler Montagesysteme. Er kritisiert die Überschneidung einzelner Prozesse in der VDI 2860 und leitet für die weitere Verwendung fünf überschneidungsfreie Grundprozesse ab: Speichern, Bewegen, Verbinden, Verändern und Vergleichen. Der Autor gibt für jeden Grundprozess eine einfache Liste von beeinflussenden Parametern an, wie etwa Vergleichen = $f(\text{Toleranz, Masse, Orientierung, Zeit, Hilfsstoff, Funktion, [...]})$. Eine Herleitung der Parameter wird nicht angeführt.

Zur verbesserten Rekonfigurationsfähigkeit beschreiben SMALE & RATCHEV (2009) sechs Fähigkeitenklassen in der Montage: Bewegen, Verbinden, Sichern, Messen, Zuführen und Bearbeiten. Eine weitere Unterteilung wird nicht vorgenommen, weshalb diese Klassifizierung dem Ansatz von SCHMIDT (1992) ähnelt.

In dem SIARAS-Forschungsprojekt wird eine Fähigkeitenklassifizierung erstellt, welche auf der DIN 8580 basiert. Die Norm listet Taxonomien für Fertigungsverfahren auf, wie etwa das Urformen oder Umformen. Für sensorische Fähigkeiten wird eine von Normen unabhängige Auflistung erstellt, deren Herleitung jedoch nicht angegeben wird. Die Taxonomie ist nicht überschneidungsfrei, so sind bspw. Menge teilen, Sichern und Bewegen im gleichen Ast einsortiert. (BENGEL 2008; STENMARK & MALEC 2015)

KLUGE (2011) kombiniert die VDI 2860 und DIN 8593 mit dem Ziel einer fähigkeitsbasierten Planung modularer Montagesysteme. Er ergänzt die entstandene Taxonomie um Justage- und Sonderoperationen, wie sie beispielsweise auch LOTTER & WIENDAHL (2012) (vgl. Abschnitt 2.1.1) aufführen. Im Zuge dessen erbt sie auch die genannten Nachteile. Der Autor sammelt aus Literaturquellen 48 Prozesscharakteristika, die er den Prozessschritten der Taxonomie als Parameter zuweist. Eine Quantifizierung wird nicht getroffen, weiterhin werden physikalische und planungsseitige Parameter vermischt, z. B. Zusammensetzen = $f(\text{Fügebewegung, Fügerichtung, Losgröße, Stückzahl, [...]})$.

Im Rahmen der aufgabenorientierten Programmierung von Montagesystemen (vgl. Abschnitt 3.3) entwickelt BACKHAUS (2016) eine Taxonomie von Gerätefähigkeiten, die von Experten zu Montageprozessen zusammengesetzt werden können. Als Basis werden die fünf von SCHMIDT (1992) definierten Grundprozesse genutzt, unter diese die Normen VDI 2860 und DIN 8593 eingegliedert werden – teilweise mit Anpassungen, um weniger Überschneidungen aufzuweisen. Der Autor liefert hierzu ein methodisches Vorgehen, das auf einer Identifikation von Merkmalen zur Unterscheidung von Prozessen basiert. Dies führt stellenweise zu einer feinteiligen Gliederung, bspw. wird ein Bewegungsprozess im kartesischen Raum in acht Unterprozesse auf vier Hierarchieebenen geteilt (Fördern diskret, Fördern kontinuierlich, Drehen diskret, Drehen kontinuierlich, Weitergeben, [...]), vgl. BACKHAUS (2016, S. 214 f.). Die Festlegung der Prozessparameter erfolgt durch den Anwender, eine Herleitung der spezifischen Prozessparameter wird nicht gegeben.

Auch JÄRVENPÄÄ (2012) lehnt sich an die DIN 8593 an und kombiniert diese mit produktionstechnischen Ontologien aus den EUPASS und CO2PE! Forschungsprojekten.

In der entstandenen Taxonomie sind ca. 90 Gerätefähigkeiten hinterlegt, neben einer Vielzahl der aus Normen bekannten Materialbearbeitungs- und Fügeoperationen sind auch logistische und Sonderprozesse enthalten, wie etwa Reinigen und Versiegeln. Eine Bestimmung der Fähigkeitenparameter erfolgt durch Sammlung aus Herstellerdatenblättern, genauere Angaben werden hierbei nicht gegeben. Die Taxonomie ist nicht frei von Überschneidungen und beschreibt kaum sensorische Fähigkeiten.

HUCKABY & CHRISTENSEN (2012) definieren eine Auflistung von Fähigkeiten für den Wissenstransfer in roboterbasierten Montagezellen. Die Liste enthält unterschiedlichste Prozesse (z. B. Erkennen, Transportieren, Bohren), deren Herleitung wird nicht weiter beschrieben. Zusätzlich wird eine Taxonomie von sechs *Einschränkungen* (engl. *Constraints*) definiert, die ein jeder Skill besitzen kann. Beispiele hierfür sind eine maximale Ausführzeit oder die Bahngeschwindigkeit des Roboters.

3.2.3 Modellierung von Betriebsmitteln

Industrielle Standards

Im Gegensatz zur Produktbeschreibung stellt die umfassende digitale Betriebsmittelbeschreibung ein jüngeres Feld dar und ist in der Industrie noch nicht umfangreich etabliert. Die unter Abschnitt 3.2.1 beschriebenen Standards zur Geometriebeschreibung gelten für diese Domäne gleichermaßen.

Im Bereich der *Kommunikation* existieren Standards zur Vereinheitlichung der Schnittstellen und des Kommunikationsverhaltens einer Geräteklasse („Profile“). Sie sind protokollabhängig. So existiert bspw. mit „PROFIdrive“ ein Profil für elektrische Antriebe mit PROFIBUS-Anschluss (HODEK 2013, S. 26 f.). Auf *Applikationsseite* existieren Profile zur Standardisierung der Ansteuerung von Feldgeräten. So liefert das „motion control“-Profil der PLCopen-Organisation Funktionsblöcke für die Programmierung von Antrieben in Industriesteuerungen (PLCOPEN 2011). Damit erstellte Anwendungen sind agnostisch von der verwendeten Softwareplattform und jeweiligen Hardware des Antriebsherstellers. Generell ist der Grad und Umfang der Standardisierung in den verschiedenen Profilarten sehr unterschiedlich und bleibt meist auf einer hardwarenahen Ebene. HODEK (2013, S. 27) merkt hierzu an, dass eine Anhebung des Abstraktionsniveaus auf eine funktionsorientierte, semantische Darstellung notwendig ist.

Eine Standardisierung im Engineering- und Inbetriebnahmeprozess von Produktionssystemen soll durch *AutomationML* (IEC 62714) erreicht werden. *AutomationML* ist ein XML-basiertes Austauschformat zwischen Engineeringtools. Geräte lassen sich objektorientiert modellieren und mit Merkmalen beschreiben. Hierfür wird der IEC 62424 Standard „Computer Aided Engineering Exchange (CAEX)“ als Dachformat verwendet. Auch etwaige Gerätehierarchien lassen sich über CAEX abbilden. Eine Geometrie- und Verhaltensbeschreibung wird nicht direkt angegeben, sondern auf die externen Standards COLLADA und PLCopen XML referenziert (Abschnitt 3.2.1 und 3.2.2). *AutomationML* ist als Meta-Modell spezifiziert, d. h., Anwender müssen auf Basis der in *AutomationML* definierten Regeln eine konkrete Beschreibung der Ressource selbst aufbauen. Zur Modellierung von Produktionsanlagen unterstützt *AutomationML* das PPR-Konzept. (VOGEL-HEUSER ET AL. 2017, S. 216 ff.; DRATH 2010)

Forschungsarbeiten

Forschungsarbeiten im Betrachtungsraum nutzen meist eine Ressourcenbeschreibung, um über die Verknüpfung der PPRS-Domänen eine hardwareseitige Interoperabilität zu erreichen. Nachfolgend wird eine Auflistung relevanter Arbeiten im Betrachtungsraum gegeben. Die zugehörige Merkmalsübersicht zeigt Tabelle 3.1.

Neben Fähigkeiten werden in dem SIARAS-Forschungsprojekt auch Ressourcen beschrieben. Ähnlich dem eCl@ss-Standard wird eine spezialisierende Hierarchie von Gerätetypen abgebildet, wie etwa Gerät \leftarrow Sensor \leftarrow Optischer Sensor \leftarrow Kamera. Zusätzlich existiert ein eigener Baum für Geräteeigenschaften, wie etwa Kommunikationsschnittstellen, Geometrie und Kosten. Eine nähere Beschreibung der Eigenschaften oder eine Herleitung wird nicht gegeben. (BENGEL 2008; STENMARK & MALEC 2015)

NAUMANN ET AL. (2010) entwickeln die in SIARAS erstellten Ontologien weiter, hin zu einem sog. „Knowledge Integration Framework“ (KIF). Dieses enthält neben Prozesswissen aus dem Montagebereich auch eine Reihe von Geräteinformationen. Hierzu zählen Fähigkeiten und Eigenschaften, Einschränkungen sowie geometrische und kinematische Modelle auf Basis von COLLADA. Einen weiteren Bereich stellt das Gefährdungspotential von Ressourcen dar, was beispielsweise für eine Mensch-Roboter-Kollaboration von Interesse ist. Die Erklärung bleibt auf einem abstrakten Niveau, detaillierte Beschreibungen der Strukturen werden nicht gegeben.

JÄRVENPÄÄ (2012) unterteilt ihre Ressourcenbeschreibung in eine Template-seitige und eine Individualbeschreibung. Die Templatebeschreibung enthält Merkmale einer Geräteklasse und ist großteils kaufmännischer Natur. Die Individualbeschreibung wird für jedes Gerät erstellt, in dem zur Laufzeit verschiedene Faktoren, wie etwa die Betriebsstunden oder Ausfallzeiten gemessen werden. Auch diese Parameter sind vorrangig kaufmännischer Art. Neben den Fähigkeiten (vgl. Abschnitt 3.2.2) existiert eine Schnittstellenbeschreibung, welche in die Bereiche mechanisch, informationstechnisch, elektrisch und fluidisch kategorisiert ist. Alle Parameter werden als Ontologien modelliert und über ein Inferenzsystem für ein Matchmaking genutzt. Die Herleitung erfolgt durch eine Sammlung von Geräteparametern aus Datenblättern der Hersteller.

Auf Basis von AutomationML definiert KRUG (2013, S. 96 f.) das Zustandsmodell einer Roboterzelle, welches Komponenten und deren Merkmale in einer herstellerneutralen Art beschreibt. Allgemeine Informationen (z. B. Name, ID) werden in CAEX modelliert, für Geometrie- und Kinematikbeschreibung wird COLLADA verwendet. Darüber hinaus existieren Merkmale für die Angabe von Dokumentationen, Kommunikationsspezifika und Programmcode zur Geräteansteuerung. Hierfür werden eigene Strukturen in CAEX aufgebaut. BACKHAUS (2016) nutzt als Basis die Ressourcenbeschreibung von KRUG (2013) und erweitert diese um eine Beschreibung der Gerätefähigkeiten. Da sein Ansatz simulationsfokussierter ist, werden in das Modell zusätzlich Informationen zur Raumpose, zu dem Arbeitsbereich und eine Verhaltensbeschreibung hinterlegt.

Tabelle 3.1: Übersicht modellierter Betriebsmittelmerkmale ausgewählter Forschungsarbeiten

Quelle	Allgemein							Funktion		Kommunikat.			Simulation				
	Allg. Informationen	Physik. Eigenschaften	Hardwarestruktur	Dokumentation	Kaufmänn. Merkmale	Schnittstellen	Safety	Funktionen	Fähigkeiten	Signalcodierung	Mapping	Programmcode	Raumpose	Kinematik	Geometrie	Arbeitsbereiche	Verhaltensbeschreibung
SIARAS (2015)	X	X	-	-	-	X	-	-	X	X	-	-	X	-	X	X	-
NAUMANN (2010)	X	-	-	-	-	X	X	-	X	-	-	-	-	CO	CO	-	X
JÄRVENPÄÄ (2012)	X	X	-	-	X	X	-	-	X	X	-	-	-	-	-	-	-
KRUG (2013)	CA	X	-	OF	-	-	-	IR	-	PR	CA	IR	-	CO	CO	-	-
BACKHAUS (2016)	CA	X	-	OF	-	-	-	CA	CA	PR	CA	IR/PL	X	CO	CO	X	SF

-: Nicht vorhanden X: Eigene Struktur CA: CAEX CO: COLLADA IR: IRL PR: Proprietär PL: PLCopen SF: SFC

3.3 Ansätze zur aufgabenorientierten Programmierung von Montageanlagen

Eine Effizienzsteigerung der Programmierung von Montageanlagen kann auf vielfältige Weise geschehen, bspw. durch einheitliche Programmierrichtlinien oder eine verbesserte Teamkollaboration. Einen vielversprechenden technischen Ansatz stellt die *aufgabenorientierte Programmierung (AoP)* dar. Hierbei beschreibt der Anwender nur mehr auf einer höheren Abstraktionsebene die durchzuführende Aufgabe des Systems, Softwarealgorithmen übersetzen diese auf Basis von Regeln und Modellen in Programmcode. (SIEGERT & BOCIONEK 1996, S. 197 f.)

Die AoP wurde intensiv im Bereich der Programmierung von Industrierobotern erforscht, um die langwierige Eingabe der Bewegungskordinaten zu automatisieren (z. B. MUNZERT 2010). Laut ULRICH (2018, S. 15 f.) fokussieren die bisher vorgestellten Methoden jedoch nur spezielle Bereiche, eine universelle Einsetzbarkeit ist noch nicht gegeben. WECK (2006, S. 409 ff.) beschreibt die Bestandteile eines aufgabenorientierten Industrieroboterprogrammiersystems in der Montage (Abbildung 3.6):

Das *Umweltmodell* enthält alle Informationen der Elemente in der zu planenden Roboterzelle. Hierzu zählen Produkt- und Ressourcendaten.

Das *Aufgabenmodell* speichert die durchzuführende Arbeitsanweisung. Über einen *Aktionsplaner* wird sie in Teilprozesse zerlegt, welche im Anschluss von verschiedenen *Planungsmodulen* weiter ausgearbeitet werden.

In den Planungsmodulen steckt das technologische Expertenwissen. Der *Fügeplaner* berechnet kollisionsfreie Fügeposen und achtet auf die Einhaltung der benötigten Fügekräfte. Das *Bahnplanungsmodul* berechnet kollisionsfreie Bewegungsabläufe zwischen den Start- und Zielposen der jeweiligen Fügeaufgabe. Die *Greifplanung* sorgt für einen sicheren und schädigungsfreien Griff der Bauteile. Der Algorithmus muss dazu die Greifart, -flächen und -position kennen, Steifigkeits- und Reibwerte des Bauteils wissen sowie die auftretenden Kräfte und Beschleunigungen bei der Transferbewegung ermitteln.

Dieser Aufbau findet sich in vielen Arbeiten zur aufgabenorientierten Programmierung wieder, etwa bei BACKHAUS (2016), HUMBURGER (1998) oder SECKNER (2008). Darüber hinaus existieren zahlreiche Konzepte, die sich mit der Ausgestaltung einzelner

Planungsmodulen beschäftigen, z. B. MAHLER & GOLDBERG (2017), LEI & WISSE (2016) (Greifplanung); ULRICH (2018), ZHANG ET AL. (2017) (Bahnplanung); oder MICHNIEWICZ ET AL. (2016), THOMAS ET AL. (2015) (Fügeplanung).

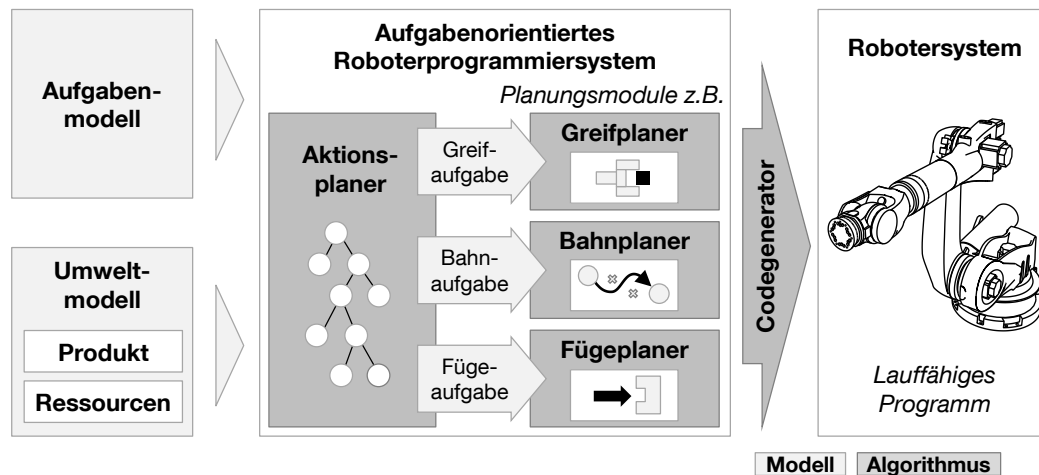


Abbildung 3.6: Struktur eines aufgabenorientierten Roboterprogrammiersystems, in Anlehnung an WECK (2006, S. 410)

Ein System zur aufgabenorientierten Programmierung von Industrierobotern beschreibt HUMBURGER (1998). Nach Auffassung des Autors ist eine rein sequentielle Lösung von Aufgaben in der Montage aufgrund deren Komplexität nur schwer möglich, weshalb er sich auf einen iterativen Ansatz fokussiert. Hierfür nutzt er eine *Blackboard*-Softwarearchitektur, die ihm Rücksprünge und Iterationen im Planungsprozess erlaubt. Das Architekturmuster ist nützlich bei Aufgaben, für die es keine vorhersagbaren Lösungsstrategien gibt und eine Lösungsfindung die Expertise aus mehreren Domänen umspannt. Hierbei beschreibt das *Blackboard* einen zentralen Datenspeicher, in dem – analog zu einer Kreidetafel – sowohl die Problemstellung, Teilprobleme wie auch Teillösungen enthalten sind. Zur Lösung der Teilprobleme existieren spezialisierte, voneinander unabhängige Softwaremodule, sog. *Wissensquellen*. Diese kommunizieren nicht direkt miteinander, sondern schreiben ihre Teillösungen zurück an das Blackboard. Da keine der Wissensquellen die Aufgabe eigenständig lösen kann, obliegt es einem *Kontroll*-Softwaremodul, die Auswahl der Wissensquellen entsprechend den zu lösenden Problemen zu übernehmen. (BUSCHMANN ET AL. 1996, S. 71 ff.)

HUMBURGER (1998) erweitert diesen Grundgedanken um zwei Kontrollebenen mit insgesamt vier Kontrollinstanzen: Die *Sequenz* gibt eine ungefähre Lösungsstrategie vor, um die Anzahl von Aufrufen der Wissensquellen gering zu halten. Die Wissensquelle *Ergebnisoptimierung* versucht, auf Basis der Produktionsgrößen (z. B. Taktzeit) bessere

Lösungen zu generieren, indem alternative Aufrufreihenfolgen probiert werden. Gerät die Planung in eine *Sackgasse*, so unternimmt das identisch bezeichnete Modul den Versuch, durch gezielte Rückschritte und Adaption der Aufrufreihenfolge neue Lösungen zu generieren. Welches dieser drei Module aus der Aufgabenebene gerade aktiv ist, wird durch die *Strategie*komponente bestimmt (vgl. Abbildung 3.7). Der Autor liefert ein methodisches Vorgehen, mit dem Wissensquellen strukturiert werden können. Der Aufbau der Softwarearchitektur und der Kontrollmodule wird jedoch nicht näher beschrieben, ebenso wenig wie die Funktionsweise der Konfliktlösung. Das Umweltmodell muss von Hand erstellt werden, ebenso wie die Zuweisung der Betriebsmittel zu Prozessschritten. Eine Umsetzung des Systems erfolgt rein simulativ, am Beispiel des Bahnschweißens von Robotern. Sensorische Prozesse und Rekonfigurationen werden nicht betrachtet.

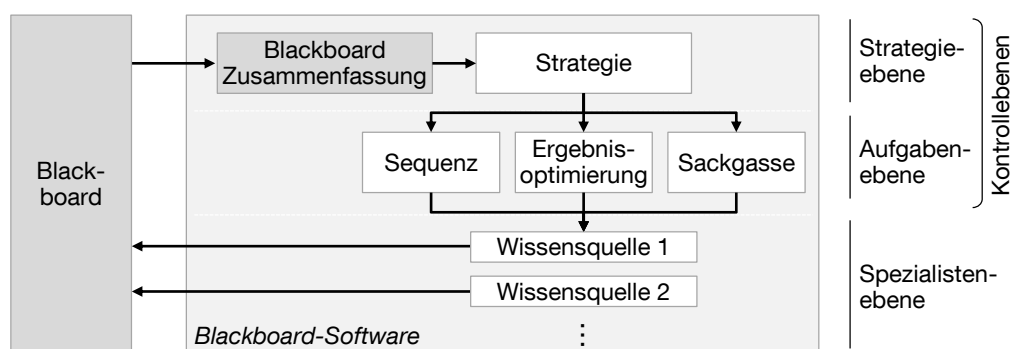


Abbildung 3.7: Aufbau des Blackboardsystems nach HUMBURGER (1998)

ARAI ET AL. (2003) und ARAI ET AL. (2001) entwickeln eine flexible roboterbasierte Montagezelle auf Basis des Plug&Produce-Gedankens. Eine Montageaufgabe wird in Teilprozesse zerlegt. Jeder Teilprozess besitzt ein eigenes Softwaremodul und sucht sich selbstständig ein freies Betriebsmittel, welches die Aufgabe durchführen kann. Hier kommt ein graphenbasierter Ansatz zu tragen, bei dem Montageaufgabe und Materialfluss mit Kosten (hier: Ausführungszeiten) beaufschlagt sind. Der günstigste Weg durch den Betriebsmittelgraph wird ausgeführt. Die Übergabebewegungen zwischen Robotern werden durch ein globales 2D-Arbeitsraumgitter generiert, bei dem jede Koordinate als exklusiv oder gemeinsam genutzt markiert ist. Gemeinsam genutzte Punkte stellen valide Übergabekoordinaten dar. Die Position und Arbeitsräume im Umweltmodell werden durch den Anwender dateibasiert importiert. Der Umgang mit sensorischen oder fehlenden Informationen steht nicht im Fokus.

Im Rahmen des „Porthos“-Forschungsprojekts entwickeln BRECHER ET AL. (2004) eine intuitive Software zur Programmierung von portablen Industrierobotern. Die Autoren

teilen das Konzept in eine Modellierungs- und Programmierphase. In der Modellierungsphase wird durch den Anwender ein Umweltmodell erstellt, das u. a. die Positionsdaten und Steuerungsanweisungen der Ressourcen an allen Einsatzorten enthält. In der späteren Programmierungsphase können Anwender ohne Expertenwissen aufgabenorientierte Abläufe zusammenstellen, inklusive logischer Kontrollflüsse. Zur Ausführung wird dann auf die hinterlegten Daten im Umweltmodell zurückgegriffen. Das System ist speziell auf Handhabungsaufgaben in kleineren und mittelständischen Unternehmen (KMUs) ausgerichtet. Eine Generierung des Aufgabenmodells wird nicht betrachtet, genauso wenig wie eine Methodik zur Herleitung der Funktionsprimitiva.

THOMAS (2008) und THOMAS & WAHL (2001) beschreiben eine durchgängige Werkzeugkette zur Planung, Evaluation und Ausführung von Montageaufgaben mit Industrierobotern. Montagereihenfolgen werden mit einem Assembly-by-Disassembly-Ansatz generiert, welcher verschiedene Metriken wie bspw. die Anzahl der Umorientierungen einbezieht, um optimale Sequenzen zu finden. Im Anschluss wird ein abstrakter Zustandsautomat des Roboters erstellt, in dem schrittweise jeder Montageprozess durch Fähigkeiten ersetzt wird, wie etwa Bewegen nach... oder Greifer öffnen. Deren Herleitung wird jedoch nicht beschrieben. Jede Fähigkeit besitzt sensorseitige Vorbedingungen, die für eine Ausführung erfüllt sein müssen, bspw. die Startposition. Zur Laufzeit überwacht eine Software alle Sensorinformationen der Roboterzelle. Entsprechen Fähigkeitsvorbedingungen einer Untermenge des aktuellen Sensorabbilds, so wird diese Fähigkeit aktiv geschaltet, etwa wenn eine bestimmte Position erreicht wird. Ein Interpreter wandelt die abstrakte Fähigkeit in konkrete Befehle für das Robotersystem um. Das System läuft ohne Zutun des Anwenders ab, benötigt hardwareseitig jedoch einen umfangreichen Sensoreinsatz und fokussiert vorrangig aktorische Prozesse.

SECKNER (2008) nutzt die AoP als Teil mehrerer Maßnahmen zur Steigerung der Flexibilität von Mikromontagesystemen. Das Konzept enthält ein Umweltmodell und ein Planungssystem. Das Umweltmodell unterteilt sich in ein Zustands- und Aktionsmodell. Während Ersteres die physischen Daten der Produktionsmittel beinhaltet, sind im Letzteren allgemeine Montageabläufe und deren zugehörige Prozessschritte hinterlegt. Die Prozessschritte bezeichnet Seckner als „Skripte“. Sie stellen eine klassenspezifische Abstraktion von Ressourcen dar, ähnlich den Fähigkeiten. Skripte enthalten Vorbedingungen, die für eine Ausführung erfüllt sein müssen, Aktionen, die während des Skripts ausgeführt werden und Folgezustände, die das Ergebnis eines Skripts darstellen. Der Softwareablauf wird durch das Planungssystem generiert, indem zur aktuellen

Aufgabe rückwärtsgerichtet Skripte gesucht werden, deren Vorbedingungen erfüllt werden können. Dies wird so lange wiederholt, bis die Vorbedingung der momentanen Ausgangslage des Montagesystems entspricht und somit gestartet werden kann.

Im Rahmen des Forschungsprojekts SIARAS untersucht BENGEL (2008) die automatisierte Ressourcenauswahl und Programmierung von Produktionsprozessen. Für die Aufgabenbeschreibung wird ein werkstückzentrierter Ansatz genutzt, bei dem in einem CAD-Tool die durchzuführenden Fähigkeiten und deren Parameter durch den Anwender ausgewählt werden. Die einzelnen Werkstückzustände werden anschließend manuell zu einem Reihenfolgegraphen verschalten. Auch das Umweltmodell wird händisch als Ontologie modelliert. Ein *Inferenzsystem* (engl. *Reasoner*) leitet auf Basis der Modelle geeignete Geräte ab. Der Abgleich von komplexeren Teilaufgaben, wie etwa der Bahnplanung, kann nicht mehr ontologiebasiert erfolgen, weshalb hierfür spezifische Softwaremodule, sog. „Utility Functions“, genutzt werden (deren Funktionsweise jedoch nicht näher erläutert wird). Nach dem erfolgten Matchmaking werden zusätzlich notwendige Bewegungs- und Transferprozesse der Ressourcen generiert. Die Evaluation erfolgt am Beispiel einer Mikromontagezelle, welche simulativ und real umgesetzt wurde. Die Prozesssteuerung der Komponenten erfolgt über einen Desktop-PC. (BENGEL 2008; BENGEL 2009; BENGEL 2010)

Im Nachfolgeprojekt ROSETTA erweitern NAUMANN ET AL. (2010) den ontologiebasierten Ansatz aus SIARAS zu einem „Knowledge Integration Framework“ (KIF), mit der Zielstellung der vereinfachten Programmierung von Industrierobotern. Das Framework besteht aus drei Schichten: In dem *Wissensspeicher* genannten Modul wird Prozess- und Entitätswissen aus verschiedenen Domänen hinterlegt, vgl. Abschnitt 3.2.3. Der Anwender gibt eine informelle Aufgabenbeschreibung in das KIF ein, der Wissensspeicher generiert hieraus einen abstrakten Prozessablauf in Form von Fähigkeitensequenzen. Fehlt Wissen bei der Aufgabendekomposition, so fragt das System bei dem Anwender nach. Der dahinter liegende Ansatz wird jedoch nicht weiter beschrieben. Die zweite Schicht, die *Codegenerierung*, nutzt diese Prozessbeschreibung, um einen hardware-spezifischen Zustandsautomaten für das Montagesystem zu erzeugen. Die Fähigkeiten im Zustandsautomat werden dabei durch das KIF oder den Anwender parametrisiert, um den Besonderheiten der jeweiligen Hardware gerecht zu werden. Der Code wird von der *Ausführungsschicht* genutzt, um den Prozessablauf zu steuern. Eine Evaluation des Systems beschreiben MALEC ET AL. (2013), indem aus einer Aufgabenbeschreibung Code für zwei verschiedene Robotermodelle eines Herstellers abgeleitet wird. Später

erweitern STENMARK ET AL. (2015) das KIF: Zur Steuerung wird ein Echtzeit-Linux-Betriebssystem genutzt, welches die Aufrufe für Roboter und Peripherie übernimmt. Zum Einsatz kommt ein eigens erstelltes Kommunikationsprotokoll. Der Aufbau des Umweltmodells innerhalb des KIF erfolgt größtenteils durch den Anwender und enthält vorrangig statische Informationen. Die Betrachtung von Rekonfigurationen und mehreren Montagestationen steht nicht im Fokus.

BÜSCHER ET AL. (2013) und MAYER ET AL. (2013) haben zum Ziel, eine selbstoptimierende Produktion auf Basis kognitiver Ansätze zu entwickeln. Sie teilen hierbei das System in einen *Offline-Planer* und einen *Online-Planer*. Der *Offline-Planer* verwendet einen Assembly-by-Disassembly-Ansatz, um die Montagereihenfolge mittels Produkt-CAD-Daten zu bestimmen. Der so entstandene Montagegraph enthält zusätzliche „Kostenmerkmale“, wie Ausführdauern und Anzahl der Werkzeugwechsel. Zur Laufzeit wird der *Online-Planer* zur Steuerung der Montage verwendet. Hierzu wird der aktuelle Zustand der Roboterzelle mit dem Graphen verglichen und der kürzeste Weg vom aktuellen zum Endzustand berechnet. Auf Basis des aktuellen Graphen erfolgt eine Montagesteuerung mittels eines kognitiven Softwareframeworks. Dieses kommuniziert direkt mit den Betriebsmitteln oder einem Anwender und erhält so Rückmeldung über die gesendeten Aufträge. Über einen „Reinforcement Learning“-Ansatz werden die Aktionen bewertet und die zugehörigen Kostenmerkmale entsprechend angepasst. Büscher et al. merken an, dass kognitive Systeme noch im Bereich der Grundlagenforschung stecken. Das Konzept konzentriert sich auf Handhabungs- und Fügevorgänge, eine Abstraktion der Gerätefunktionalität steht nicht im Fokus.

HUCKABY ET AL. (2013) beschreiben das Aufgabenmodell von Industrierobotern mit der „Planning Domain Definition Language“. Auf Basis einer einfachen Grammatik lassen sich hiermit Objektbeziehungen, Ausgangs- und Zielzustände beschreiben. Durch ein Inferenzsystem werden automatisiert die notwendigen Einzelschritte zur Zielerreichung generiert. Dabei entsteht nur eine generelle Sequenz von Fähigkeiten, die Parametrierung (z. B. Bewegungstrajektorien, Bauteilfeatures) muss von Hand eingegeben werden. Hierfür werden jedoch kein Modell oder eine Softwarelösung beschrieben. Die Abbildung der definierten Fähigkeiten auf reale Ressourcen und deren Evaluation wird nicht erläutert.

SCHOU ET AL. (2013) liefern einen weiteren fähigkeitsbasierten Ansatz zur Beschreibung von Montageaufgaben mit Industrierobotern. Die Autoren definieren eine dreistufige Abstraktionshierarchie: Auf dem untersten Level existieren *Geräteprimitive* (engl. *Device*

Primitives), die herstellerspezifische Befehle in eine Reihe von elementaren Funktionalitäten abstrahieren, wie etwa Greifen oder Bewegen nach XYZ. Auf der nächst höheren Ebene existieren *Fähigkeiten*, welche die Geräteprimitiva zu komplexeren Funktionalitäten kombinieren, bspw. Pick and Place. Die höchste Ebene stellen *Aufgaben* (engl. *Tasks*) dar, welche aus einer Sequenz von Fähigkeiten bestehen. Anwender können in einer Software eigenständig Tasks erstellen, indem sie aus einer Liste von Fähigkeiten eine Abfolge definieren – diese ist jedoch nur rein sequentiell gestaltbar. Die Programmierung geschieht in zwei Stufen: Zuerst werden für jede Fähigkeit notwendige Prozessparameter eingegeben, z. B. Geschwindigkeit. Im Anschluss erfolgt die Festlegung notwendiger Positionen durch ein Playback-Verfahren (vgl. Abschnitt 2.2.2). Zur Laufzeit werden die fertig definierten Tasks aufgerufen.

ANDERSEN ET AL. (2016) erweitern das Konzept auf den Teach-in von Fähigkeiten zur Gegenstandserkennung. Der Ablauf bleibt hierbei ähnlich. Die Behandlung der Fähigkeiten erfolgt auf einer generischen Ebene, die Herleitung spezifischer Fähigkeiten und Parameter wird nicht beschrieben. Eine Generierung fehlender Prozessschritte steht in den Arbeiten nicht im Fokus, genauso wenig wie eine Auswahl möglicher Ressourcen.

Im Kontext Cyber-physischer Produktionssysteme gehen OTTO & NIGGEMANN (2015) davon aus, dass sich Automatisierungsgeräte durch ihre Fähigkeiten selbst beschreiben und automatisch mit einem Produktionsprozess abgeglichen werden können. Sie stellen fest, dass in einer fähigkeitenbasierten Prozessmodellierung die Angabe von Parametern notwendig ist, um das universell ausgelegte Automatisierungssystem auf die momentane Aufgabe einzustellen. Als Beispiele werden u. a. Schaltschwellen von Sensoren und Zeitfaktoren der Anlage genannt. Hierfür entwickeln sie ein System zur automatischen Parameterfindung. Dieses basiert auf einem iterativen Optimierungsansatz, der mit einem Simulationsmodell der Anlage gekoppelt ist. Als Optimierungsgrößen werden ein niedriger Energieverbrauch und kurze Taktzeiten definiert. Das Konzept wird am Beispiel eines Flaschenabfüllprozesses evaluiert. Die hierfür notwendigen Fähigkeiten werden nicht beschrieben, weiterhin wird von einem existierenden Simulationsmodell der einzelnen Komponenten ausgegangen, das jeweils die Optimierungsgrößen zurückliefert.

BACKHAUS (2016) und BACKHAUS & REINHART (2015) entwickeln ein aufgabenorientiertes Programmiersystem für Montageanlagen. Auch hier erstellt der Anwender aus einer vordefinierten Prozessbibliothek (vgl. Abschnitt 3.2.2) einen Montageablauf. Montageprozesse enthalten dabei eine Liste der zur Umsetzung zugehörigen Geräte-

fähigkeiten. Deren Definition wird in einer ausführlichen Methodik dargelegt. Die Parameter von Fähigkeiten werden durch eine Sammlung typischer Geräte erstellt. Für eine Ausführung auf realen Geräten enthalten Fähigkeiten intern Zeiger, die auf Befehle und Schnittstellen der jeweiligen Ressourcen gerichtet sind. Eine Zuordnung der Prozessparameter auf die gerätetyp-abhängigen Fähigkeitenparameter wird nicht gegeben. Aufgaben- und Umweltmodell sind in AutomationML beschrieben. Hierbei wird auf den Arbeiten von KRUG (2013) aufgesetzt. Die Struktur des Aktionsplaners orientiert sich an den Bestandteilen von HUMBURGER (1998). Hierbei existieren mehrere Wissensquellen, die bspw. fehlende Sekundärprozesse hinzufügen oder den Gesamttablauf optimieren. Deren Beschreibung bleibt jedoch häufig allgemein. Die Auswahl von Ressourcen geschieht maßgeblich über eine Fähigkeiten-ID, die Beziehungen kombinierter Ressourcen stehen nicht im Fokus. Das Konzept wird auf Basis eines Simulationsmodells und eines realen Demonstrators evaluiert.

MICHNIEWICZ & REINHART (2016) und MICHNIEWICZ & REINHART (2014) beschreiben einen Ansatz zur aufgabenorientierten Programmierung von Cyber-physischen Roboterzellen auf Basis des PPRS-Konzepts. Das Aufgabenmodell, bestehend aus Füge- und Handhabungsprozessen, wird über einen entwickelten Assembly-by-Disassembly-Ansatz aus dem CAD-Modell des Produkts gewonnen. Das Konzept beherrscht mögliche Fügealternativen und prüft auf Kollisionen zwischen Bauteilen und Betriebsmitteln während des Fügevorgangs. Das Umweltmodell der Roboterzelle wird automatisiert erstellt. Hierbei wird als Voraussetzung angenommen, dass alle Ressourcen über einen Digital Twin verfügen. Ressourcen werden über Fähigkeiten beschrieben und besitzen als wichtiges Merkmal Schnittstellen (elektrisch, mechanisch etc.), über welche sie kombiniert werden können. Aus modellierten Arbeitsräumen und deren Überschneidungen wird ein Produktionsgraph erstellt, welcher die Transportmöglichkeiten in der Zelle beschreibt. Diese Informationen werden verwendet, um automatisiert Robotersysteme auf Basis der ermittelten Prozesse zu programmieren und einen Materialfluss der Bauteile in der Zelle zu steuern. Das Konzept wurde simulativ sowie am Beispiel eines Industrieroboters zur Montage von Lego-Produkten validiert.

3.4 Ansätze zur automatisierten Ressourcenauswahl in Montageanlagen

Neben Ansätzen zur aufgabenorientierten Programmierung beschäftigen sich viele Arbeiten mit der Fragestellung, wie Planungsaktivitäten in Montageanlagen stärker automatisiert werden können. Im Rahmen der Rekonfiguration steht insbesondere die *Auswahl* von geeigneten Ressourcen im Fokus vieler Forschungsarbeiten.

BENGEL (2010) und BENGEL (2008) untersucht dies im Bereich von Mikromontage-systemen. Das Konzept beruht auf Fähigkeiten sowie deren Parametern. Ein Abgleich der Parameter wird erwähnt, jedoch nicht im Detail beschrieben. Kommen für eine Zuweisung mehrere Ressourcen infrage, gibt der Autor drei Lösungsvorschläge an: Im einfachsten Fall wird die zuerst gefundene Ressource durch die Software ausgewählt. Im zweiten Fall trifft der Anwender eine Auswahl aus der Menge an geeigneten Geräten. Im dritten Fall werden Ressourcen nach Qualitätskriterien bewertet, zu welchem Grad sie die Anforderungen des Prozesses erfüllen und das entsprechend optimale Gerät selektiert. In dem Konzept von Bengel findet der dritte Fall Anwendung, die Funktionsweise wird jedoch nicht näher beschrieben. Die Einflüsse von kombinierten Ressourcen stehen hierbei nicht im Fokus.

KLUGE (2011) entwickelt ein Konzept zur Unterstützung bei der Grobplanung von Montagesystemen. Dabei werden Fähigkeiten aus einem Katalog ausgewählt und manuell ganzen Montagestationen zugewiesen, Geräte innerhalb der Stationen spielen eine untergeordnete Rolle. Über ein Softwaretool kann der Anwender einen Montageprozess erstellen, auf Basis dessen ein Softwaresystem mittels eines syntaktischen Fähigkeitenabgleichs geeignete Stationen vorschlägt. Um die Güte der Aussage zu erhöhen, werden Fähigkeiten mit Merkmalen hinterlegt (vgl. Abschnitt 3.2.2) und auf der Ressourcenseite spezifiziert. Während der Auswahlphase kann der Anwender die Merkmale geeigneter Stationen untereinander vergleichen – eine automatisierte Überprüfung findet nicht statt. Eine Besonderheit stellt in dem Konzept die kostenbasierte Beschreibung der Module und Transportwege dar. Dies wird für eine Bewertung verschiedener Konfigurationsalternativen benutzt.

JÄRVENPÄÄ (2012) nutzt Fähigkeiten zur Adaption von Produktionssystemen an geänderte Randbedingungen. Der Fähigkeitenablauf wird aus einem speziellen CAD-Programm heraus erstellt, in welchem Produktmerkmale mit Fähigkeiten

verknüpft werden. Der Abgleichsprozess geschieht dreistufig: Zuerst erfolgt ein Matchmaking mit vorhandenen Ressourcen auf semantischer Ebene. Anschließend wird ein Vergleich auf Parameterebene durchgeführt. Hierbei liegt ein Schwerpunkt auf dem Abgleich kombinierter Ressourcen. Die sich ergebenden Fähigkeiten- und Merkmalsänderungen des Verbunds werden über ein Regelsystem abgebildet, wie bspw. kombinierte Werkzeuglänge = Halterung + Fräserlänge - Einspanntiefe. Der Fokus liegt auf dem numerischen Abgleich von Prozessgrößen. Geometrische Eigenschaften werden mit einfachen analytischen Parametern angenähert (z. B. Bauteilhöhe = 50 mm), eine simulative Überprüfung erfolgt nicht. Existieren mehrere Ressourcenalternativen, werden in einem dritten Schritt die jeweils notwendigen Rekonfigurationsaufwände verglichen. Hierzu wird erneut ein Regelsystem angewandt, was durch den Anwender vorgegeben wurde. Die Auswahl der optimalen Ressourcenkombination wird nicht betrachtet, ebenso wenig wie eine weitergehende Programmierung des Systems.

CAVIN ET AL. (2013) beschreiben einen mehrstufigen Ansatz für einen fähigkeitenbasierten Abgleich in Montagessystemen. In einem ersten Schritt werden Fähigkeiten anhand ihres Typs verglichen. Anschließend erfolgt eine Überprüfung der übereinstimmenden Anzahl von Ein- und Ausgabeparametern, gefolgt von einem Parametertypenvergleich, wie etwa Name, Datentyp, Richtung etc. Abschließend erfolgt ein abstrakter geometrischer Abgleich, bei dem überprüft wird, ob sich die Montagestationen in der gleichen räumlichen „Zone“ befinden. Über dieses Hilfsmittel soll sichergestellt werden, dass die verschiedenen Systeme zusammenarbeiten können, wenn sie für einen Prozessablauf bestimmt werden. Später erweitern die Autoren ihren Ansatz auf einen rekursiven Algorithmus, der die Anforderungen zusammengesetzter Montageaufgaben stückweise in die Anforderungen untergeordneter Elementarfähigkeiten zerlegt (CAVIN & LOHSE 2014). Hierbei wird auf „Fähigkeitenmuster“ (engl. Pattern) zurückgegriffen. Diese stellen Transformationsregeln zwischen der zusammengesetzten Fähigkeit und deren Subfähigkeiten dar. So wird bspw. der Prozess Pick-and-Place in die Elementarfähigkeiten Bewegen und Greifen zerlegt. Wie eine Abbildung der Anforderungen eines zusammengesetzten Prozesses auf die Eigenschaften der Geräte und deren Subfähigkeiten transformiert wird, wird nicht näher beschrieben. Rekonfigurationen und konkurrierende Betriebsmittel stehen nicht im Fokus.

Dem PPRS-Ansatz folgend, definieren LOSKYLL (2013) und LOSKYLL ET AL. (2012) mehrere Ontologien, um eine kontextbasierte Orchestrierung von Feldgeräten durchzuführen. Die Ontologien stehen in einer hierarchischen Beziehung zueinander. Hierbei

sind sowohl Geräteklassen, die aktuelle Fabrikstruktur als auch Fähigkeiten semantisch abgebildet. Der Produktionsprozess ist auf Basis von Fähigkeiten beschrieben und wird durch den Anwender in einem Editor modelliert. Die Suche nach geeigneten Geräten geschieht dynamisch während der Herstellung des Produkts, wodurch vorab keine Aussagen zu den Betriebsmittelbelegungen gemacht werden können. Den Abgleich übernimmt ein sog. „Matchmaker“, welcher die Beschreibung der Fähigkeit sowie dessen Ein- und Ausgabeparameter abgleicht. Es wird eine Rangliste theoretisch geeigneter Gerätefähigkeiten erstellt, aus welcher ein kontextsensitives Bewertungsmodul anhand verschiedener Umweltinformationen (z. B. Gerätezustand, Energieverbrauch) das jeweils beste Gerät auswählt. Die Gewichtung der Bewertung kann durch den Anwender festgelegt werden, die unterlagerten Metriken und Abläufe des Matchmakings werden jedoch nicht abschließend erklärt, bspw. die Erstellung von Materialflüssen oder die Prüfung von geometrischen Zugänglichkeiten.

KEDDIS ET AL. (2014) und KEDDIS ET AL. (2013) beschreiben in ihrer Arbeit ein automatisiertes Planungskonzept für adaptierbare Produktionssysteme. Der Produktionsablauf wird erneut durch den Anwender auf Basis von Fähigkeiten erstellt. Das Ressourcenmodell enthält Montagestationen, welche wiederum aus Modulen aufgebaut sind. Module stellen Fähigkeiten bereit, die für eine Umsetzung notwendigen Geräte werden vorab herkömmlich programmiert. Eine Zuweisung der Module erfolgt automatisiert. Fähigkeitenparameter werden hierbei berücksichtigt, eine weitergehende Erläuterung bleibt jedoch aus. Eine Besonderheit stellt die Überprüfung möglicher Materialflüsse zwischen verschiedenen Stationen dar. Hierbei werden auch Flussrichtungen beachtet, jedoch keine exakte mechanisch/geometrische Kompatibilität der Schnittstellen. Auf Grundlage einer Kürzesten-Weg-Suche werden geeignete Stationen dem Produktionsplan zugewiesen und der Ablauf durch eine zentrale Steuerungseinheit koordiniert. Das Konzept wurde mit mehreren Produktvarianten an einem realen Demonstrator erfolgreich getestet.

PFROMMER ET AL. (2015) fokussieren sich auf eine dynamische Ressourcenplanung und -orchestrierung im Rahmen eines PPRS-Konzepts. Das Aufgaben- und Umweltmodell wird in einem „Asset Management System“ (AMS) genannten Softwaremodul gespeichert. Produktionsaufträge werden durch den Anwender im AMS eingelastet, das im Anschluss nach verfügbaren Gerätefähigkeiten sucht. Bei Erfolg wird auf dem Gerät eine um die aktuellen Produktionsparameter angepasste Fähigkeit instanziiert, „Aktion“ genannt. Die kurzfristige Planung wird durch ein MES ausgeführt, welches die

Produktionsziele mit den angebotenen Aktionen vergleicht und entsprechende Aufrufe startet. Fehlende Hilfsprozesse, wie etwa Materialtransport, werden im Konzept beachtet und teilweise automatisch generiert. Bei spontan auftretenden Produktionsänderungen erfolgt eine dynamische Neuplanung, bspw. durch Anpassung der Transportwege. Die Funktionsweise der jeweiligen Abgleiche wird nicht näher erläutert, ebenso wenig wie die Struktur des AMS.

Im Rahmen der Mensch-Roboter-Kollaboration entwickeln MÜLLER ET AL. (2016a) einen Ansatz für die Zuteilung von Aufgaben zwischen den Teilnehmern. In dem Konzept wird von einer detaillierten Beschreibung des Montageprozesses ausgegangen, der auf Basis von Fähigkeiten modelliert ist. Für jeden Schritt werden durch den Anwender Parameter definiert (z. B. Einpresskraft), die im Anschluss sukzessive mit den Fähigkeiten von Roboter und Mensch abgeglichen werden. Für die Fähigkeitenbestimmung des Roboters wird dessen Peripherie sowie das Layout der Zelle analysiert, ein genaueres Vorgehen wird jedoch nicht beschrieben. Die Autoren geben an, dass eine quantitative Bestimmung von menschlichen Fähigkeiten schwer realisierbar ist, weshalb dieser Abgleich manuell durchgeführt wird. Rekonfigurationen stehen hierbei nicht im Fokus.

MÜLLER ET AL. (2016b) konzentrieren sich auf die Planung Cyber-physischer Montageanlagen. In dem Konzept bestehen Produkte aus einer Aggregation von Bauteilen, welchen wiederum Montageprozesse mit spezifischen Anforderungen zugeordnet sind. Diese werden vom Anwender eingegeben, der auch eine Montagereihenfolge festlegt. Die Bestimmung der Betriebsmittel erfolgt einer ähnlichen Strukturierung: In einem Simulationstool werden allgemeine Montagestationen erstellt, indem durchzuführende Prozesse (*Aufgaben*) und notwendige Taktzeiten mittels des Montageplans zugewiesen werden. Im Anschluss erfolgt die Ausdetaillierung der Stationen durch Montagemodule. Diesen sind *Prozessschritte* zugeordnet, über die Stationsaufgabe können so programmunterstützt geeignete Module vorgeschlagen werden. Hierbei wird über ein einfaches Arbeitsraummodell auch überprüft, ob das jeweilige Modul geometrisch in die Station passt. Darüber hinaus beschäftigen sich die Autoren mit der Aktualisierung des digitalen Planungssystems bei der Anpassung von realen Anlagen (MÜLLER ET AL. 2016c). Hierzu stellen sie einen Demonstrator mit einem Trackingsystem aus, welches an den Montagemodulen angebrachte Marker registriert. Dadurch ist eine ungefähre Zuordnung von Modulen zu den Arbeitsräumen von Stationen möglich. Den Aufbau des Prozess- und

Ressourcenmodells stellen die Autoren in einem umfangreichen Klassendiagramm vor. Einzelne Geräte stehen in dem Konzept nicht im Fokus.

Neben der aufgabenorientierten Programmierung Cyber-physischer Roboterzellen (Abschnitt 3.3) fokussieren sich MICHNIEWICZ & REINHART (2016) und MICHNIEWICZ & REINHART (2014) auf eine Auswahl geeigneter Betriebsmittel für eine vom Produkt geforderte Fähigkeitensequenz. Hierbei stehen Fügeprozesse im Mittelpunkt, sensorische Fähigkeiten werden nicht betrachtet. Das Abgleichskonzept besteht aus mehreren Stufen: Zuerst erfolgt eine syntaktische Überprüfung des erforderlichen Prozessschritts (z. B. Greifen) mit entsprechenden Fähigkeiten. Anschließend wird ein Vergleich des mechanischen Schnittstellentyps zwischen Produkt und Ressource gestartet. Für geeignete Ressourcen wird deren CAD-Modell geladen und an den Schnittstellen positioniert, um Kollisionen während des Fügevorgangs zu erkennen. Auftretende Wechselwirkungen (z. B. Reduktion der Traglast bei Anbringen eines Endeffektors) werden beachtet und mit Produktanforderungen verglichen, der Ablauf jedoch nicht im Detail beschrieben. Nach dem Abgleich prozessnaher Ressourcen werden geeignete Materialflussmittel gesucht. Hierzu ermittelt ein Algorithmus aus der Überschneidung der jeweiligen Arbeitsräume kollisionsfreie Übergabepunkte. Kann zu einem Prozessschritt keine zugehörige Fähigkeit gefunden werden, wird der Anwender über die nicht erfüllte Anforderung informiert.

3.5 Bewertung

Nachdem eine Übersicht über den Stand der Erkenntnisse gegeben worden ist, können die vorgestellten Arbeiten hinsichtlich ihrer Erfüllung der Problemstellung aus Kapitel 1 und 2 bewertet werden.

Automatisierte Vernetzung von Betriebsmitteln

Echtzeitfähigkeit spielt in der Produktionstechnik eine unverzichtbare Rolle (vgl. Kapitel 2). Ein echtzeitfähiges System muss hierbei sowohl eine deterministische Steuerung als auch Kommunikation besitzen. Dabei besteht die Kommunikation in heutigen Fabriken aus verschiedenen, hierarchischen Netzen (Abschnitt 2.2.1), weshalb Konzepte zur automatisierten Vernetzung hiermit umgehen müssen. Als weitere Randbedingung der Montage ist zu nennen, dass in den Anlagen eine Vielzahl von „primitiven“ Geräten

verbaut ist, welche keine eigenen Beschreibungsmittel oder Server bereitstellen (z. B. pneumatische Greifer). Diese Geräteklasse sollte von Vernetzungsansätzen berücksichtigt werden, um kompatibel zu heutigen Fabriken zu sein.

Für eine Übertragbarkeit des Konzepts auf reale industrielle Gegebenheiten sollten die umgesetzten Konzepte dabei auf etablierte Standards in der Automatisierungstechnik aufbauen und industrielle Standardkomponenten einsetzen. Tabelle 3.2 zeigt die so entstandene Bewertung in der Übersicht.

Tabelle 3.2: Bewertung von Ansätzen zur automatisierten Vernetzung

Kriterien	Bus-Master-Konfiguration						SOAs			
	GRUNWALD ET AL. (2008)	KRUG (2013)	DÜRKOP ET AL. (2014)	HODEK (2013)	ANDREN ET AL. (2012)	REGULIN ET AL. (2015)	BOHN ET AL. (2006)	LOSKYLL ET AL. (2012)	PFROMMER ET AL. (2015)	NAUMANN & VERL (2008)
Echtzeitfähige Kommunikation	●	●	●	○	●	●	○	○	◐	○
Echtzeitbasiertes Steuerungssystem	◐	●	●	●	●	○	-	○	○	◐
Durchsuchen von Netzwerkhierarchien möglich	○	○	○	○	○	○	◐	○	○	-
Einbindung primitiver Systeme möglich	○	◐	○	○	○	○	○	○	○	○
Umsetzung auf Basis etablierter Standards in der AT	◐	◐	●	◐	◐	●	◐	◐	●	○
Nutzung etablierter Automatisierungskomponenten	◐	◐	◐	◐	○	◐	○	◐	-	◐

○: nicht im Fokus, ◐: teilweise im Fokus, ●: im Fokus, -: keine Aussage möglich

In der Bewertung fällt auf, dass die meisten Arbeiten von einem flachen und uniformen Netzwerk ausgehen, in welchem intelligente Komponenten sich auf eine bestimmte Art selbst beschreiben können. Diese Annahmen sind im heutigen Stand der Technik nicht gegeben. Insbesondere bei dienstorientierten Konzepten erfolgt eine dauerhafte Kommunikation meist auf Ethernet-Basis, weshalb eine Echtzeitfähigkeit nicht erreicht werden kann. Als Steuerung der Produktionsabläufe werden oft Windows-basierte Rechner eingesetzt, die in der Feldgeräteebene wegen ihrer nicht-deterministischen Arbeitsweise jedoch nur in seltenen Fällen Anwendung finden.

Digitale Modelle zur Erhöhung der Interoperabilität

Digitale Modelle stellen im Sinne des PPRS-Konzepts Hilfsmittel dar, um die beschriebenen Ansätze zur Vereinfachung der Vernetzung, Programmierung und Planung realisieren zu können. Die einzelnen Domänen unterscheiden sich sehr stark, weshalb eine gemeinsame kriterienbasierte Bewertung hier nicht sinnvoll erscheint.

Allgemein ist festzuhalten, dass die geometrische Produktbeschreibung anhand von CAD-Daten de facto standardisiert ist. Dies gilt weniger für die Beschreibung produktseitiger Merkmale. eCl@ss zeigt hier Ansätze, bleibt jedoch bezüglich Automatisierungstechnischer Merkmale unzureichend. Bei der Beschreibung standardisierter Prozessschritte bzw. geräteseitiger Fähigkeiten existieren mehrere Normen, die hierfür verwendet werden können, dabei jedoch Überschneidungen aufweisen und Unzulänglichkeiten im Bereich der Sensorik haben. Eine strukturierte Beschreibung der fähigkeitsbezogenen Parameter fehlt hierbei gänzlich. Existierende Forschungsarbeiten basieren größtenteils auf diesen Normen, weshalb sie deren Defizite erben. Für eine Beschreibung gerätetechnischer Merkmale zur Umsetzung PPRS-orientierter Ansätze existiert bis dato kein anerkanntes Informationsmodell. Auch wenn Gemeinsamkeiten zwischen den Ansätzen zu erkennen sind, etwa die Geometrie und Kinematikbeschreibung mit COLLADA, werden die meisten Modelle von Grund auf neu erstellt und deren Struktur nur abstrakt beschrieben.

Aufgabenorientierte Programmierung von Montageanlagen

Die AoP ist ein hilfreiches Werkzeug zur Reduktion von Programmieraufwänden in Montageanlagen, indem Teile des Anwenderwissens durch ein softwareseitiges Expertensystem erzeugt werden. Die Möglichkeit zur Aufgabenabstraktion und der damit einhergehenden Generierung fehlender Prozesse ist hierbei zentraler Bestandteil und somit Bewertungskriterium. Diesbezüglich gilt es zu untersuchen, inwieweit unterschiedliche Problemarten abgedeckt werden, die während der Programmierung gelöst werden müssen (z. B. Zuweisungs-, Optimierungs- oder Interaktionsprobleme). Die Erstellung der dafür notwendigen Aufgaben- und Umweltmodelle bedingt Aufwände im Unternehmen, weshalb eine Automatisierung diesbezüglich untersucht wird. Dies schließt auch den Umgang mit fehlenden Modellinformationen mit ein. Die Umsetzung

der AoP-Konzepte sollte dabei auf möglichst unmodifizierten und etablierten Automatisierungskomponenten und -steuerungen erfolgen, um eine industrielle Übertragbarkeit zu gewährleisten. Tabelle 3.3 zeigt die so entstandene Bewertung in der Übersicht.

Tabelle 3.3: Bewertung von Ansätzen zur aufgabenorientierten Programmierung von Montageanlagen

Kriterien	HUMBURGER (1998)	ARAI ET AL. (2003)	BRECHER ET AL. (2004)	THOMAS (2008)	SECKNER (2008)	BENGEL (2010)	ROSETTA (2015)	BÜSCHER ET AL. (2013)	HUCKABY ET AL. (2013)	SCHOU ET AL. (2013)	BACKHAUS (2016)	MICHNIEWICZ (2016)
Betrachtungsebene Automatisierungsgeräte	●	●	◐	◐	◐	◐	●	●	●	●	●	●
Ganzheitliche Programmierung von Montageanlagen möglich	○	○	○	○	○	○	◐	◐	○	◐	◐	○
Möglichkeit zur Aufgabenabstraktion	●	●	○	●	◐	●	●	○	●	○	●	●
Problemheterogenität (Zuweisung, Optimierung, Interaktion etc.)	●	○	○	◐	○	◐	◐	◐	○	○	◐	◐
Automatisierter Aufbau des Aufgabenmodells	◐	-	○	●	○	○	◐	●	●	○	○	◐
Automatisierter Aufbau des Umweltmodells	○	○	○	○	○	○	◐	○	○	○	◐	◐
Umgang mit fehlenden Informationen	○	○	○	○	○	○	◐	◐	○	○	○	◐
Fokus auf industrielle Steuerungsarchitekturen	○	○	-	◐	○	○	◐	○	○	◐	●	◐
Nutzung etablierter Automatisierungskomponenten	-	◐	◐	●	◐	◐	●	◐	-	●	●	●

○: nicht im Fokus, ◐: teilweise im Fokus, ●: im Fokus, -: keine Aussage möglich

Arbeiten im Bereich der AoP sind stark auf Industrieroboter fokussiert. Eine Programmierung der Abläufe gesamter Montageanlagen erfordert jedoch ein Zusammenspiel einer Vielzahl an weiteren Geräten aus dem sensorischen und indirekten Bereich (wie Halteeinrichtungen), weshalb hier Defizite bestehen. Die Generierung fehlender Prozesse (z. B. Transferbewegungen) wird durch mehrere Arbeiten betrachtet, eine Beschreibung der internen Abläufe und Konzepte bleibt jedoch vage. Die Heterogenität der auftretenden Probleme steht nur bei wenigen Arbeiten im Fokus. Dabei gehen viele Konzepte von einem vollständig modellierten System aus, der Umgang mit fehlenden Informationen wird kaum betrachtet. Zur Reduktion von Aufwänden bei der Modellerstellung widmen sich mehrere Arbeiten der automatisierten Generierung des Aufgabenmodells, wobei hier Fügeoperationen im Fokus stehen. Der automatisierte Aufbau des Umweltmodells findet kaum Berücksichtigung. Eine Umsetzung und Eva-

luation erfolgt hierbei mit teilweise stark modifizierten Automatisierungsgeräten und nicht-automatisierungsgerechten Steuerungen.

Automatisierte Ressourcenauswahl in Montageanlagen

Eine automatisierte Auswahl von Ressourcen bezüglich eines definierten Produkts und Produktionsprozesses stellt eine wichtige Aktivität beim Systementwurf von Montageanlagen dar. Als Bewertungskriterium wird festgelegt, inwieweit sich eine Auswahl auf den Rekonfigurationsfall und damit auf bereits existierende Anlagen und Strukturen bezieht. Die Granularität der Ressourcen-Suche soll dabei einzelne Automatisierungsgeräte abbilden können. Konzeptuell setzen alle untersuchten Arbeiten auf eine Art von Fähigkeitenbeschreibung, weshalb dies nicht als Kriterium aufgenommen wird. Es existieren innerhalb des Betrachtungsbereichs (Kapitel 1) jedoch verschiedene zu untersuchende Abgleichsdomänen, welche eine Validität des Ergebnisses entsprechend verbessern: Notwendige Prozessparameter müssen mit verfügbaren Geräteeigenschaften verglichen werden, ebenso wie eine technologische Eignung des Automatisierungsgeräts. Dies gilt neben direkt den Prozess betreffenden Komponenten auch für Materialflussaufgaben zwischen Montagestationen. Ein Planungssystem muss diese heterogenen Problemstellungen möglichst behandeln können.

Nachdem Automatisierungsanlagen häufig aus kombinierbaren Einzelkomponenten erstellt werden (bspw. Roboter + Endeffektor, vgl. Kapitel 2), stellt eine Untersuchung bez. der Merkmalsveränderung in Geräteverbänden einen weiteren Punkt dar. Selbiges gilt für den Umgang mit möglichen Ressourcenalternativen. Als abschließendes Kriterium wird untersucht, ob das Konzept neben der Ressourcenauswahl auch Informationen zur Programmierung der Anlage erzeugen kann. Tabelle 3.4 zeigt die so entstandene Bewertung in der Übersicht.

Viele der gezeigten Arbeiten gehen von einer Neuplanung der Anlage aus und behandeln nicht bereits existierende Ressourcen innerhalb einer Rekonfiguration. Die Betrachtungsebene liegt dabei nur teilweise auf Automatisierungsgeräten, oftmals findet eine Zuweisung auf der abstrakteren Montagesystemebene statt. Während sich die meisten Konzepte auf einen Abgleich zwischen semantisch identischen Fähigkeiten und Prozessschritten konzentrieren, werden zugehörige Parameter weniger genau behandelt. Zwar geben mehrere Arbeiten diesbezüglich an, ein Matchmaking durchzuführen, der Umgang mit unterschiedlichen Parameterarten, unvollständigen Parametern oder verschiedenen Wirkprinzipien steht jedoch nicht im Fokus. Die Untersuchung dynamisch

Tabelle 3.4: Bewertung von Ansätzen zur automatisierten Ressourcenauswahl in Montageanlagen

Kriterien	BENGEL (2010)	KLUGE (2011)	JÄRVENPÄÄ (2012)	CAVIN & LOHSE (2014)	LOSKYLL (2013)	KEDDIS ET AL. (2014)	PFROMMER ET AL. (2015)	MÜLLER ET AL. (2016a)	MÜLLER ET AL. (2016b)	MICHNIEWICZ (2016)
Fokus auf Rekonfiguration (existierende Anlage)	○	○	●	○	◐	●	◐	○	●	●
Betrachtungsebene Automatisierungsgeräte	●	○	●	●	●	○	●	●	○	●
Abgleich auf Parameterebene	◐	○	◐	●	◐	-	●	◐	◐	●
Heterogenität der Abgleichsdomänen	◐	○	◐	○	○	○	○	○	○	○
Abgleich des Materialflusses	◐	◐	◐	○	○	◐	-	○	◐	◐
Abgleich von dyn. kombinierten Ressourcen	○	○	●	○	○	○	○	◐	○	◐
Konkurrierende Ressourcen / Ressourcenalternativen	◐	◐	◐	○	●	○	○	●	◐	●
Programmierung möglich	●	○	○	-	◐	◐	-	○	○	◐

○: nicht im Fokus, ◐: teilweise im Fokus, ●: im Fokus, -: keine Aussage möglich

kombinierbarer Ressourcen werden von MÜLLER ET AL. (2016a) und MICHNIEWICZ & REINHART (2016) genannt, hierbei jedoch nicht im Detail erläutert. Die nach einer Auswahl zu erfolgende Programmierung der Komponenten wird nur von wenigen Ansätzen als Bestandteil aufgenommen.

4 Zielsetzung und Lösungsansatz

4.1 Handlungsbedarf

Als wesentlichen Bestandteil für eine veränderungsfähige Produktion sieht die PLATTFORM I4.0 (2016, S. 17 f.) „[...] autonom kommunizierende und interoperable Module, die sich weitestgehend selbstständig anpassen“. Die gezeigten Forschungsarbeiten weisen in dieser Richtung gute Ansätze auf, besitzen jedoch auch Defizite, die einer Anwendbarkeit in Montageanlagen entgegenstehen (vgl. Abschnitt 3.5). Insbesondere müssen noch folgende Herausforderungen adressiert werden:

1. *Vernetzung*:
 - Automatisierte, echtzeitfähige Einbindung von komplexen und primitiven Komponenten in hierarchischen Netzwerken
 - Automatisierte Bereitstellung eines synchronen digitalen Abbilds der Komponenten (Digital Twin¹)
2. *Modellierung*: Erstellung eines standardisierten Digital Twins zur Selbstbeschreibung von Komponenten. Wichtiger Bestandteil ist hierbei eine umfassende Fähigkeitendefinition, welche nicht-aktorische Komponenten einschließt.
3. *Ressourcenauswahl*: Ein Softwareframework, welches dem Anwender geeignete Ressourcen zur Durchführung der Aufgabe (Programmierung) vorschlägt. Das Framework muss die Verschiedenartigkeit der Matchingdomänen unterstützen, um die Validität der Aussage zu erhöhen und hierbei den Digital Twin der Ressourcen nutzen. Weiterhin soll es möglich sein auch Fähigkeiten von menschlichen Arbeitskräften abgleichen zu können, um damit die Eignung manueller Arbeitsstationen zu überprüfen.

¹Im Kontext dieser Arbeit wird von einem umfassenden Echtzeitabbild ausgegangen, welches auch Simulations- und Verhaltensmodelle enthalten kann. Aus diesem Grund wird von einem Digitalen Zwilling (nicht Schatten) gesprochen.

4. *Aufgabenorientierte Programmierung*: Ein Softwareframework, welches die anfallenden heterogenen Aufgaben zur Programmierung unterschiedlicher Automatisierungsgeräte in Montageanlagen lösen kann. Dies beinhaltet auch den Umgang mit fehlenden Informationen.

Auch die PLATTFORM I4.0 (2016, S. 17 f.) sieht den aktuellen Forschungsbedarf u. a. in der *Selbstbeschreibung* von Modulfähigkeiten und Prozessabläufen, der *automatisierten Verhandlung* sowie einem *erleichterten Rekonfigurationsengineering* zur kurzfristigen Planung auf *Basis des Digitalen Zwillings*, was mit dem vorgestellten Handlungsbedarf korreliert.

4.2 Zielsetzung

Dem Handlungsbedarf folgend, soll ein holistisches Rekonfigurationskonzept für Montageanlagen gefunden werden, welches die automatisierte *Ressourcenvernetzung*, *-auswahl* und *-programmierung* verbindet. Das Konzept soll die im Handlungsbedarf aufgezeigten Herausforderungen adressieren.

Als grundlegender Ansatz wird die Vision der Cyber-physischen Systeme auf Automatisierungskomponenten übertragen: Feldgeräte sollen selbstständig kommunizieren, sich beschreiben und eigenständig an gestellte Aufgaben anpassen können (vgl. Abschnitt 2.3). Den PPRS-Konzepten aus Abschnitten 3.2 – 3.4 folgend, sollen diese „Cyber-physischen Feldgeräte“ automatisiert mit einer vom Anwender gestellten Produktionsaufgabe abgeglichen werden können, um geeignete Ressourcen zu ermitteln. Das Ergebnis wird für eine automatisierte Programmierung der Montageanlage verwendet.

Als Nebenbedingung soll das Konzept mit bereits existierenden Produktionssystemen verwendet werden können, um eine leichte Übertragbarkeit in die industrielle Anwendung zu schaffen (Retrofit). PLATTFORM I4.0 (2016, S. 17) sieht hierin eine wichtige Zielstellung. Dies erfordert den Einsatz von verbreiteten, unmodifizierten Standardgeräten und -technologien. Nachdem aktuell jedoch keine Feldgeräte existieren, welche die geforderten cyber-physischen Eigenschaften aufweisen, müssen diese über externe Softwaremodule bereitgestellt und mit den entsprechenden Geräten verlinkt werden.

Als technische Grundlage wird eine dienstorientierte Architektur verwendet. Sie bietet eine anerkannte Lösung zum Aufbau von lose gekoppelten Softwaresystemen in der IT-Ebene (TAKAI 2017; MATHAS 2008). Von ihrer Zielsetzung (Interoperabilität) und ihren Charakteristika (z. B. standardisierte Selbstbeschreibung) ähnelt sie CPS. Weiterhin bilden SoAs nach OASIS (2016 S. 8 ff.) ein mächtiges Werkzeug zum Abgleich von Bedarfen mit Fähigkeiten sowie der Kombination von Fähigkeiten.

Teilziele

Zusammengefasst ergeben sich drei Teilziele für diese Arbeit:

1. Konzept zur automatisierten Vernetzung von Feldgeräten und synchronen Bereitstellung eines dienstebasierten Digital Twins (→Kapitel 5)
2. Definition eines Fähigkeitenmodells zur lösungsneutralen Beschreibung von Montageprozessen. Das Modell dient als Bindeglied zwischen der Anlagenseite (angebotene Fähigkeiten) und den durchzuführenden Prozessen (notwendige Fähigkeiten) (→Kapitel 6)
3. Konzeption eines Softwareframeworks zur aufgabenorientierten Ressourcenauswahl und Programmierung auf Basis des zu entwickelnden dienstebasierten Digital Twins (→Kapitel 7)

Referenz-Anwendungsfall

Um die Forschungskonzepte repräsentativ zu gestalten, wird ein industrieller Anwendungsfall geschaffen, anhand dessen Lösungsansätze induktiv entwickelt werden. Im Sinne des Betrachtungsraums der Rekonfiguration wird von einer existierenden Montageanlage ausgegangen, bei der Anpassungen aufgrund von geänderten Randbedingungen vorgenommen werden müssen.

Der Anwendungsfall besteht aus den PPR-Domänen des herzustellenden *Produkts*, dessen zugehörigen *Montage-* und *Prüfprozessen* sowie der *Anlagenstruktur*.

Ausgangsbasis bildet ein *Produkt*, welches aus zwei Baugruppen besteht und innerhalb der Endmontage gefügt wird. Die Anlieferung des Fügebasis-Bauteils erfolgt automatisiert, ebenso wie der Fügevorgang. Anschließend wird zu Qualitätssicherungszwecken ein automatisierter Funktionstest durchgeführt und das Ergebnis dokumentiert. In

Anlehnung an die von SAYLER & DILLMANN (2011) durchgeführte Analyse häufiger Bauteilmerkmale in der Automobilindustrie wird von einem Produkt ausgegangen, dessen Abmaße ca. 100 mm betragen, als Bestandteile Metall- und Kunststoffelemente (Elektronik) aufweist und eine zylindrische Form besitzt. Dies kann bspw. ein Elektroantrieb sein.

Bei den durchzuführenden *Prozessen* wird darauf geachtet, dass Elemente aus jeder Hauptkategorie der Montagefunktionen nach LOTTER & WIENDAHL (2012, S. 2) genutzt werden (vgl. Abschnitt 2.1.1). Neben den voranstehend genannten Fügeoperationen kommen Handhabungsaufgaben des Bauteiltransports (Bewegen, Sichern) und Sonderoperationen wie etwa eine Spannungskontaktierung zur Funktionsprüfung hinzu. Zusätzlich sind Kontrollfunktionen zur Durchführung der Handhabungsvorgänge notwendig (Lagemessung der Bauteile, Anwesenheitsprüfung) sowie zur abschließenden Qualitätssicherung (z. B. Spaltmaß- und Drehmomentenmessung).

Auf *Ressourcenseite* wird eine teilautomatisierte Montageanlage angenommen, welche aus mehreren modularen Montagestationen besteht und einen Großteil der typischen Geräteklassen in der Automatisierungstechnik abdeckt (vgl. Abschnitt 2.1.4). Das Produkt wird mittels eines Werkstückträgers (WST) auf Förderbändern zwischen den Stationen transportiert und über Haltezyylinder an prozesseitig relevanten Orten im Raum gestoppt und fixiert. Der Modulgedanke erfordert pro Station eine autarke Modulsteuerung (SPS) sowie Sensoren zur Anwesenheitsprüfung und Identifikation der Werkstückträger. Hierfür werden Lichttaster und Barcodescanner festgelegt. Der Fügevorgang wird durch ein Robotersystem mit Formschlussgreifer vorgenommen. Dieser misst die Anwesenheit der Produktoberteile in einem Lagerplatz und fügt diese mit dem Fügebasisteil. Abbildung 4.1 zeigt den beschriebenen Anwendungsfall in der Übersicht.

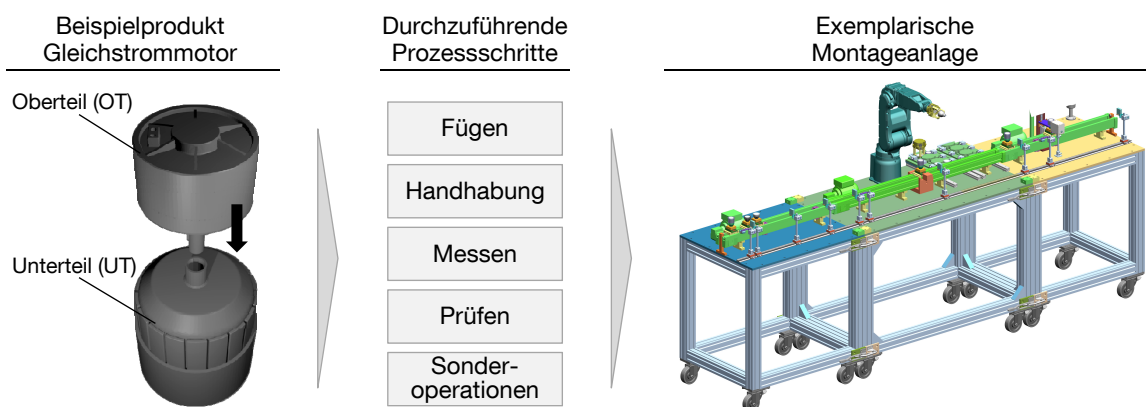


Abbildung 4.1: Referenz-Anwendungsfall

4.3 Vorstellung des Lösungsansatzes aus Anwendersicht

Initial wird ein Abbild der aktuellen Fähigkeiten des Montagesystems benötigt (Umweltmodell bzw. Angebotsebene). Eine in Kapitel 5 beschriebene Software generiert dieses, indem nach angeschlossenen Feldgeräten gesucht wird, diese automatisch in die zugehörigen Steuerungssysteme integriert werden und für jede Ressource ein zugehöriger Digital Twin veröffentlicht wird (Abbildung 4.2, ①a).

Parallel hierzu beschreibt der Anwender den Montageprozess auf Basis von Fähigkeiten (①b). Dieser ist rein produktbezogen und enthält keinerlei Anlagendaten. Jede Fähigkeit wird durch den Anwender näher spezifiziert, bspw. zur Vorgabe notwendiger Taktzeiten, Prozessgrößen oder Qualitätsmerkmale. Dies stellt das Aufgabenmodell bzw. die Anforderungsebene dar (Kapitel 6).

Anschließend gleicht das in Kapitel 7 zu erstellende Framework automatisiert den modellierten Prozess mit dem digitalen Fabrikabbild ab, um geeignete Ressourcen für die Durchführung der Prozessschritte zu finden. Dies beinhaltet auch die dynamische Kombination von passenden, aber noch nicht verbundenen Ressourcen, wie etwa Roboter + Endeffektor, sowie die Einbeziehung von Arbeitskräften in das Matchmaking. Je Prozessschritt wird dem Anwender aufgrund seiner spezifizierten Anforderungen eine Liste geeigneter Ressourcen geliefert. Hieraus kann ein bevorzugtes Produktionsmittel gewählt werden. Nach Festlegung dieser prozessnahen Ressourcen kann automatisiert der erforderliche Materialfluss zwischen Geräten und Montagestationen berechnet werden. Auch hierbei wird eine Eignungsprüfung der Verkettungsmittel durchgeführt und bei Alternativen dem Anwender präsentiert. Eine Optimierung nach bspw. kleinster Transportzeit oder -kosten erlaubt eine Unterstützung oder gar Automatisierung dieser Aktivität (②).

Sind alle Geräte zugewiesen, wird ein Ablaufprogramm für die einzelnen Steuerungen generiert und übertragen. Hiermit wird innerhalb des Steuerungsverbunds (Montagestation) Echtzeitfähigkeit garantiert, steuerungsübergreifend wird über herstellerneutrale Protokolle kommuniziert (z. B. OPC UA). Im Anschluss wird die Anlage gestartet und beginnt mit dem Montageablauf (③).

Tritt ein Rekonfigurationsfall auf, etwa ein Hardwareaustausch oder eine Prozessanpassung, wird der Ablauf der Ressourcenvernetzung, der Prozessspezifikation und des Matchmakings teilweise oder gänzlich erneut durchlaufen. Durch eine Automatisierung

dieser Bestandteile lassen sich sowohl physische als auch logische Rekonfigurationsaufwände ganzheitlich reduzieren.

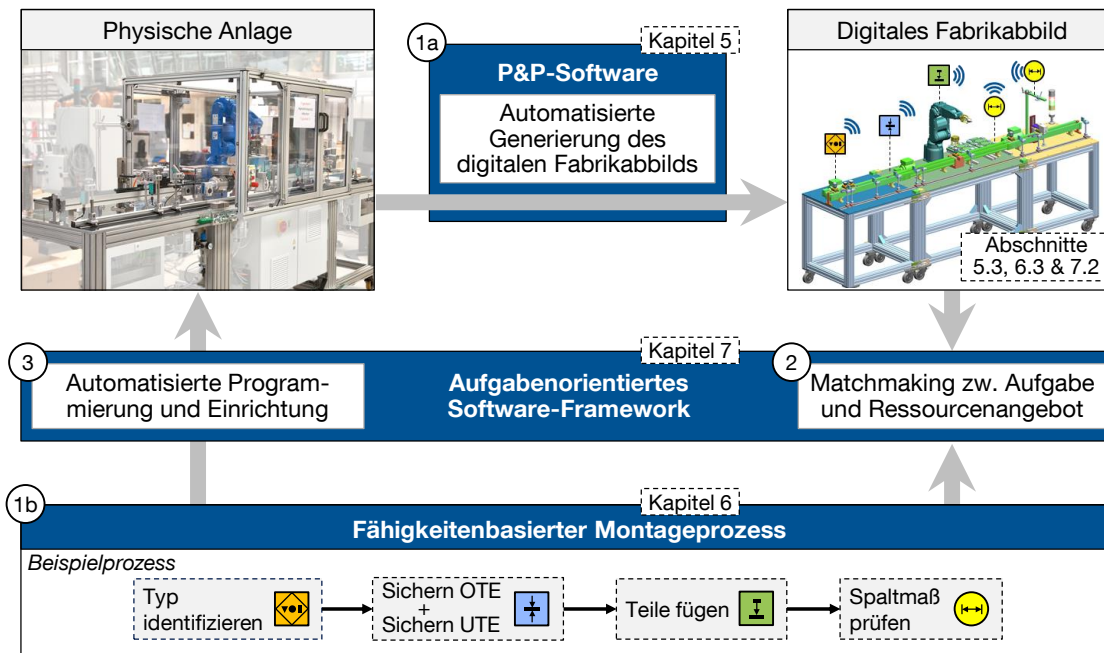


Abbildung 4.2: Ablauf des Gesamtkonzepts aus Anwendersicht

4.4 Methode zur Beschreibung der Softwarekonzepte

Zur Erstellung einer zielgerichteten Softwareapplikation unterteilt die Softwareentwicklung die Phasen *Anforderungsanalyse*, *Konzeption*, *Implementierung* und *Test* (ABTS & MÜLDER 2017, S. 479 f.). Diese Strukturierung wird für die nachfolgenden Konzeptkapitel 5 – 7 aufgenommen. So wird entsprechend eine Anforderungsanalyse durchgeführt, auf welche die Beschreibung des jeweiligen Konzepts folgt. Nach ABTS & MÜLDER (2017, S. 483 ff.) müssen dabei sowohl die Softwarestruktur (*statische Sicht*) als auch die Abläufe des Systems (*dynamische Sicht*) beschrieben werden. Im anschließenden Kapitel 8 werden Informationen zur Implementierung sowie eine Bewertung dieser gegeben.

5 Automatisierte Vernetzung von Feldgeräten

5.1 Anforderungen

5.1.1 Referenzarchitektur

Automatisierungsanlagen besitzen eine hohe Geräte- und Technologieviefalt innerhalb der Kommunikationsebene (Abschnitt 2.2). Um relevante Gerätemerkmale für ein automatisiertes Vernetzungskonzept zu definieren, wird eine Referenzarchitektur hergeleitet, welche einen technologischen Überblick liefert. Als Anlagenstruktur wird der Anwendungsfall aus Abschnitt 4.2 verwendet. Hierfür werden die für eine Umsetzung notwendigen Komponenten aus Katalogen international tätiger Automatisierungskomponenten-Hersteller gesucht und deren Kommunikationstechnologien analysiert. Um die Validität der Architektur zu stützen, erfolgte zusätzlich eine öffentliche Umfrage (HAMMERSTINGL & REINHART 2015a), deren Ergebnisse in die Architekturerstellung einfließen.

Dabei ist es für ein Vernetzungskonzept weniger wichtig, über welche spezifische Technologie eine Geräteart angeschlossen wird, vielmehr ist die generelle Merkmalskombination der vorkommenden Kommunikationstechnologien von Bedeutung. Aus diesem Grund werden aus der Referenzarchitektur generelle Geräteklassen abgeleitet, die jeweils für eine bestimmte kommunikationsseitige Ausprägung stehen. Hieraus lassen sich dann allgemeingültige Anforderungen an das Konzept definieren. Das beschriebene Vorgehen zeigt Abbildung 5.1.

5.1 Anforderungen

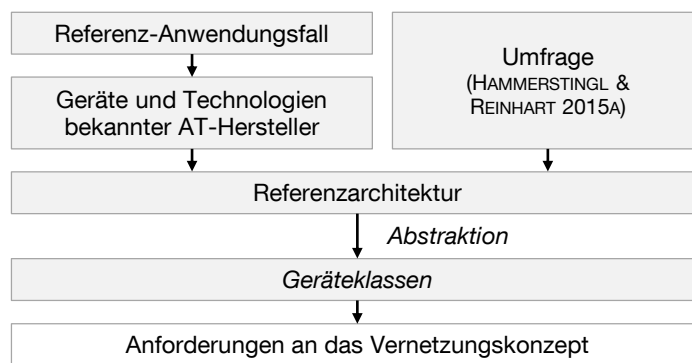


Abbildung 5.1: Methode zur Herleitung der konzeptuellen Anforderungen

Die so entstandene Referenzarchitektur ist in Abbildung 5.2 dargestellt. Generell ist eine hierarchische Kommunikationsstruktur über mehrere Ebenen feststellbar, bedingt durch die Aufteilung zwischen Steuerungskomponenten und Peripheriegeräten sowie dem Umstand, dass einfache Peripheriegeräte aus Kosten- und Performancegründen oftmals nicht die komplexeren Busprotokolle implementieren. Weiterhin ist zu erkennen, dass eine große Anzahl von unterschiedlichen Kommunikationstechnologien parallel existiert (Analog, Fluidisch, IO-Link, Ethernet, Feldbus-/RTE-Varianten, USB, weitere proprietäre Protokolle), mit jeweils unterschiedlichen Merkmalen in Bezug auf Datendurchsatz, Echtzeitfähigkeit etc.

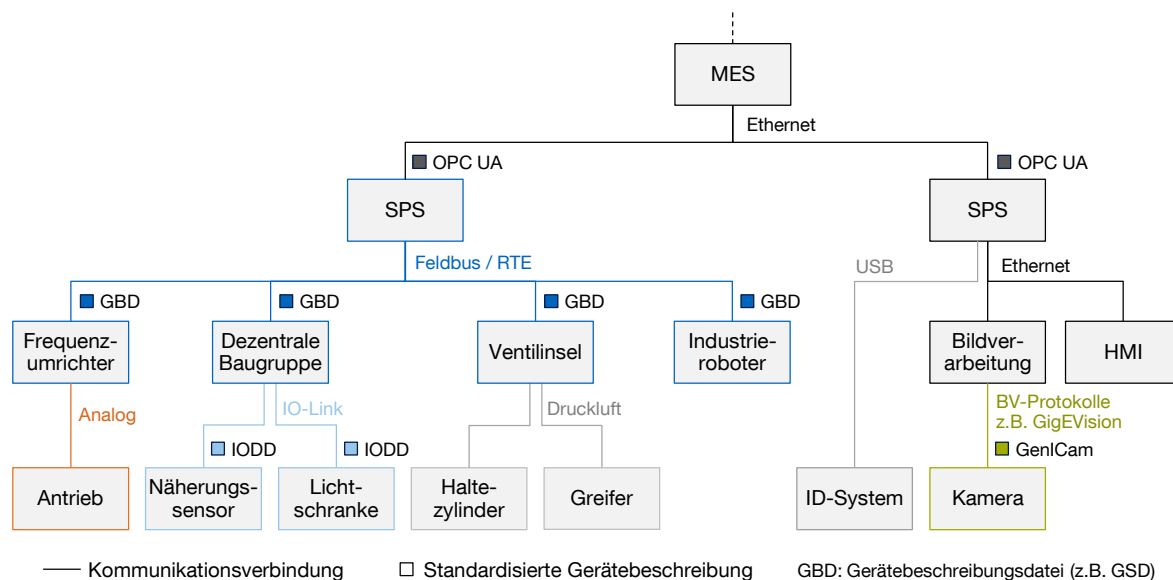


Abbildung 5.2: Referenzarchitektur von Montagesystemen nach HAMMERSTINGL & REINHART (2015b)

Aus der Referenzarchitektur werden insgesamt fünf kommunikationsseitige Geräteklassen abgeleitet:

Primitive Sensoren und Aktoren stellen die erste Klasse dar. Sie sind jeweils direkt mit dem übergeordneten System verdrahtet (z. B. 24 V oder Druckluft), die übertragenen Informationen sind rein physikalischer Natur (binär oder analog). Sie liefern keine digitalen Selbstbeschreibungen, sodass für die Steuerung nicht bekannt ist, welches Gerät angeschlossen ist. Die semantische Interpretation übernimmt der Nutzer während der Programmerstellung. Eine Parametrierung muss direkt am Feldgerät erfolgen, z. B. über Taster oder codierte Drehschalter.

Die zweite Klasse wird als *Basisgeräte* bezeichnet. Auch sie sind direkt verdrahtet (kein Bussystem), wobei allerdings ein Minimalprotokoll existiert, das mindestens die Identifikation angeschlossener Geräte ermöglicht. Hierdurch lassen sich neben binären und numerischen Werten auch komplexe Daten wie etwa Codewörter übertragen. Dies ermöglicht eine externe Parametrierung. Informationen zur Signalcodierung werden über standardisierte Beschreibungsmittel bereitgestellt. Als etabliertes Protokoll in dieser Kategorie gilt „IO-Link“ (SPINNAKE 2016), mit dessen Gerätebeschreibung „IODD“.

In die nächsten Kategorien fallen Geräte, die über einen Bus kommunizieren. Das Ethernet-Protokoll ist hierbei der bekannteste Vertreter, weshalb diese Komponenten als *Standard*-Geräte bezeichnet werden. Aufgrund des hohen Datendurchsatzes und fehlenden Determinismus wird diese Merkmalsklasse hauptsächlich zur Kommunikation mit der IT-Ebene und Anlagensvisualisierung eingesetzt. Im prozessnahen Bereich sind Bildverarbeitungssysteme bekannte Vertreter hierfür. Kameradaten werden hier über domänenspezifische Protokolle verschickt, bspw. das auf Ethernet/UDP aufbauende „GigE-Vision“-Protokoll.

Im Bereich der deterministischen Fabrik-Kommunikation gibt es traditionelle Feldbusse oder Real-time-Ethernet (RTE). Diese existieren in einer Vielzahl von – meist inkompatiblen – Ausprägungen (vgl. Abschnitt 2.2.1) und werden allgemein als *Feldgeräte*-Klasse bezeichnet. Eine Selbstbeschreibung ist auf die Kommunikationsschnittstelle beschränkt, z. B. GSD bei PROFINET, die jedoch manuell in den Kommunikationsmaster importiert werden muss. Eine Parametrierung erfolgt in der Regel über proprietäre Hersteller-Tools oder den FDT-Standard. Hierbei liefern Hersteller einen ausführbaren Treiber, der als Plug-In in eine sog. „Rahmenapplikation“ geladen wird.

Im Rahmen von INDUSTRIE 4.0-Konzepten wird in jüngster Zeit verstärkt an Geräten gearbeitet, die über einen eigenen Server zur Online-Beschreibung ihres Zustands verfügen (z. B. OPC UA). Dies geschieht hauptsächlich bei komplexeren Geräten, wie etwa IRS. Um eine Zukunftsfähigkeit des hier vorgestellten Konzepts sicherzustellen, werden diese Geräte als zusätzliche Klasse aufgenommen und in Anlehnung an GORECKY ET AL. (2017, S. 561) als „Smarte Feldgeräte“ (SFG) bezeichnet¹. Nachdem die IT-seitige Vernetzung auf Ethernet basiert, wird angenommen, dass bei diesen Geräten ein zusätzlicher Echtzeit-Kommunikationskanal existiert (vgl. bspw. SPSen).

Tabelle 5.1 zeigt die genannten Geräteklassen in der Übersicht.

Tabelle 5.1: Kommunikationsseitige Geräteklassen in der Montage

Geräteklasse	Kommunikationstopologie	Kommunikationsdeterminismus	Selbst-Identifikation	Digitale Selbst-Beschreibung	Übertragene Daten	Verbindung	Beispiel
<i>Primitiv</i>	Direktverdrahtung	Ja	Nein	–	b, n	Analog, Druckluft	Greifer
<i>Basis</i>		Ja	Ja	Kommunikation (Datei)	b, n, w	IO-Link	Abstandssensor
<i>Standard</i>		Nein	Nein	–	b, n, w	Ethernet, USB	HMI
<i>Feld</i>	Bus	Ja	Ja	Kommunikation (Datei)	b, n, w	Feldbus, RTE	Schweißgerät
<i>Smart</i>		Nein & Ja	Ja	Merkmale (Online-Server)	b, n, w, k	RTE	SPS, Roboter

b: bool, n: numerisch, w: Wort, k: komplex

5.1.2 Anforderungen an das Konzept

Betrachtet man die Referenzarchitektur und hergeleiteten Geräteklassen aus Abschnitt 5.1.1, so lassen sich verschiedene Anforderungen an das Konzept ableiten:

Als wichtige Anforderung gilt in der Automatisierungstechnik die Echtzeitfähigkeit der Komponentenansteuerung, was die Verwendung von echtzeitfähigen Industriesteuerungen (SPS) und -protokollen erfordert (Anforderung Vernetzung AV 1). Nachdem die Einbindung neuer Feldgeräte bisher manuell über die Engineeringtools der Hersteller erfolgt, müssen die in Abschnitt 2.2.1 beschriebenen Konfigurationsschritte

¹Eine Bezeichnung als (primitives) CPS wird als unpassend angesehen, da SFGs nur eine mangelnde semantische Standardisierung aufweisen, keinen synchronen Digital Twin besitzen und keine eigenständige Entscheidungsfindung für die jeweilige Aufgabe aufweisen.

automatisiert werden (AV 2). Für eine Konfiguration des Kommunikationsmasters wird entsprechend eine informationstechnische Schnittstelle zum Zugriff benötigt (AV 3). Da die AT-Landschaft jedoch durch eine Vielzahl proprietärer Lösungen und Schnittstellen geprägt ist, muss das Konzept diese Heterogenität unterstützen (AV 4). Der Konfigurationsansatz muss dabei mit verschiedenen, hierarchischen Netzwerken umgehen können (AV 5) und die Einbindung primitiver, sich nicht selbstständig identifizierender Komponenten beherrschen (AV 6). Das Konzept soll dabei bestehende Beschreibungsstandards nutzen, um eine Kompatibilität zu bisherigen Softwaretools und Komponenten sicherzustellen (AV 7).

Neben dem Ergebnis einer automatisiert eingerichteten Echtzeitkommunikation ist für eine aufgabenorientierte Auswahl und Programmierung (Kapitel 7) die Bereitstellung eines Digital Twins notwendig. Da nur smarte Feldgeräte diesen über ihren Online-Server selbst publizieren könnten, wird für alle anderen Geräteklassen eine Art Stellvertreter-Server notwendig (Proxy), der die Digital Twins stattdessen veröffentlicht (AV 8) und mit den realen Geräten synchronisiert (AV 9).

Eine Übersicht aller Anforderungen kann Anhang A.1 entnommen werden.

5.2 Konzept zur automatisierten Vernetzung von Feldgeräten

5.2.1 Gesamtarchitektur

Smarte Feldgeräte können ihre Fähigkeiten direkt über einen Server als hardwareunabhängige Dienste im Sinne einer SOA anbieten. Deren interne Logik wandelt hierbei die hardware-spezifischen Größen in standardisierte Informationen des auf dem Server laufenden Digital Twin um.

Die überwiegende Mehrheit der Feldgeräte in heutigen Fabriken verfügt jedoch nicht über diese technologische Fähigkeit, weshalb eine externe Software diese Aufgabe erfüllen muss (nachfolgend *Plug&Produce-Software* genannt). Bedingt durch das Vorherrschen proprietärer Systemlandschaften, muss diese Verknüpfung zwischen Hardware- und Digital-Twin-Informationen für jedes Feldgerätemodell spezifisch erfolgen. Hieraus folgt der Ansatz, die Konvertierungslogik in Gerätetreiber zu kapseln, um so die Modularität der Plug&Produce-Software sicherzustellen.

Die Plug&Produce-Software hat dabei drei Hauptaufgaben, welche sequentiell wiederholend abgearbeitet werden (vgl. Abbildung 5.3):

- ① Identifikation angeschlossener Geräte
- ② Laden des korrespondierenden Gerätetreibers und Aufbau der Kommunikation
- ③ Publikation des Digital Twins² und Synchronisation über den Treiber mit dem realen Gerät

Die Summe der publizierten Geräteserver stellt das aktuelle, digitale Fabrikabbild dar (Digital Factory Twin, ④). In Kombination mit einem fähigkeitsbasierten Prozessablauf (⑤) kann das in Kapitel 7 zu definierende Software-Framework geeignete Ressourcen bestimmen und einen Anlagenablauf bestehend aus Dienstaufrufen zu diesen Ressourcen generieren (⑥).

Die Transformation des erstellten Prozessablaufs in ausführbaren Steuerungscode übernimmt ein Codegenerator (⑦). Dieser exportiert den Prozess als SFC im „PLCopen XML“-Standard, welcher automatisiert in das jeweilige Engineering-Tool der Steuerung integriert wird (⑧). Das Tool übernimmt die Kompilierung und den Upload für die entsprechende Hardwareplattform. Anschließend erfolgt der Start der Steuerung, die den importierten Prozessablauf ausführt und mit den Feldgeräten in Echtzeit kommuniziert (⑨). Die Synchronisation mit den Digital-Twin-Servern der Geräte erfolgt dabei über die Schnittstelle der Steuerung – der eigentliche Programmablauf bleibt hiervon unberührt, wodurch die Zuverlässigkeit und Echtzeitfähigkeit des Systems nicht eingeschränkt wird.

5.2.2 Architektur der Plug&Produce-Software

Angeschlossene Ressourcen müssen dem aufgabenorientierten System bekannt sein, um hierin genutzt werden zu können. Smarte Feldgeräte können sich hierbei selbstständig sichtbar machen, das OPC UA-Protokoll besitzt hierfür beispielsweise Discovery-Mechanismen („Local Discovery Server“, IEC 62541).

²In Kapitel 7 wird gezeigt, dass der Digital Twin auch ohne Verknüpfung zu realer Hardware geladen werden kann, insofern stellt dessen hardwareseitige Publikation nur eine Möglichkeit dar.

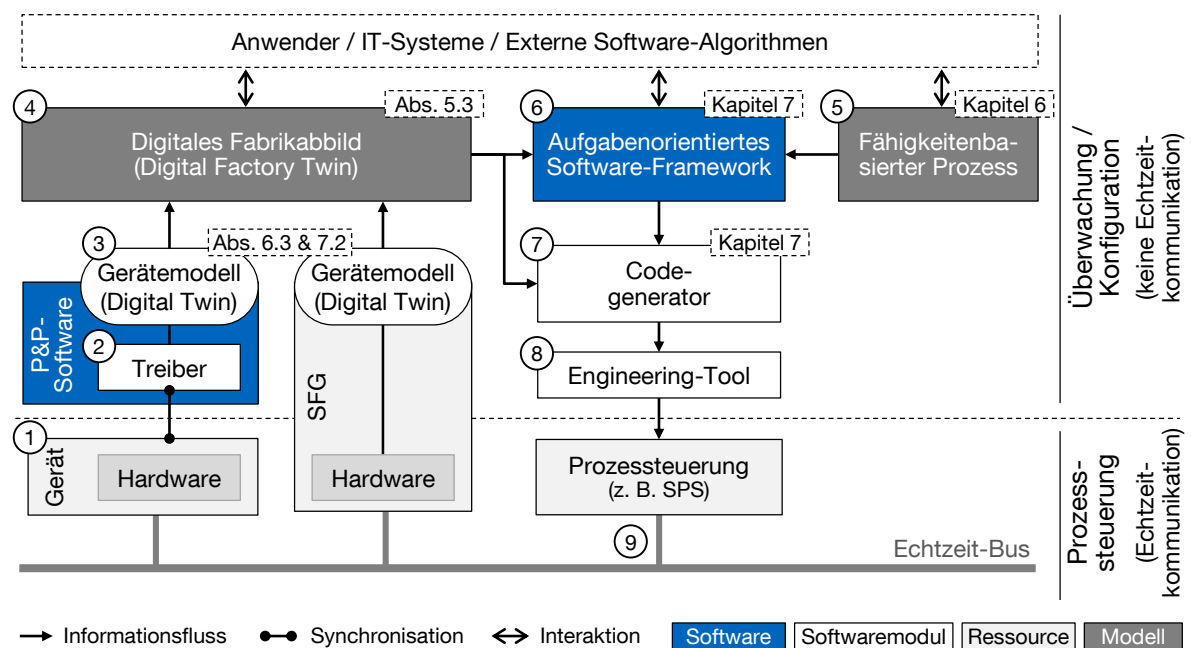


Abbildung 5.3: Schematische Darstellung der Gesamtarchitektur und des Informationsflusses im Vernetzungskonzept

Für alle anderen Feldgeräte übernimmt dies die Plug&Produce-Software. Die Gerätetreiber stellen hierbei eine einheitliche Schnittstelle zur Verfügung, die es erlaubt, technologieunabhängig generische Anfragen an das Netzwerk und deren Komponenten zu stellen (→AV4). Ausgehend von dem umgebenden Host-System, fragt die Software verfügbare Kommunikationskanäle des Systems ab. Pro Kanal wird ein Kommunikationskanaltreiber (KKT) geladen, welcher die anschließende allgemeine Suchanfrage nach Netzwerkteilnehmern in das entsprechende Protokoll übersetzt.

Mit jedem erkannten Gerät wird eine Grundkommunikation aufgebaut, über die eine eindeutige Kennung angefordert wird (bspw. Modell- und Seriennummer, MAC-Adresse). Bei Feldbussen, wie PROFIBUS, ist dies bspw. die Ident-Nummer (REISSENWEBER 2009, S. 172). Ethernet und RTE-Protokolle (z. B. PROFINET) bieten hierzu das „Address Resolution Protocol (ARP)“ oder das „Link Layer Discovery Protocol (LLDP)“ an. Ist dies nicht der Fall, wie etwa bei Druckluft-Verbindungen, öffnet der zugehörige KKT bei Aufruf eine Eingabemaske, in der Anwender die Geräte-ID manuell eingeben oder vom Gerät abscreiben können. Auf diesen Ablauf zur Konfiguration primitiver Ressourcen wird in Abschnitt 5.2.3 genauer eingegangen.

Der beschriebene Ablauf ist für jedes Kommunikationsnetz durchzuführen und setzt sowohl einen physischen Zugriff auf das Netzwerk als auch die Implementierung ei-

nes jeden verwendeten Protokolls in der Plug&Produce-Software voraus. Für weit verbreitete Protokolle, wie etwa Ethernet, ist das ein gangbarer Weg, für die Vielzahl an vorherrschenden Automatisierungsprotokollen jedoch sehr aufwandsintensiv. Aus diesem Grund wird ein hybrider Ansatz vorgeschlagen, der neben der Implementierung bekannter Protokolle auch die verbauten Steuerungen nutzt, um die Suchaufgabe stattdessen durchzuführen. Dies ist möglich, da Steuerungen für die Abarbeitung des Prozessablaufs sowohl physisch mit den Komponenten verbunden als auch das jeweilige Kommunikationsprotokoll implementiert haben müssen. SPS-Hersteller bieten hierzu im Rahmen des Engineeringprozesses Suchmechanismen an, welche über das Engineeringtool oder eine API zur SPS-Hardware aufgerufen werden können³. Da der Steuerungszugriff jedoch meist proprietär gelöst ist, wird die Ansteuerung über einen *Hosttreiber* abstrahiert (→AV3).

Durch ihren modularen Hardwareaufbau (vgl. Abschnitt 2.2.2) können Steuerungen unterschiedliche Schnittstellen und Protokolle anbieten. Um dies auch softwareseitig widerzuspiegeln, wird kein monolithischer Hosttreiber gestaltet, sondern jedes Kommunikationsprotokoll in einem KKT gekapselt. KKTs benutzen in diesem Falle Schnittstellen des Hosttreibers, um hierüber Zugang zum Netzwerk zu erhalten (HKKTs). Sie bilden zusammen mit dem Hosttreiber ein Bündel für den Hardwarezugriff auf die Steuerung.

Generell kann der KKT eigenständig oder als Bestandteil der Hosttreiber-Suche implementiert werden (HKKT). Welcher Weg zu bevorzugen ist, muss über den Aufwand zur Implementierung des Kommunikationsprotokolls bzw. für den Zugriff auf die Steuerungshardware abgewogen werden. Für Industrie-PCs, die über einheitliche Betriebssysteme (Windows, Linux) und Schnittstellen (Ethernet, USB) verfügen, erscheint die erste Variante sinnvoller (Abbildung 5.4, Feldebene rechts), für traditionelle Steuerungssysteme (SPS) die zweite Variante (Abbildung 5.4, Feldebene links).

Durch die zurückgelieferte Geräte-Identifikation kann die Plug&Produce-Software aus einem Repository den zugehörigen *Gerätetreiber* laden. In diesem ist eine Beschreibung für die Signalcodierung der Gerätekommunikation enthalten. Hierfür wird auf die existierenden Beschreibungsmittel der jeweiligen Protokolle gesetzt (z. B. GSD, IODD), um eine Kompatibilität zu bisherigen Lösungen sicherzustellen (→AV7). Zusammen mit den Daten aus der Geräteerkennung kann die Plug&Produce-Software diese Informationen

³Beispiele sind das ADS-Protokoll und *TwinCat*-Schnittstelle des Herstellers Beckhoff, das *OpenCore-Interface* bei Bosch oder der *System Diagnose Manager* bei B&R.

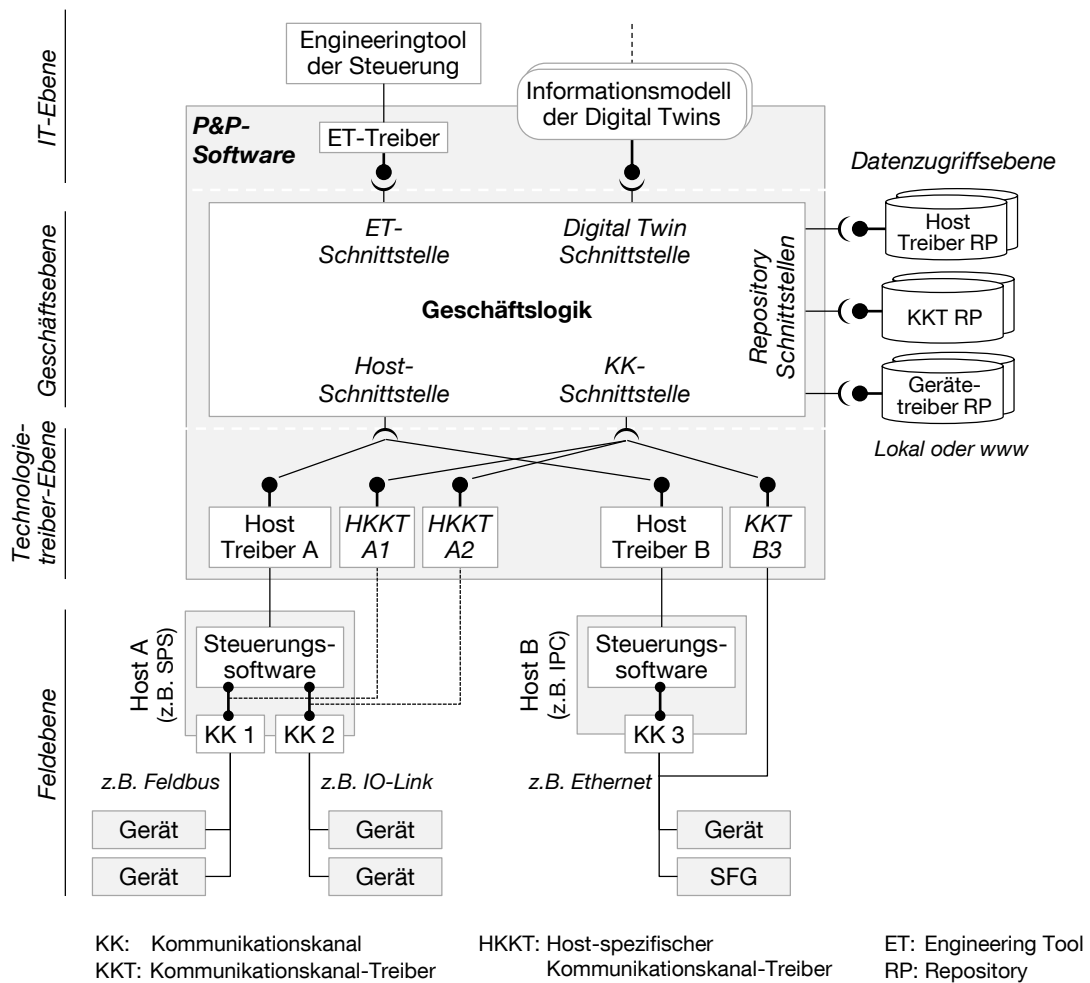


Abbildung 5.4: Struktur der Plug&Produce-Software. Der Ansatz von Geräte-, Host- und Kommunikationskanaltreibern ermöglicht den Umgang mit heterogenen Technologien.

in den Kommunikationsmaster (z. B. SPS) importieren, um das IO-Mapping anzupassen (→AV2) und beide Teilnehmer deterministisch kommunizieren zu lassen (→AV1). Der Kommunikationsmaster kann dabei mit einem Hostgerät zusammenfallen, in diesem Fall wird die Ressource über den entsprechenden Host-Treiber konfiguriert.

Neben Kommunikationsinformationen enthält der Gerätetreiber auch eine umfassende digitale Beschreibung der Ressource (Digital Twin), welche alle für eine aufgabenorientierte Planung und Programmierung notwendigen Informationen bereitstellt. Die Plug&Produce-Software lädt den Digital Twin und publiziert ihn stellvertretend für das Gerät (→AV8).

Nach dem Import des Gerätetreibers fragt die Plug&Produce-Software über dessen API bei der Ressource an, ob sie eventuelle Subnetze bereitstellt. Bei monolithischen

5.2 Konzept zur automatisierten Vernetzung von Feldgeräten

Geräten kann die Antwort direkt im Treiber erfolgen, für modulare Geräte muss ein tatsächlicher Netzwerkaufruf geschehen. Im positiven Fall wird für jedes Subnetz der entsprechende KKT geladen und der Ablauf beginnt erneut (→AV5). Die Rekursion wird dabei solange ausgeführt, bis keine neuen Geräte mehr gefunden werden. Der beschriebene Ablauf für HKKTs wird in Sequenzdiagramm Abbildung 5.5 gezeigt. Die alternative Variante für Host-unabhängige KKTs ist in Anhang A.2 zu finden.

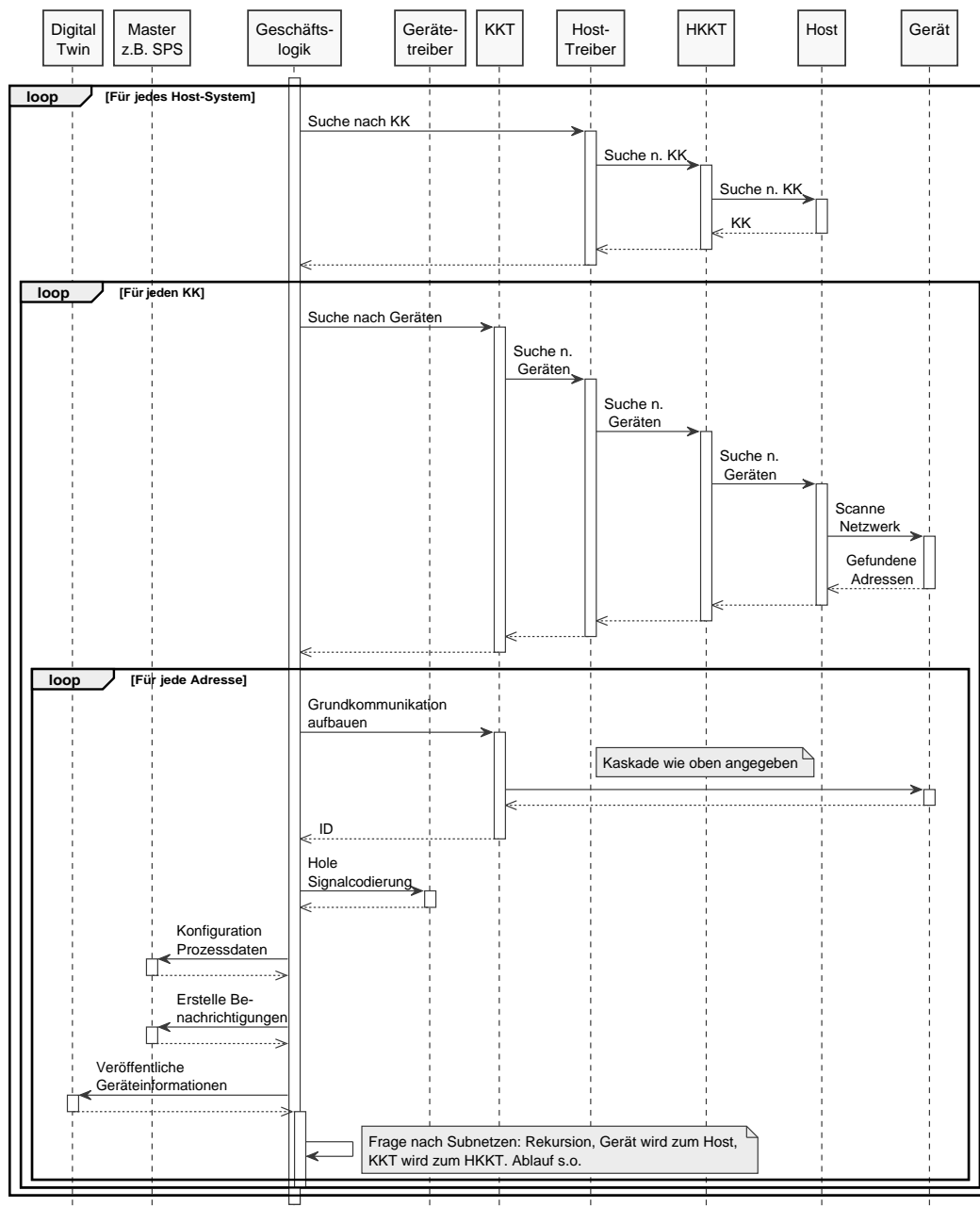


Abbildung 5.5: Ablauf der Vernetzung, Variante mit Host-KKTs, wie etwa bei einer SPS-Suche. Der Kommunikationsmaster ist als separates Element dargestellt, um die Rekursion verständlicher darzustellen – in Realität erfolgt die Integration als Host-Gerät.

5.2.3 Ansteuerung von Geräten in Kommunikationshierarchien

Wie der Referenzarchitektur (Abbildung 5.2) zu entnehmen ist, treten oftmals mehrstufige Gerätehierarchien auf, bspw. wird ein pneumatischer Aktor mit einer Ventilinsel verbunden, die wiederum mit einer SPS kommuniziert. Dabei setzt jede Ressource Eingangssignale in Ausgangssignale um. Dies geht meist mit einer Informationswandlung einher (oftmals physikalischer Natur), etwa von informationstechnischen Signalen auf mechanische oder elektronische Größen. Ressourcenklassen, die keine prozessnahen Funktionen ausführen, sondern hauptsächlich zur Informationswandlung eingesetzt werden, sind nachfolgend als *Gateways* bezeichnet.

Die Informationswandlung kann dabei als Funktion gesehen werden, der Zusammenschluss von Geräten als mathematische Verkettung der Funktionen. Mit dem in Abbildung 5.6 gezeigten Beispiel aus SPS → Gateway → Gerät ergibt sich folgende Informationsstrecke:

$$d = f_{\text{gerät}}(c) \circ f_{\text{gateway}}(b) \circ f_{\text{sps}}(a) \quad (5.1)$$

mit f_x als Übertragungsfunktion $f_x : x \rightarrow y$ und a bis d als die jeweiligen Eingangs- bzw. Ausgangssignale der Ressourcen bzw. Übertragungsfunktionen.

Soll ein definiertes Verhalten d einer Ressource hierbeigeführt werden, ist ein entsprechend gestaltetes Eingangssignal a in die Informationsstrecke einzubringen. Hierzu ist die Inverse der Strecke zu bilden:

$$a = \underbrace{f_{\text{sps}}^{-1}(b)}_{\text{SPS-Konfiguration}} \circ \underbrace{f_{\text{gateway}}^{-1}(c)}_{\text{Gateway-treiber}} \circ \underbrace{f_{\text{gerät}}^{-1}(d)}_{\text{Geräte-treiber}} \quad (5.2)$$

Die Inverse des Gateways und der Ressource wird durch die jeweiligen Gerätetreiber bereitgestellt, welche mit dem in Abschnitt 5.2.2 beschriebenen Ablauf sukzessive in die Steuerung integriert und entlang der Verkettungsreihenfolge verknüpft werden. Die Inverse der Steuerung ist durch die zuvor erfolgte Konfiguration der Kommunikation erfolgt. Über die Definition der Gerätefähigkeit (Kapitel 6) erfolgt die Abbildung des realen Geräteverhaltens d auf den entsprechenden Eingangsbefehl für die Steuerung a : $f_{\text{fähigkeit}} : d \rightarrow a$

Somit kann die Ansteuerung von Geräten in Kommunikationshierarchien als eine Aneinanderkettung der einzelnen Treiberbausteine in der Steuerung ausgeführt werden (→AV5). Bezogen auf das Beispiel aus Abbildung 5.6a, soll ein Befehl zum Öffnen des

Greifers d gesendet werden. Der Gerätetreiber übersetzt den Aufruf in den am Greifer zugehörigen Ventileingang c , welcher als Eingangsgröße für den Gatewaytreiber dient und in den zugehörigen Protokollaufwurf b gewandelt wird, den die SPS-Konfiguration schließlich physisch ausführt (vgl. Abbildung 5.6b).

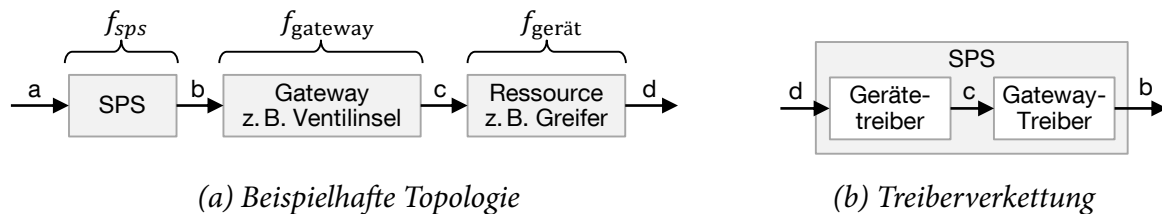


Abbildung 5.6: Informationsverkettung von Kommunikationshierarchien

Kann im Suchalgorithmus keine automatisierte Identifikation erfolgen (etwa bei primitiven Ressourcen bzw. Kommunikationskanälen), so erscheint bei Aufruf des Suchbefehls durch den entsprechenden KKT eine Eingabemaske, bei welcher der Anwender den Treiber für das unterlagerte Gerät (Kind-Gerät) manuell auswählen kann (\rightarrow AV6). Hierzu wird die Seriennummer benötigt, die bspw. durch einen optischen Code am Gerät ausgelesen wird. Nach erfolgter Auswahl läuft der Suchalgorithmus identisch zur automatisierten Konfiguration weiter.

Existieren mehrere Anschlussmöglichkeiten am übergeordneten Gerät (Elter) oder müssen mehrere Einzelverbindungen zwischen den Teilnehmern angesprochen werden, so muss der Anwender die physisch erfolgte Verbindung $f_{\text{kommunikation}}$ digital definieren. Hierzu werden alle Ausgangsanschlüsse (Out-Ports) des Hostgeräts und alle Eingangsschnittstellen der unterlagerten Ressource (In-Ports) über die Gerätetreiber geladen und grafisch gegenübergestellt. Die Schnittstellen sind typbehaftet, der Anwender kann also nur kompatible Verbindungen analog den realen Gegebenheiten definieren (vgl. Abbildung 5.7).

Das Ergebnis der Verbindung wird im Treiber des Elter-Geräts gespeichert und bei der Integration dessen in die Steuerung übernommen. Im obigen Beispiel entspräche dies der tatsächlich genutzten Druckluftausgänge an der Ventilinsel für den Öffnen- und Schließen-Kanal des Greifers.

5.2.4 Synchronisation des Digital Twins

Nach der initialen Publikation muss der Digital Twin fortlaufend mit Werten des realen Geräts synchronisiert werden, um ein aktuelles Abbild dessen zu liefern (\rightarrow AV9). Dies

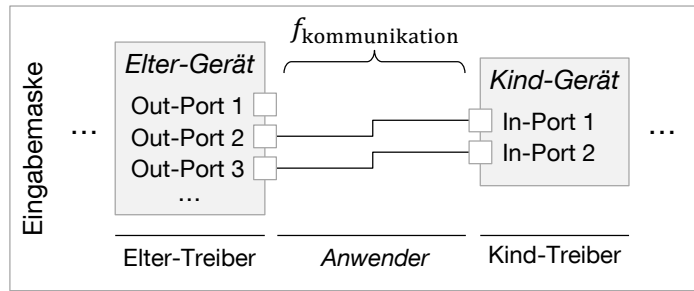


Abbildung 5.7: Dynamische Kommunikationsverkettung zwischen zwei Ressourcen

kann über den Hosttreiber geschehen, welcher auf der Steuerung einen Benachrichtigungskanal zur Plug&Produce-Software erzeugt. Falls Wertänderungen auftreten, erkennt dies die SPS-Konfiguration und sendet über den Aktualisierungsmechanismus des Hosttreibers eine Nachricht an die Plug&Produce-Software, die wiederum die Änderungen in den Digital Twin spiegelt (Abbildung 5.8a). Soll auf die proprietäre Schnittstelle verzichtet werden, kann auf der Steuerung auch ein bekannter Server-Standard (z. B. OPC UA) verwendet werden. Dieser wird so konfiguriert, dass Geräteaktualisierungen an den zugehörigen Client in der Plug&Produce-Software geschickt werden und umgekehrt (Abbildung 5.8b). Welcher Ansatz umzusetzen ist, hängt von den technologischen Gegebenheiten ab: Ersterer benötigt keinen lauffähigen Steuerungsserver und kann somit bereits zur Konfigurationszeit der Steuerung eingesetzt werden, Letzterer bietet eine herstellerunabhängige Kommunikation, hierfür müssen jedoch die Digital-Twin-Daten zuvor in den Host-Server integriert und verlinkt werden.

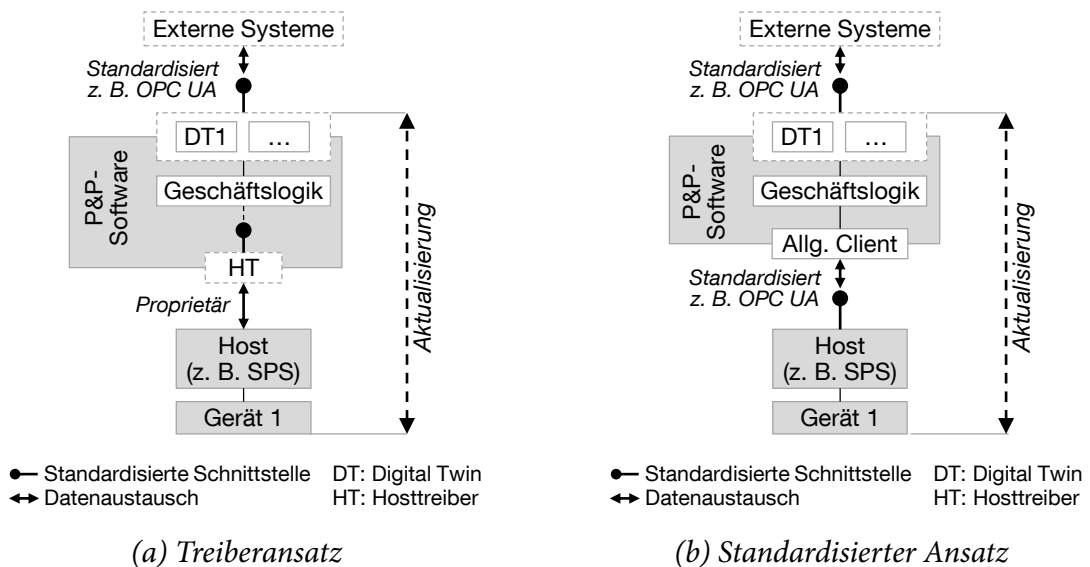


Abbildung 5.8: Mechanismen zur Aktualisierung des Digital Twins auf dem Stellvertreter-Rechner

Über den Digital Twin können externe Systeme die im Gerätetreiber hinterlegten Funktionalitäten ansprechen, die über den Aktualisierungsmechanismus wiederum an das Gerät weitergeleitet werden. Der Synchronisationsmechanismus erlaubt prinzipiell eine Ansteuerung der Geräte über den Digital Twin. Für Konfigurationsaufgaben ist dies möglich, im Produktivbetrieb aufgrund des fehlenden Determinismus in Protokoll und Plug&Produce-Software weniger geeignet. Hier wird ein Ansatz verfolgt, bei dem Steuerungen konventionell mit den Ressourcen kommunizieren und steuerungsübergreifend ein standardisierter Datenaustausch erfolgt, bspw. über OPC UA. Der Codegenerator (Abschnitt 7.3.6) zerlegt aufgrund der vorherrschenden Verbindungstopologie (Abschnitt 5.3) den Prozessablauf für die einzelnen Steuerungsverbünde und integriert die Subprozesse in die Steuerungen. Über den Gerätetreiber wird für jedes Gerät ein Steuerungsbaustein für den entsprechenden Fähigkeitsaufruf importiert. Hierdurch ergibt sich eine echtzeitfähige Ressourcenansteuerung (AV1), welche unabhängig von Technologie und Latenz des Aktualisierungsmechanismus ist.

5.2.5 Bestandteile des Ressourcentreibers

Geräte- und Hosttreiber werden allgemein als *Ressourcentreiber* bezeichnet. Sie erfüllen zwei Aufgaben: Bereitstellung von Schnittstellen zur Konfiguration der Ressource sowie deren Selbstbeschreibung. Dabei wird zwischen klassen- und instanzenspezifischem Wissen unterschieden: *Klassenspezifisches Wissen* ist über alle Instanzen gemeinsam gültig, beispielsweise die Konfigurationsschnittstelle eines Gerätemodells. *Instanzenspezifisches Wissen* unterscheidet sich für jede Geräteentität, etwa die aktuelle Zugriffsadresse oder die Anzahl der Betriebsstunden. Der Treiber ist deshalb zweigeteilt: Befehle und die Merkmalsstruktur sind im klassenspezifischen Treiberkern gespeichert. Zu diesem wird auf Basis der Geräteidentifikation eine zugehörige Wertesammlung geladen, welche die Merkmale mit Inhalt befüllt.

In Abbildung 5.9 ist die zugehörige Treiberstruktur abgebildet, sowie die in den vorherigen Abschnitten beschriebenen Treiberbestandteile.

5.2.6 Besonderheiten in den Domänen Robotik und Bildverarbeitung

Das Konzept ist nicht auf den Bereich von SPSen beschränkt, sondern kann analog in die Steuerungsdomänen der Robotik und Bildverarbeitung übertragen werden. Im Bereich

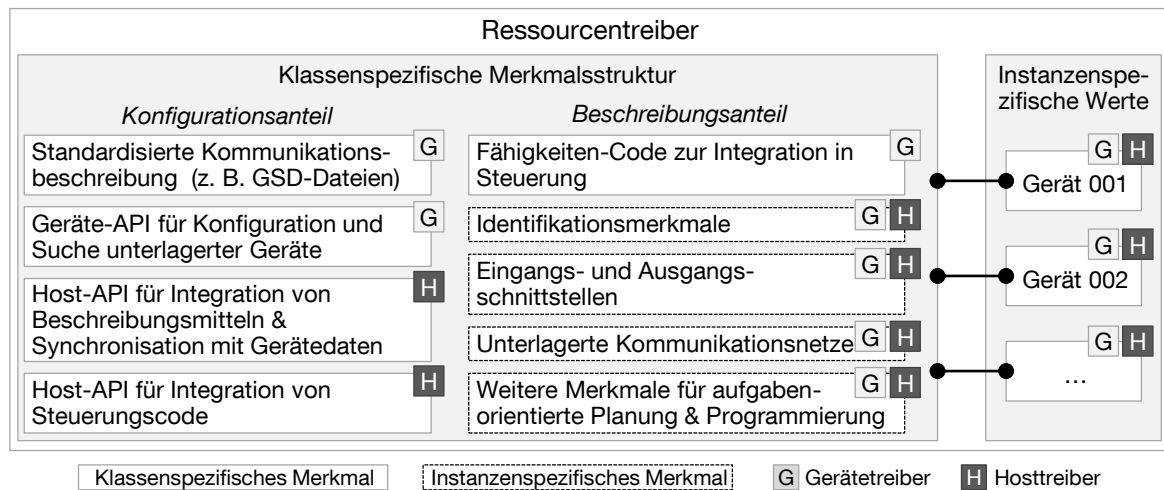


Abbildung 5.9: Der Ressourcentreiber enthält Schnittstellen für die gerätespezifische Konfiguration sowie beschreibende Merkmale, die über einen Instanztreiber geladen werden.

der Robotik bietet sich die Nutzung des Frameworks *Robot Operating System* (ROS) an, hier werden bereits Treiber für den Zugriff auf gängige Robotermodelle bereitgestellt (ROS 2017). Analog zu dem beschriebenen Konzept sucht eine Plug&Produce-Software innerhalb des ROS-Frameworks nach Robotersteuerungen, für gefundene Ressourcen wird der entsprechende Gerätetreiber (ROS-Knoten) in das Framework geladen. Ähnlich zu den standardisierten Beschreibungsdateien im Bereich der Echtzeitkommunikation stellt dieser Treiber jedoch keine weiterführenden Merkmale und Fähigkeiten bereit, welche für eine aufgabenorientierten Planung und Programmierung notwendig sind. Hierzu gehören z. B. Arbeitsräume, Traglasten oder Anschlussmöglichkeiten. Auch hier wird deshalb eine zusätzliche Gerätebeschreibung geladen, die fehlende Geräteinformationen ergänzt und über einen Server im Netzwerk publiziert. Weitere Informationen können HAMMERSTINGL ET AL. (2016) entnommen werden.

Bei BVS wird von einem Industrie-PC ausgegangen, welcher die Bildakquise bei einem Kamerasystem anstößt und eine Auswertung durchführt. Hier läuft die Plug&Produce-Software direkt auf der Steuerungseinheit (IPC) und nicht auf einem dezidierten Rechner, wie im SPS-Fall. Der vorgestellte Suchablauf bleibt hierbei identisch. Auch eine Suchunterstützung ist in dieser Domäne möglich, so bietet bspw. die Bildverarbeitungssoftware „Halcon“ des Unternehmens MVTec eine API, um automatisiert Kamerasysteme im Netzwerk zu identifizieren (MVTEC GMBH 2018, S. 14 ff.). Auch hier reicht eine Gerätebeschreibung des verfügbaren GenICam-Standards nicht aus, weshalb mit dem zuvor beschriebenen Ablauf ein dezidiertes Gerätetreiber geladen und veröffentlicht werden muss.

5.3 Hierarchie des Ressourcen-Informationsmodells

Durch das beschriebene Suchkonzept stellt jede Recheneinheit automatisiert die Digital Twins der unterlagerten Geräte als Proxy im Netzwerk bereit. Analog zu den real existierenden Netzwerktopologien wird eine Gerätestruktur aufgebaut (vgl. Abbildung 5.10). Daneben existieren smarte Feldgeräte, die über ihren inhärenten Server direkt Informationen publizieren. Dies gilt auch für Arbeitskräfte (Werker), die mit Smart Devices (z. B. Tablets) ihre Merkmale bereitstellen und somit durch IT-Systeme eingebucht werden können.

Dabei ergibt die *Summe aller Ressourcenserver* das virtuelle Gesamtabbild der Fabrik. Die Heterogenität der unterschiedlichen Beschreibungsstandards und Übertragungsprotokolle wird hierbei über die einheitliche Digital-Twin-Struktur und das Kommunikationsprotokoll abstrahiert – übergeordnete Planungs- und Programmiersysteme können somit aufwandsarm Informationen einer großen Menge an Ressourcen akquirieren.

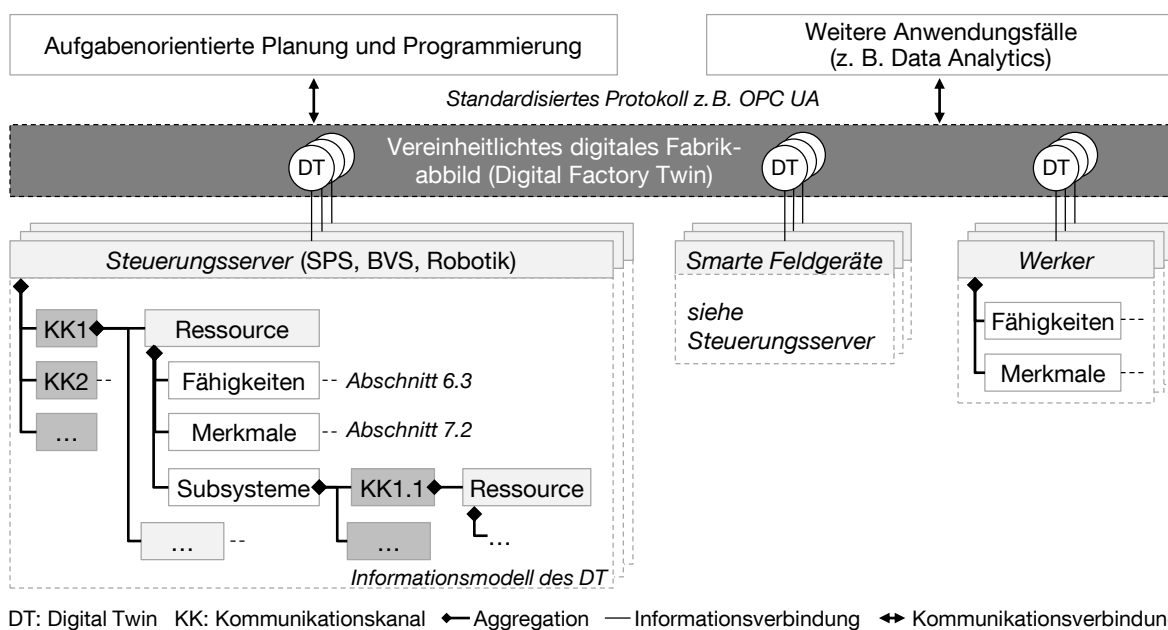


Abbildung 5.10: Das virtuelle Fabrikabbild ist die Sammlung der aktuellen Ressourcen-Server. Durch den vereinheitlichten Kommunikationsweg und Informationsstruktur des Digital Twins können externe Systeme einfach Ressourceninformationen abrufen.

6 Fähigkeiten zur Prozessmodellierung

6.1 Anforderungen

Im vorliegenden Konzept sollen Ressourcen automatisiert mit einem Montageprozess abgeglichen werden. Als generelle Anforderung wird ein Fähigkeitenkatalog benötigt (Anforderung Fähigkeiten AF 1), welcher im Sinne des PPRS-Gedankens als Bindeglied zwischen den Domänen dient, d. h., es muss eine Modellierung von Montageprozessen mit Fähigkeiten möglich sein, welche auch durch Ressourcen angeboten werden können. Analog zu der von SCHMIDT (1992) und BACKHAUS (2016) aufgestellten Forderung zur Definition überschneidungsfreier Prozesse, gilt es hierbei klarer als bisherige Quellen zwischen einfachen und kombinierten Fähigkeiten bzw. Ressourcen zu trennen (AF2).

Um die Güte der Ressourcenauswahl zu erhöhen und eine Programmierung der Geräte zu ermöglichen, sollen hierbei Fähigkeiten durch Parameter spezifiziert werden können (z. B. Messunsicherheit, Bahngeschwindigkeit). Neben der Definition der Parameterstruktur (AF3) ist eine Methode zur Definition neuer Fähigkeitenparameter erforderlich (AF4), außerdem sind übergreifende Standardparameterklassen zu identifizieren (AF5).

Werden in einer ersten Voruntersuchung die erforderlichen Montageprozesse des Referenz-Anwendungsfalls (Abschnitt 4.2) ermittelt und mit den beschriebenen Ressourcen abgeglichen (Tabelle 6.1), so ergeben sich mehrere Erkenntnisse: Die meisten Prozesse im exemplarischen Montagesystem sind Kontrollaufgaben (41 %), gefolgt von Handhabungs- und Transportprozessen (24 %). Entsprechend sind beide Aspekte stärker in den Fokus als bei bisherigen Ansätzen zu rücken (AF6).

Darüber hinaus existieren Fähigkeitenkategorien, welche nur indirekt dem Montageprozess zuzuordnen sind: So befinden sich in Montageanlagen zahlreiche Anzeige- und Bedienelemente, mit denen eine Mensch-Maschine-Interaktion (MMI) realisiert

6.1 Anforderungen

wird, etwa Signallampen oder Quittierungstaster. Hinzu kommt eine generelle Notwendigkeit des Datenaustauschs mit externen IT-Systemen, etwa zur Archivierung von Messergebnissen. Sollen im Rahmen eines ganzheitlichen Rekonfigurationskonzepts auch Ressourcen aus diesen Bereichen einbezogen werden, so müssen diese indirekten Fähigkeiten definiert werden (AF7).

Tabelle 6.1: Domain-Mapping-Matrix zwischen Montageprozessen und Ressourcen im Referenz-Anwendungsfall

		Prozessesequenz im Referenz-Anwendungsfall →															
		Anwesenheitsprüfung WST	Entnahme Produkt UTE	Identifikation Produktvariante	Fügen [UTE→WST]	Puffern WST	Transport in nächstes Modul	Kontrolle Anw. [OTE an Lagerplatz X]	Messen [Pose OTE]	Anwesenheitsprüfung WST	Fügen [OTE→UTE]	Transport in nächstes Modul	Anwesenheitsprüfung WST	Bestromung des Produkts	Messen Produktmerkmale	Ausgabe fehlerhafter Produkte	Speichern d. Qualitätsinformationen
Anlage im Referenz-Anwendungsfall	Werker	-	x	-	x	-	x	x	-	-	x	x	-	-	-	-	-
	Lichttaster	x	-	-	-	-	-	-	-	x	-	-	x	-	-	-	-
	Barcodescanner	-	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-
	Förderband	-	-	-	-	x	x	-	-	-	-	x	-	-	-	-	-
	Haltezylinder	-	-	-	-	x	-	-	-	-	-	-	-	-	-	-	-
	Lager	-	-	-	-	-	-	x	-	-	-	-	-	-	-	-	-
	Roboter	-	-	-	-	-	-	-	-	-	x	-	-	-	-	-	-
	Greifer	-	-	-	-	-	-	-	-	-	x	-	-	-	-	-	-
	Kamerasystem	-	-	-	-	-	-	x	x	-	-	-	-	-	-	x	-
	Kontaktierstation	-	-	-	-	-	-	-	-	-	-	-	-	x	-	-	-
	Signallampe	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	-
	Datenbank	-	-	x	-	-	-	-	-	-	-	-	-	-	-	-	x
	Kategorie	K	T	K/D	F	T	T	K	K	K	F	T	K	S	K	M	D

F: Fügen, D: Datenaustausch, K: Kontrolle, M: MMI, S: Sonderoperation, T: Transport

6.2 Fähigkeiten in der Montage

6.2.1 Begriffsklärung

Elementare Fähigkeit (engl. *Skill*): stellt eine abstrakte Funktionalität dar, welche aus einer aufgabenspezifischen Sicht (Prozesssicht) beschrieben ist und durch eine Ressource angeboten wird. „Abstrakt“ bedeutet in diesem Zusammenhang eine hardware- und teilweise wirkprinzipunabhängige Beschreibungsform. Fähigkeiten sind elementar, beschreiben somit die Funktionalität von Produktionsressourcen, welche grundlegende Prozesse umsetzen, beispielsweise *Greifen*, *Bewegen*, *Anwesenheit prüfen*. Zur näheren Spezifikation besitzen Fähigkeiten Parameter, diese dienen der Ansteuerung oder Selbstbeschreibung der Ressource.

Prozessschritt (engl. *Process Step*): elementare Operation eines Produktionsprozesses, auf Basis dessen sich durchzuführende Produktionsabläufe funktional beschreiben lassen. Parameter spezifizieren den Prozessschritt und stellen somit Anforderungen an die umzusetzenden Ressourcen. Prozessschritte sind semantisch und strukturell das Pendant zu *Fähigkeiten* und deren Merkmalen, was die Grundlage eines Abgleichs zwischen beiden Domänen darstellt.

Aufgabe (engl. *Task*): beschreibt einen Produktionsprozess oder Teil hiervon und besteht aus einer definierten Sequenz von mindestens zwei Prozessschritten. Tasks¹ lassen sich wiederum zu (höherwertigen) Tasks verknüpfen und besitzen ein Set von Ein- und Ausgangsparametern, welches nicht per se mit den Parametern der Einzelprozessschritte übereinstimmen muss. Beispiele sind *Gehäuse montieren* oder *Bauteile kommissionieren*.

Zusammengesetzte Fähigkeit (engl. *Composite Skill*): durch einen Zusammenschluss von Ressourcen angebotene höherwertige abstrakte Funktionalität, die aus einer aufgabenspezifischen Sicht (Prozesssicht) beschrieben ist. Zusammengesetzte Fähigkeiten können mit Tasks und deren Parametern abgeglichen werden. Als Beispiel kann der Zusammenschluss eines Roboters (Fähigkeit *Bewegen*) und eines Greifers (Fähigkeit *Sichern*) die zusammengesetzte Fähigkeit *Bauteile kommissionieren* anbieten.

¹Nachdem „Aufgabe“ im deutschen Sprachgebrauch eine vielseitige Verwendung findet, wird zur Beschreibung in diesem Kontext die englische Bezeichnung verwendet.

Dienst (engl. *Service*): Ein Dienst ist die softwareseitige Umsetzung einer elementaren oder zusammengesetzten Gerätefähigkeit und entspricht der üblichen Definition aus der Informationstechnik (Abschnitt 3.1.2). In einer Automatisierungskomponente kann eine Fähigkeit bspw. mit OPC UA als Dienst im Netzwerk angeboten werden.

6.2.2 Vorgehensweise zur Herleitung

Als Ausgangsbasis werden die Taxonomien der Montagefunktionen nach LOTTER & WIENDAHL (2012) sowie die Normen VDI 2860 und DIN 8593 verwendet. Mit den unter Abschnitt 6.2.1 gegebenen Definitionen werden hieraus zwei Taxonomien für elementare und zusammengesetzte Fähigkeiten erstellt (→AF2). Die in Abschnitt 3.2.2 gewonnenen Erkenntnisse werden hierbei genutzt, um eine weitere Verfeinerung und klarere Trennung der Elemente zu erreichen.

Für den Bereich der Kontrollaufgaben wird eine gänzlich neue Strukturierung eingeführt, da die VDI 2860 als am häufigsten verwendete Quelle in dieser Kategorie die geringste Strukturierung ausweist (→AF6). Eine Kategorisierung wird nach der gewünschten Messaufgabe getroffen, hierzu werden die Auflistungen von HESSE & SCHNELL (2014) und HERING & SCHÖNFELDER (2012) konsolidiert. Zusätzlich werden die Funktionalitäten bekannter Bildverarbeitungssysteme untersucht und die Auflistung entsprechend ergänzt. Weitere Informationen diesbezüglich sind HAMMERSTINGL ET AL. (2015) zu entnehmen.

Da Prozessschritte und Tasks semantisch und strukturell mit der Fähigkeitenseite korrelieren, existieren zwei weitere (identische) Taxonomien, die nicht separat aufgeführt werden.

6.2.3 Elementare Fähigkeiten

Insgesamt werden sechs Hauptkategorien elementarer Fähigkeiten definiert (→AF1, Abbildung 6.1):

Im Bereich der *Handhabung* gelten Bewegen und Sichern als elementar. Die Verbindung beider Fähigkeiten erlaubt etwa Transport- oder Speicheraufgaben, die jedoch den zusammengesetzten Fähigkeiten zuzuordnen sind. Die von Bewegen beschriebene Trajektorie kann hierbei auf ein produkt- oder ein anlagenseitiges Koordinatensystem

bezogen sein. Erstere Art ist im Rahmen der Montageprozessmodellierung notwendig, da diese rein produktbezogen ist (zugehörige Ressourcen sind bis zum Zeitpunkt des Matchmakings unbekannt) und findet im Rahmen von Fügeprozessen (Abschnitt 6.2.4) häufig Anwendung. Letztere wird für Materialflussaufgaben in der Anlage eingesetzt.

Verbinden stellt mittels Form-, Stoff- oder Reibschluss einen Zusammenhalt zwischen zwei Partnern her. Als elementare Prozesse werden formschlüssige Verbindungen gesehen, wie etwa Schrauben. Fähigkeiten der anderen beiden Wirkprinzipien erfordern ein Zusammenspiel mehrerer Prozessarten (so besteht Löten aus Erwärmen und Dosieren) und sind deshalb den zusammengesetzten Fähigkeiten zuzuordnen.

Einen großen Bereich nehmen die *Kontrollaufgaben* ein. Hierbei wird generell davon ausgegangen, dass jede physikalische Größe sowohl mess- als auch prüfbar ist, weshalb sich für beide Unterkategorien identische Bäume ergeben. Die Aufgliederung erfolgt wie in Abschnitt 6.2.2 beschrieben. Da mehrere Dutzend messbarer Größen existieren, wird hier nur ein Auszug präsentiert.

Im Bereich des *Datenaustauschs* werden vier Fähigkeiten definiert (→AF7), abgeleitet von den informationstechnischen Grundoperationen persistenter Systeme (vgl. MASH-KOOR & FERNANDES 2007): Daten lesen, speichern, aktualisieren und löschen. Eine weitere Spezialisierung ist stark vom Dateninhalt abhängig und wird somit über die Parameter der Fähigkeiten beschrieben.

Die Fähigkeiten der *Mensch-Maschine-Interaktion* teilen sich auf in Anzeige- und Bedienelemente. Nach SCHMIDTKE & JASTRZEBSKA-FRACZEK (1993, S. 421) existieren hierbei vier Wirkprinzipien: visuell, akustisch, haptisch und kinästhetisch. Diese werden für eine Fähigkeitenunterscheidung genutzt (→AF7), der Anwender gibt somit im Prozess an, über welche Sinnesart er Informationen erhalten oder geben möchte (z. B. Daten Akustisch anzeigen). Eine weitere Spezialisierung, wie etwa der Darstellungsform, erfolgt auf Parameterebene.

Sonderfunktionen stellen alle weiteren elementaren Fähigkeiten in der Montage dar. Beispiele hierfür sind das Dosieren oder Bestrahlen.

Auszüge dieser Taxonomie sind von REINHART ET AL. (2017) und HAMMERSTINGL ET AL. (2015) veröffentlicht.

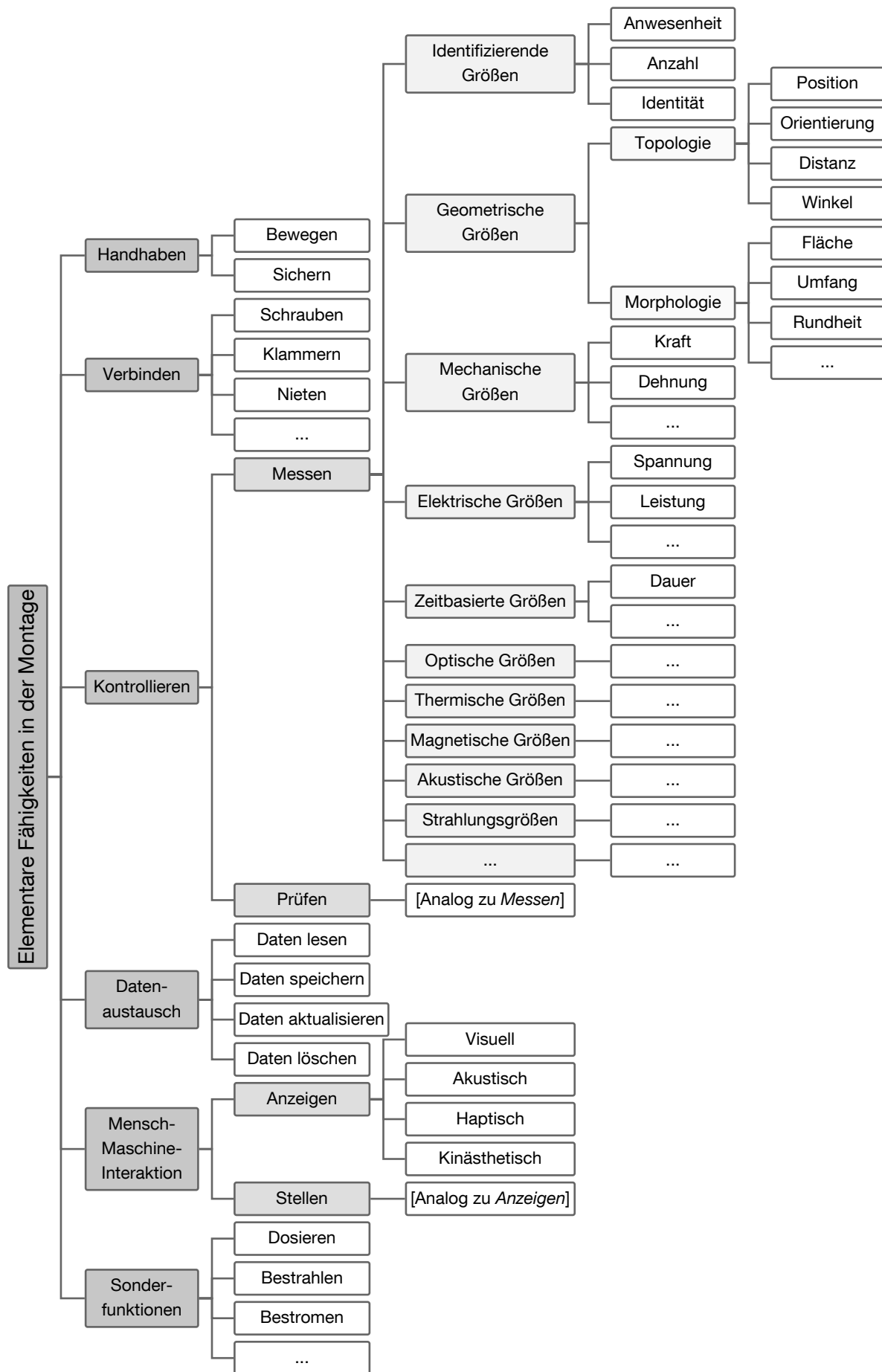


Abbildung 6.1: Elementare Fähigkeiten in der Montage (weiß)

6.2.4 Zusammengesetzte Fähigkeiten

Zusammengesetzte Fähigkeiten bestehen aus einer Sequenz von mindestens zwei elementaren oder zusammengesetzten Fähigkeiten. Durch beliebige Kombinationsmöglichkeiten können Anwender neue Fähigkeiten frei definieren – aus diesem Umstand kann eine vollständige Auflistung nicht gegeben werden. Die vorliegende Taxonomie in Abbildung 6.2 bietet aufgrund ihrer Herleitung aus bekannten Normen jedoch einen Grundstock an, der von Anwendern bei Bedarf erweitert werden kann.

Die *Handhabungsfähigkeiten* setzen sich hierbei aus einer Kombination der elementaren Bewegungs- und Sicherungsprozesse zusammen, hierzu gehören das Speichern, Transportieren und Menge verändern. *Fügeoperationen* stellen den größten Ast für zusammengesetzte Fähigkeiten in der Montage dar. Hierunter fallen alle Prozesse aus DIN 8593, wie z. B. das Schmelzschiessen oder Löten. In die dritte Kategorie werden erneut *Sonderoperationen* eingeordnet. Fähigkeiten in diesem Ast kombinieren meist elementare Sonderoperationen mit zusätzlichen Kontroll- oder Bewegungsfunktionen, z. B. Erwärmen = Bestrahlen + Temperatur messen.

Die Umsetzung zusammengesetzter Fähigkeiten innerhalb von Ressourcen kann hierbei auf zwei Arten realisiert werden:

a) *Blackbox-Betrachtung*: Eine fertig installierte Gerätekombination, wie etwa ein Montagemodul, bietet die zusammengesetzte Fähigkeit an. Die Aufteilung der Eingangsparameter zur Ansteuerung der Ressourcen erfolgt rein intern, z. B. durch ein vorgefertigtes Steuerungsprogramm. Dieser Ansatz ist einfach umzusetzen und erlaubt einen Abgleich der gesamten Gerätekombination, bietet jedoch keine Rekonfigurationsmöglichkeiten innerhalb des Ressourcenverbunds.

b) *Whitebox-Betrachtung*: Für jede elementare Fähigkeit innerhalb der Sequenz wird eine Ressource gesucht und diese anschließend physisch kombiniert, um einen entsprechenden Ressourcenverbund zu erhalten. Dieser Ansatz birgt eine wesentlich höhere Flexibilität im Rahmen des Ressourceneinsatzes, stellt jedoch auch größere Anforderungen an eine Modellierung. Folgende Inhalte müssen dabei im Modell hinterlegt werden (vgl. Abbildung 6.3):

- *Sequenz der Subfähigkeiten*: Soll mit dem System ein Produktivablauf beschrieben werden, muss eine Fähigkeitensequenz, inklusive Kontrollfluss (z. B. Parallelitäten, Verzweigungen), angegeben werden können.

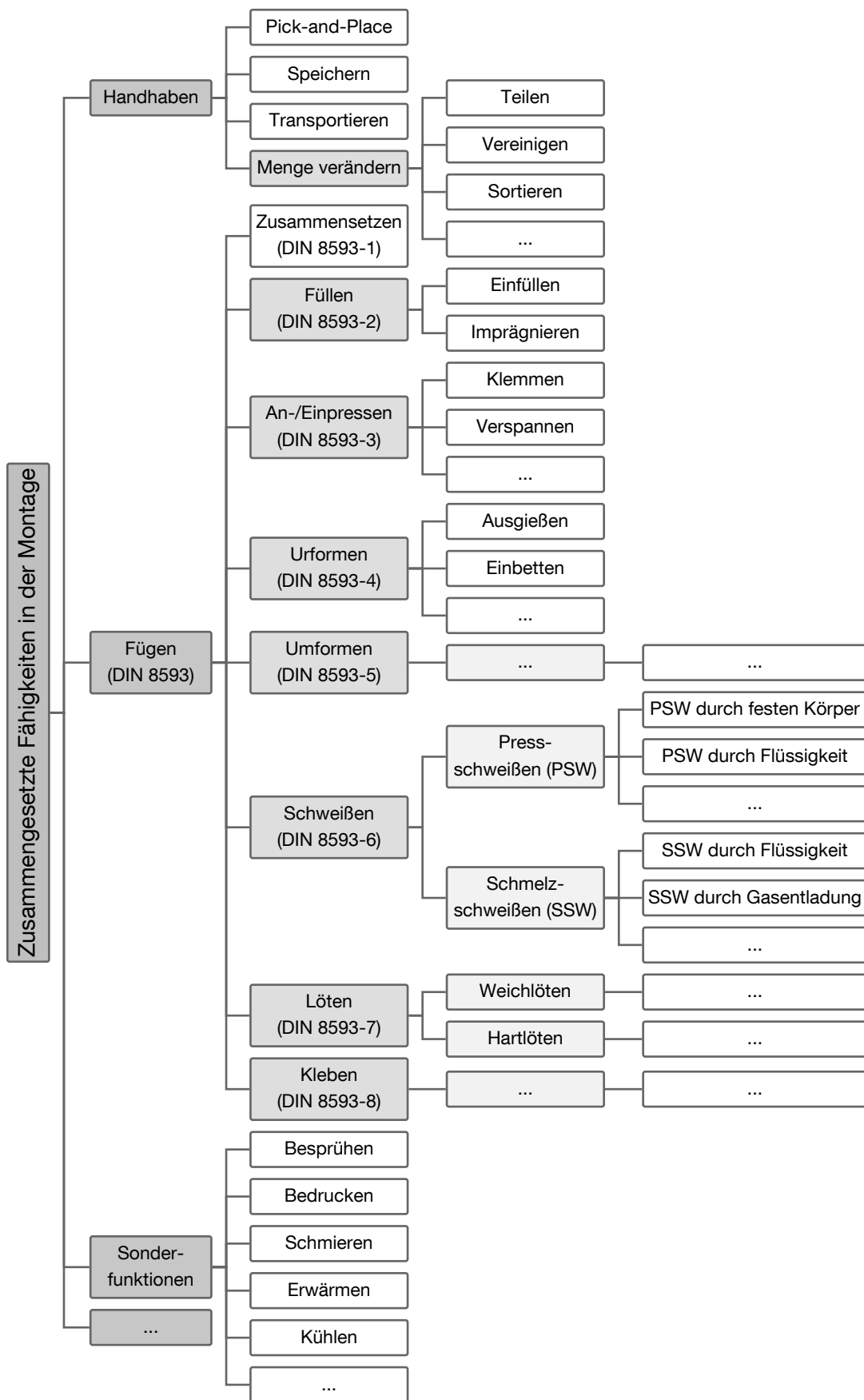


Abbildung 6.2: Zusammengesetzte Fähigkeiten in der Montage (weiß)

- *Ein- und Ausgangsparameter* der zusammengesetzten Fähigkeit: Diese müssen semantisch und typseitig nicht mit den Parametern der elementaren Fähigkeiten korrelieren, sondern können prinzipiell frei definiert werden.
- *Parameter-Transformationsfunktion*: In Anlehnung an den Gedanken von CAVIN & LOHSE (2014, S. 534 ff.) muss ein Transformationsmodul existieren, welches die übergeordneten Parameter wandelt, um von den Parametern der elementaren Fähigkeiten genutzt werden zu können. Je nach Definition der allgemeinen Parameter ist dies keine triviale Aufgabe, da die Transformationsfunktion unter Umständen von Wirkprinzip und Anordnung der Gerätekombination abhängt.
- *Template-Zuordnung von Ressourcen*: Kommt in der Sequenz von Subfähigkeiten eine Fähigkeit mehrmals vor, muss über ein Templatesystem festgelegt werden, ob bei einer späteren Ressourcenzuweisung für beide Fähigkeiten das identische Gerät verwendet werden soll. Bei einer Pick-and-Place-Aufgabe (Sichern → Bewegen → Entsichern) ist es beispielsweise notwendig, dass beide Sicherungsprozesse von der gleichen Greifressource ausgeführt werden.
- *Beschreibung der räumlichen Zuordnung*: Innerhalb der Fähigkeitensequenz muss angegeben werden, ob Ressourcen mechanisch verbunden sind (insbesondere bei Gerätehierarchien wie Roboter → Endeffektor), teilweise auch unter Angabe ihrer räumlichen Beziehungen (etwa bei Sensorsystemen).

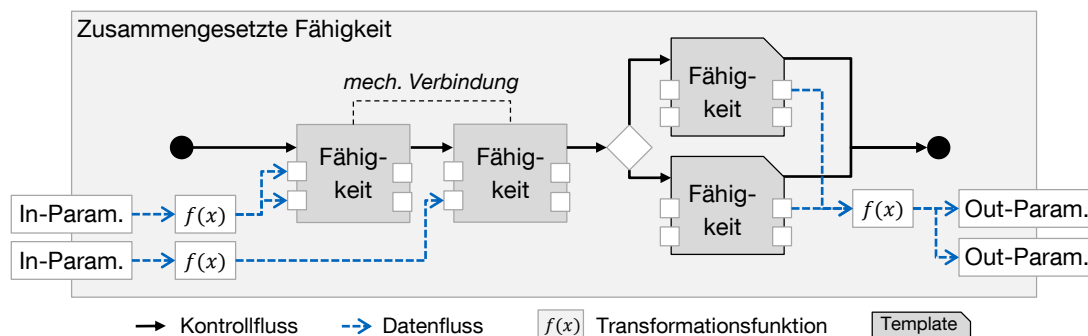


Abbildung 6.3: Bestandteile zusammengesetzter Fähigkeiten

6.2.5 Parameter von Fähigkeiten

Fähigkeiten (und deren Prozessschritt-Pendants) besitzen Ein- und Ausgangsparameter. Im Rahmen der Spezifikation des Montageprozesses stellen *Eingangsparameter* Anforderungen dar, welche durch die Eigenschaften von Ressourcen erfüllt werden müssen. Auf Ressourcenseite werden diese Eigenschaften mit den identischen „Eingangsparametern“

der Fähigkeit beschrieben – hierdurch ist ein automatisierter Vergleich möglich. Die Wertedefinition unterscheidet sich jedoch zwischen beiden Domänen:

Auf Prozessseite wird ein *Wertebereich* spezifiziert, mit dem Anwender Prozessbereiche, Produktvarianten oder -schwankungen definieren können, z. B. Prozessschritt Bewegen: Bahnabweichung $\leq 1 \text{ mm}$ ². Auf Ressourcenseite wird auf Basis der Prozessanforderungen und aktuellen Ressourcenkombination pro Parameter ein *Einzelwert* berechnet, z. B. Fähigkeit Bewegen: Bahnabweichung = 0,98 mm, und durch einen Matchmaking-Algorithmus verglichen (vgl. Kapitel 7). Informationstechnisch beschrieben, existiert eine Fähigkeitenklasse, die jeweils als Objekt auf Prozess- und Ressourcenseite instanziiert ist und deren Attribute verglichen werden. Nachdem Fähigkeiten und Prozessschritte semantisch und strukturell die gleiche Klasse beschreiben, wird zur Vereinfachung der Lesbarkeit in den nachfolgenden Abschnitte nur mehr von Fähigkeiten gesprochen.

Ausgangsparameter liefern das Ergebnis einer Fähigkeit, und sind somit abhängig von deren Ausführung. Demnach können sie nicht durch den Anwender spezifiziert werden, lassen sich aber als Eingangsgröße für nachfolgende Fähigkeiten nutzen, bspw. das Ergebnis einer Positionsmessung als Eingang für eine Bewegen-Fähigkeit.

Jeder prozessbezogene Parameter besitzt dabei folgendes Set an Charakteristika:

Input/Output	Bezeichner	Datentyp	Wert <i>oder</i> Wertebereich	Einheit	Notwendigkeit (optional/verpflichtend)
--------------	------------	----------	-------------------------------	---------	--

Neben Parametern, die eine Fähigkeit bzw. Prozess alphanumerisch definieren („wie“), ist zusätzlich eine *Geometriebeschreibung* erforderlich, die angibt, „wo“ an einem Produkt die Fähigkeit ausgeführt werden soll. Beispiel sind die Festlegung der produktseitigen Haltebereiche bei der Fähigkeit *Sichern* oder der Messpunkt einer Kontrolloperation, etwa von *Oberflächenrauigkeit Prüfen*. Dies kann über die Angabe von Punkten, Linien, Flächen oder Volumina geschehen und wird allgemein als *Region-of-Interest (ROI)* bezeichnet. Zusätzlich werden ein Richtungsvektor (meist die Normale) sowie eine Winkeltoleranz angegeben (Abbildung 6.4). Sie bestimmen die Aktionsrichtung der zugehörigen Fähigkeit. Beide Aspekte sind notwendig, um ein räumliches Matchmaking von Ressourcen über deren Arbeitsbereiche durchzuführen (vgl. Kapitel 7). Die ROI wird rein produktseitig anhand des aktuellen CAD-Modells definiert, deshalb ist sie für

²Dies würde informationstechnisch einem Wertebereich von 0 – 1 mm entsprechen.

alle direkten Prozesse erforderlich. Indirekte Prozessschritte wie *Datenaustausch* und *MMI* benötigen sie nicht.

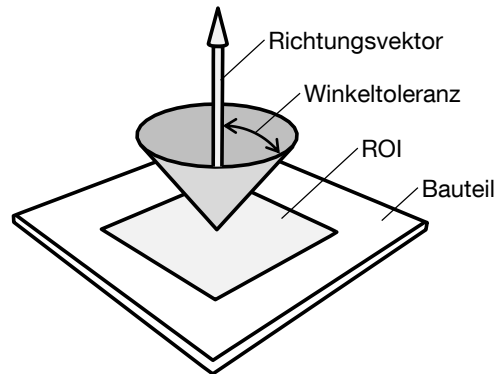


Abbildung 6.4: Auf Bauteil definierte ROI zur Angabe des Fähigkeitenwirkbereichs

Eine weitere Parameterklasse stellen alphanumerische Produktparameter dar. Sie liefern ergänzende Informationen für das Matchmaking und können direkt aus dem PDM-System abgeleitet werden. Beispiele hierfür sind das Bauteilgewicht oder der Werkstofftyp. Abbildung 6.5 zeigt die vorgestellte Parameterstruktur in der Übersicht (→AF3).

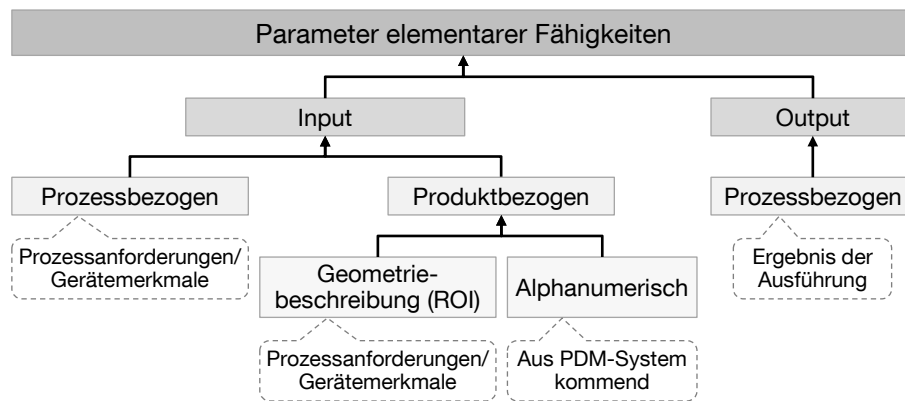


Abbildung 6.5: Übersicht der Parameter elementarer Fähigkeiten

Methodik zur Herleitung

Die Herleitung *prozessbezogener* Parameter kann auf zwei Arten erfolgen:

Methode A: Es werden typische Ressourcen bez. einer Fähigkeit gesammelt, die Menge aller Geräteparameter ergibt das Set für die jeweilige Fähigkeit. Dieser Ansatz mischt die reine Parametersemantik mit deren spezifischer Verarbeitung, wie etwa Einheit

und Befehlssatz. Sie stellt in jedem Fall die Ausführbarkeit auf allen gefundenen Ressourcen ohne Hardwareanpassungen sicher, hat jedoch den Nachteil, dass aufgrund der fehlenden Standardisierung eine sehr große Menge an heterogenen Parametern entsteht. Eine Ausführbarkeit könnte nur sichergestellt werden, wenn alle Parameter angegeben werden, damit das Subset des jeweilig verwendeten Geräts in jedem Fall enthalten ist. Dies erzeugt einen hohen, potentiell nutzlosen, Modellierungsaufwand. Weiterhin könnten Ressourcen, welche in der eingangs untersuchten Menge fehlen, die Fähigkeit mitunter nicht ausführen, da deren Parameterspezifika nicht angegeben sind.

Diese Defizite führen zu *Methode B*, bei der Fähigkeitenparameter rein prozessbezogen und physikalisch beschrieben werden. Hierdurch werden nur für den Prozess notwendige Spezifika getroffen (Semantik) und somit eine Hardwareneutralität erreicht. Aus Gerätesicht ergibt sich jedoch eine Informationslücke durch nicht per se interpretierbare und eventuell fehlende Parameter.

Für eine Definition der Parameter wird Methode B genutzt (\rightarrow AF4). Die Informationslücke wird mit zwei Ansätzen geschlossen: Geräteparameter, die bereits über die Fähigkeit abgedeckt werden, können über den treiberbasierten Ansatz aus Kapitel 5 in Gerätebefehle gewandelt werden. Zu einer Ausführung fehlende Parameterwerte muss das Gerät selbstständig akquirieren. Dies kann beispielsweise durch einen Teach-In, externen Softwarealgorithmus/Treiber oder Interaktion mit dem Anwender geschehen und ist entsprechend als Anforderung an die aufgabenorientierte Planung und Programmierung aufzunehmen.

Pro Fähigkeit werden aus Prozesssicht Parameter gesammelt und deren Charakteristika spezifiziert. Dies kann mithilfe von Normen geschehen, bspw. der DIN 1319-1 für Kontrollprozesse. Im Anschluss erfolgt eine Verifikation mit ausgewählten Geräten, um eine möglichst hohe Überdeckung zum Parameterset der Ressourcenseite zu erhalten. Dieser Prozess wird iterativ wiederholt, bis ein geeignetes Parameterset gefunden wurde. Abbildung 6.6a zeigt das Vorgehen in der Übersicht.

Die Bestimmung *produktbezogener* Fähigkeitenparameter (\rightarrow AF4) wird vorrangig unter dem Gesichtspunkt des Matchmakings notwendig: Neben Prozessangaben werden auch Produktinformationen benötigt, um eine valide Aussage zu einer Geräteeignung treffen zu können. Ob ein Induktionssensor mit der Fähigkeit Anwesenheit prüfen für einen konkreten Fall geeignet ist, kann bspw. nur ermittelt werden, wenn die Materialkennwerte des zu detektierenden Objekts bekannt sind.

Die Herleitung produktbezogener Parameter geschieht aus den Charakteristika physikalischer Wirkprinzipien. In einem ersten Schritt werden hierzu mögliche Wirkprinzipien zur Fähigkeitsausführung gesammelt. Um die spätere Parameterzahl zu reduzieren, erfolgt im Anschluss die Suche nach Wirkprinzipien von Geräten, welche diese Fähigkeit auch tatsächlich umsetzen. Die gefundene Menge wird dann in für die Fähigkeitsausführung mögliche Funktionsprinzipien eingeteilt. Funktionsprinzipien stellen eine Untergruppe innerhalb des jeweiligen Wirkprinzips dar (Beispiel s. Abbildung 6.6b). Für jedes Prinzip werden abschließend die dafür notwendigen physikalischen Bauteilinformationen spezifiziert, die in Summe die produktbezogenen Parameter der jeweiligen Fähigkeit ergeben. Werte für diese Parameter müssen im PDM-System des Unternehmens vorgehalten werden. Vorteilhaft ist hierbei, dass oftmals ein Charakteristikum (z. B. Materialkennwert) für mehrere Wirkprinzipien bzw. Fähigkeitenparameter genutzt werden kann.

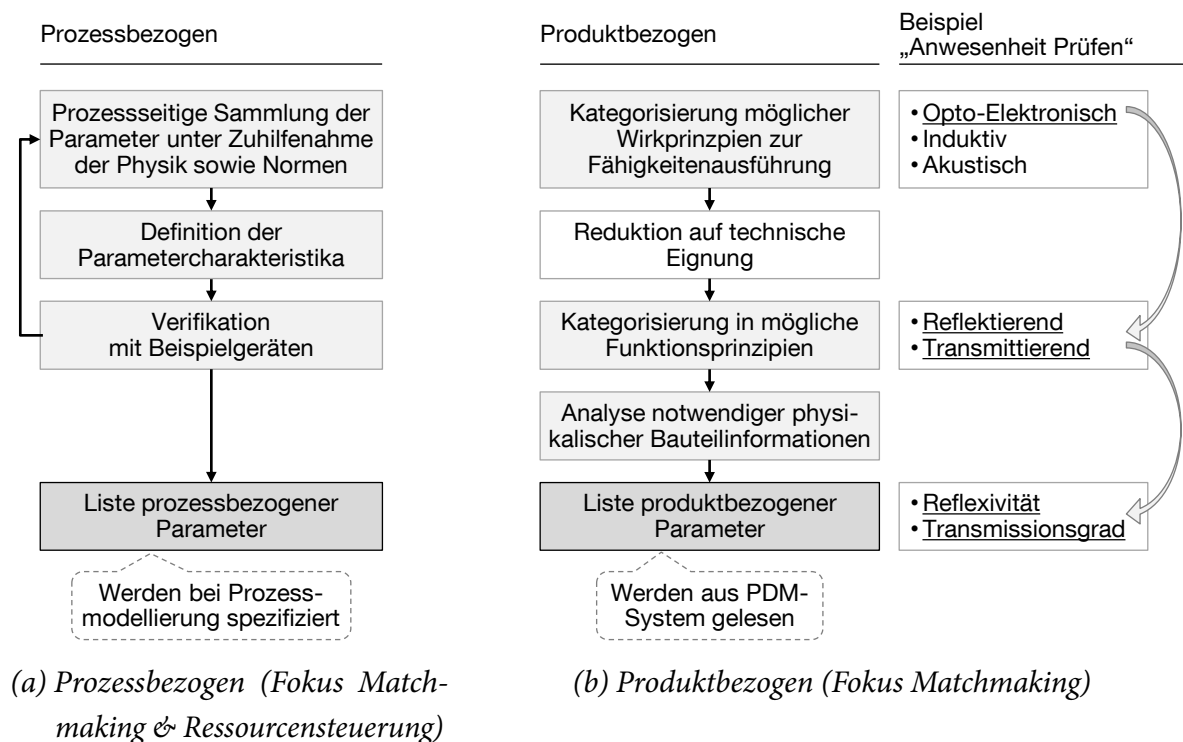


Abbildung 6.6: Methodik zur Definition der Fähigkeitenparameter

Grundparameter elementarer Fähigkeiten

Auf Basis der im letzten Abschnitt definierten Methode können Parameter zu den in Abbildung 6.1 gelisteten Fähigkeiten bestimmt werden. Aufgrund des hohen Umfangs wird

eine Darstellung in Anhang A.3 gegeben. Bei der Parameterermittlung treten Gemeinsamkeiten auf, die nachfolgend als Grundparameter vorgestellt werden (→AF5). Hierbei wird eine generelle Aufteilung zwischen aktorischen und sensorischen Fähigkeiten getroffen:

Die Analyse *aktorischer Fähigkeiten* zeigt eine große Parameterheterogenität – als allgemeine Eingangsparameter können die *maximale Ausführzeit* sowie die *Region-of-Interest (ROI)* bestimmt werden. Als Ausgangsparameter geben alle Fähigkeiten ihren *Ausführungsstatus* und den *-fortschritt* wieder (Tabelle 6.2).

Tabelle 6.2: Grundparameter aktorischer Fähigkeiten

	Parameter	Notwendigkeit	Datentyp	Einheit
Input	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	v	real	ms
Output	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

v: verpflichtend, o: optional

Der *Status* aller Fähigkeiten wird als Zustandsautomat modelliert, der sich an der bekannten Norm ANSI/ISA-88 anlehnt (Abbildung 6.7). Als zusätzlicher Zustand wurde *Not configured* aufgenommen, in welchem sich das Gerät initial vor seiner Parametrierung befindet. Durch Aufruf des Einrichtungsprozesses, im Zuge dessen fehlende Geräteinformationen akquiriert werden, schaltet das Gerät in *Idle*. Über den Aufruf eines *Factory Resets* kann das gestoppte Gerät wieder in den *Not configured*-Zustand gelangen.

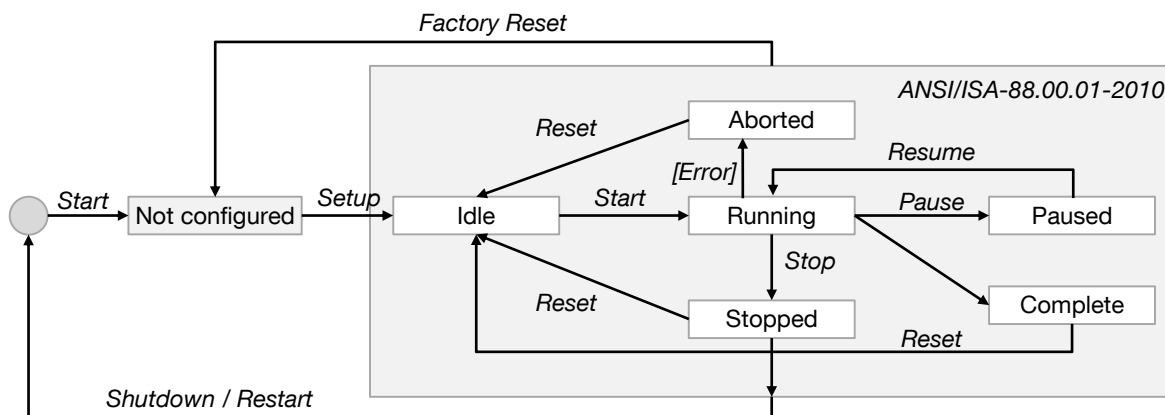


Abbildung 6.7: Zustandsautomat von Fähigkeiten (Erweiterung von ANSI/ISA-88.01-2010)

Sensorische Fähigkeiten werden in Mess- und Prüfaufgaben unterschieden. Allen *Messaufgaben* gemein sind die anzugebende *Einheit* des Ergebnisses sowie die maximale *Messunsicherheit*. Hinzu kommen die von aktorischen Fähigkeiten bekannten Grundparameter *ROI* und *Ausführzeit*. Als Output wird neben den allgemeinen Fortschrittsmerkmalen das *Ergebnis* als numerischer Wert zurückgeliefert, in der zuvor angegebenen SI-Einheit (Tabelle 6.3).

Tabelle 6.3: Grundparameter sensorischer Fähigkeiten: Messen

	Parameter	Notwendigkeit	Datentyp	Einheit
<i>Input</i>	Einheit	v	string	[SI-Einheit]
	Messunsicherheit	v	real	[SI-Einheit] oder Prozent
	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	v	real	ms
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%
	Ergebnis	v	real	[SI-Einheit]

v: verpflichtend, o: optional

Bei *Prüffähigkeiten* ist generell ein *Prüfwert* mit SI-Einheit anzugeben, gegen den die aktuelle Messung verglichen wird. Die *Prüftoleranz* spezifiziert dabei den Bereich, innerhalb dessen eine Prüfung erfolgreich ist (z. B. *Prüfe Abstand* = 2 mm ± 0,1 mm). Über die *Prüfbedingung* wird angegeben, ob der aktuelle Messwert innerhalb, außerhalb oder gleich der Prüfmenge sein muss. Das *Prüfergebnis* stellt einen booleschen Wert dar (Tabelle 6.4).

Tabelle 6.4: Grundparameter sensorischer Fähigkeiten: Prüfen

	Parameter	Notwendigkeit	Datentyp	Einheit
<i>Input</i>	Prüfwert	v	real	[SI-Einheit]
	Prüfbedingung	v	string	[Operator]
	Prüftoleranz	v	string/real	[SI-Einheit] oder Prozent
	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	v	real	ms
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%
	Ergebnis	v	bool	true/false

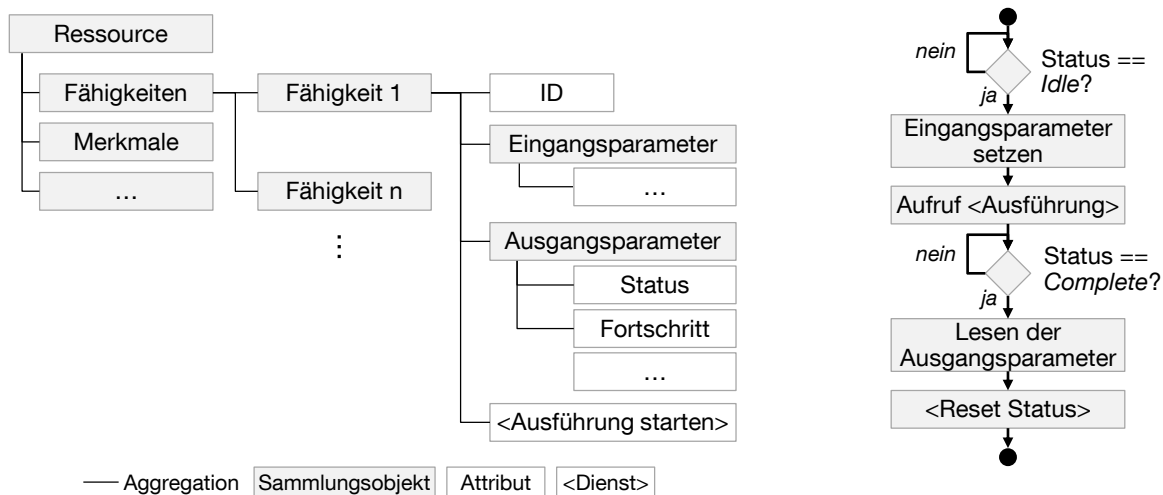
v: verpflichtend, o: optional

Grundparameter zusammengesetzter Fähigkeiten

Aufgrund der freien Kompositionsmöglichkeit zusammengesetzter Fähigkeiten können beliebig weitere und abstraktere Elemente definiert werden, mit entsprechend freier Wahlmöglichkeit der Parameter. Eine allgemeine Angabe von Grundparametern ist deshalb nicht zielführend. Die in den vorherigen Abschnitten beschriebene Methodik zur Herleitung sowie die Auflistung von Grundparametern können jedoch als Hilfestellung bei einer Definition dienen. Parameter von ausgewählten zusammengesetzten Fähigkeiten können Anhang A.4 entnommen werden.

6.3 Fähigkeiten im Ressourcen-Informationsmodell

Die Publikation des Digital Twins geschieht durch die Plug&Produce-Software im Netzwerk des Montagesystems (Kapitel 5). Im Informationsmodell des Digital Twins werden Fähigkeiten als Sammlungsobjekte („Ordner“) modelliert, welche die zugehörigen Ein- und Ausgangsparameter als Attribute enthalten (Abbildung 6.8a). Zusätzlich existiert ein ID-Attribut, welches die Fähigkeit eindeutig und sprachunabhängig identifiziert.



(a) Struktur

(b) Ablauf der Ansteuerung
(aus Sicht des aufrufenden Teilnehmers)

Abbildung 6.8: Fähigkeiten im Ressourcen-Informationsmodell

Zur Ausführung einer Ressourcenfähigkeit müssen zuerst die gewünschten Eingangsparameter gesetzt und im Anschluss ein generischer „Ausführungsdienst“ über das

Netzwerk aufgerufen werden. Die Fähigkeitenausführung erfolgt hierbei asynchron, um das aufrufende System während der Ausführung nicht zu blockieren: Zuerst überprüft die aufrufende Ressource, ob das Gerät bereit ist (Status *Idle*). Im positiven Fall werden alle Eingangsparameter gesetzt und im Anschluss der Ausführungsdienst im Informationsmodell aktiviert. Daraufhin beginnt die angesprochene Ressource mit der Abarbeitung, während die aufrufende Ressource den Fähigkeitenstatus überwacht. Ist die Abarbeitung beendet (Status *Complete*), liest die aufrufende Ressource die Ausgangsparameter und setzt die aufgerufene Ressource zurück (vgl. Abbildung 6.8b).

7 Aufgabenorientierte Ressourcenauswahl und Programmierung

7.1 Anforderungen

7.1.1 Anforderungen an die Ressourcenauswahl

Es wird ein Softwareframework benötigt, welches einen modellierten Montage- und Prüfplan mit Ressourcenfähigkeiten abgleicht (Anforderung Ressourcenauswahl AR1). Die durch den Anwender spezifizierten Randbedingungen müssen hierbei mit den Merkmalen von Ressourcen verglichen werden, um nicht nur prinzipiell fähige Komponenten zu finden, sondern auch für den aktuellen Fall geeignete (Matchmaking, AR2).

Die Berechnung der Ressourceneignung bezogen auf einen spezifischen Prozess stellt ein komplexes Problem dar, da bspw. interne Hardwareeigenschaften, die Interaktion von Komponenten, Prozesswissen oder allgemeine physikalische Gesetzmäßigkeiten gegeneinander abgewogen werden müssen. Dies bedeutet, dass ein Abgleich nicht zwingend eine vollständige Eignungsaussage liefern kann. Die Systemtheorie schlägt zur Lösung komplexer Probleme eine hierarchisierte Problemstrukturierung vor („Vom Groben zum Detail“, DAENZER & HABERFELLNER 2002, S. 30 ff.). Übertragen auf den Anwendungsfall soll die Frage der Geräteeignung in entkoppelte Teilprobleme zerlegt werden. Für kleinere Problemumfänge lassen sich einfachere und robustere Lösungsstrategien entwickeln, außerdem können nicht-automatisiert lösbare Probleme (bspw. aufgrund einer nicht vollständigen Systembeschreibbarkeit) leichter identifiziert werden. Das Konzept muss deshalb die Lösung von Teilproblemen beherrschen (AR3). Weiterhin gilt es, die Arten der Teilprobleme in der Montage für eine spätere Lösungsfindung zu klassifizieren (AR4).

Es kann dabei der Fall auftreten, dass die Lösung eines Teilproblems von der Lösung anderer Teilprobleme abhängig ist. So kann bspw. ein Greifer dessen Eignung für einen sicheren Griff nur bestimmen, wenn die auf ihn wirkenden Kräfte und Momente bekannt sind. Diese sind u. a. vom verwendeten Handhabungsgerät und Produkt abhängig, welches sich wiederum aus dem Einsatzort ergibt. Das Konzept muss somit die Verknüpfung von Problemen und Rückfragen unterstützen (AR5). Für den Fall, dass einzelne Teilprobleme nicht automatisiert gelöst werden können, gilt es eine effiziente Erweiterbarkeit für neue Problemlösungsalgorithmen vorzusehen (AR6) und zusätzlich den Mensch mit seinem Erfahrungswissen als zusätzliche Problemlösungsinstanz in das System zu integrieren (AR7).

Damit auch eine Eignung von (noch) nicht physisch vorhandenen Ressourcen überprüft werden kann, bspw. zur Einplanung neuer Betriebsmittel, muss das Konzept das Matchmaking mit rein virtuellen Geräten unterstützen (AR8). Hinzu kommt eine Unterstützung der Überprüfung von kombinierbaren Ressourcen (AR9).

7.1.2 Anforderungen an die Programmierung

Zur Reduktion von Aufwänden in der Softwareentwicklung soll sich die Anlage aus einem fähigkeitenbasierten Modell des Montageablaufs selbst programmieren. Das Modell ist hierbei so zu wählen, dass es effizient aus vorliegenden Planungsdaten erstellt werden kann (Anforderung Programmierung AP1).

Da die Modellierung rein produkt-/prozessseitig geschieht, ist bei der Modellerstellung nicht bekannt, welche Ressourcen den Ablauf zukünftig ausführen werden. Hieraus folgt, dass der Materialfluss zwischen den Ressourcen durch das Framework automatisiert generiert und dem Modell hinzugefügt werden muss (AP2). Geeignete Materialflussgeräte sollen mit dem Konzept ebenso auffindbar sein wie prozessnahe Ressourcen (AP3). Existieren mehrere Materialflusspfade, so gilt es, diese dem Anwender für eine Auswahl zu präsentieren (AP4). Da sich die Abfolge des Materialflusses unter Umständen verschieden zur logischen Abfolge der Montageprozesse verhält, ist hierfür eine getrennte Modellierungsmöglichkeit vorzusehen (AP5). Dies trifft auch für die Möglichkeit der Verknüpfung von Prozesswerten zu (Datenfluss), bspw. das Ergebnis einer Positionsmessung als Zielkoordinate für die nachfolgende Bewegung eines Aktors (AP6).

Der fertig modellierte und zugewiesene Ablauf muss abschließend auf die jeweiligen Steuerungseinheiten für den Ablauf des Montageprozesses übertragen werden. Hierfür ist ein Codegenerator vorzusehen, welcher das Prozessmodell in kompatiblen Steuerungscode überführt (AP7). Benötigt ein Gerät Informationen zur Ausführung, die weder durch die prozessseitige Fähigkeitenspezifikation gegeben noch durch das Software-Framework generiert wurden, so muss ein Konzept existieren, um dieses Informationsdefizit zu beheben (vgl. Abschnitt 6.2.5, AP8).

7.2 Merkmale im Ressourcen-Informationsmodell

Das Informationsmodell des Digital Twins stellt die wesentliche Quelle für das Softwareframework dar, um relevante Größen für die Ressourcenauswahl und Programmierung zu akquirieren. Um eine möglichst breite Anwendbarkeit des Digital Twins zu schaffen, werden die im Betrachtungsraum vorgestellten Informationsdomänen aus Abschnitt 3.2.3 zu einem gemeinsamen Modell konsolidiert. Ein wesentliches Unterscheidungsmerkmal stellt dabei der Umstand dar, dass es sich hierbei um ein online zugreifbares Live-Abbild der Ressource handelt, weshalb die Verwendung von dateibasierten Standards wie AutomationML oder COLLADA nur vereinzelt sinnvoll ist. Abbildung 7.1 zeigt die resultierende Struktur im Überblick. Für die jeweiligen Merkmalsausprägung wird nach Möglichkeit auf bekannte Normen verwiesen, die letztlich informationstechnische Implementierung ist abhängig vom gewählten Kommunikationsstandard. Im Falle von OPC UA kann eine Umsetzung bspw. innerhalb des Begleitstandards „OPC UA for Devices“ (OPC 2013) entstehen, unter dem auch PLCopen eingegliedert ist.

Das hier vorgestellte Modell gibt den standardisierten Teil an, welcher der Interoperabilität von Ressourcen und IT-Anwendungen dient. Anwender und Komponentenhersteller können für spezialisiertere Anwendungsfälle problemlos weitere Merkmale hinzufügen. Werden Merkmale durch Ressourcen nicht benötigt (etwa die Achsenwerte bei Sensoren), dann sind entsprechende Einträge als leer gekennzeichnet.

Neben den in Kapitel 6 beschriebenen Fähigkeiten existieren in dem Modell weitere Dienste, deren Aufgabe die Konfiguration der Ressource ist. Hierzu zählen allgemeine Funktionen zur Gerätekontrolle (Start, Stop, Reset etc.) sowie der „Setup“ genannte Dienst, welcher eine nutzerzentrierte Einrichtung des Geräts ermöglicht. Dem Vorbild

des Self-Service aus dem Consumer-Bereich folgend (etwa bei Kopiersystemen), wird diese Schnittstelle genutzt, um eine geräteseitige Parametrierung mithilfe des Anwenders durchzuführen. Dieses Konzept wird bspw. für die Sammlung fehlender Informationen zur Fähigkeitsausführung genutzt (→AP8) und in Abschnitt 7.3.5 näher beschrieben.

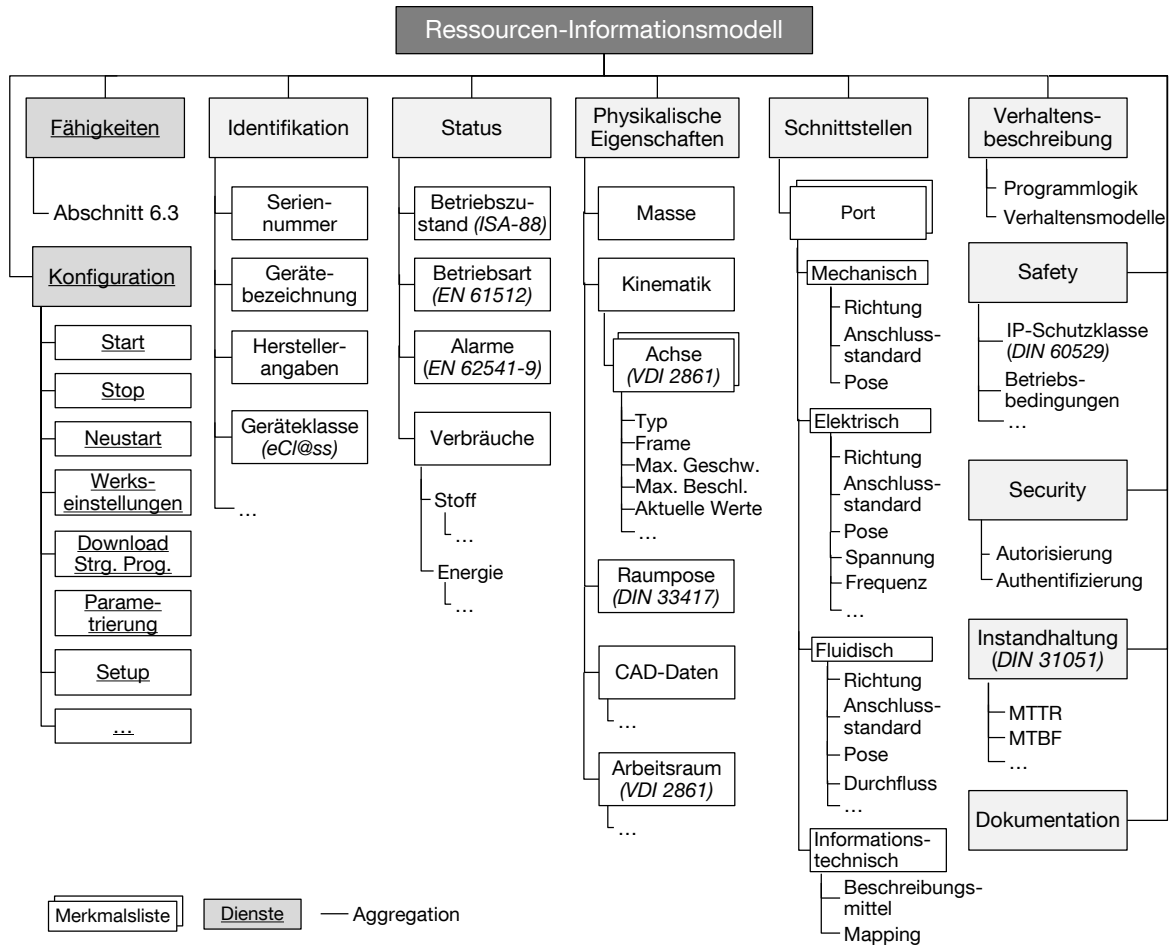


Abbildung 7.1: Ressourcenseitiges Informationsmodell

Das beschriebene Informationsmodell wurde in Auszügen in HAMMERSTINGL ET AL. (2016) und REINHART ET AL. (2017) veröffentlicht.

7.3 Konzept zur aufgabenorientierten Ressourcenauswahl und Programmierung

7.3.1 Gesamtablauf

Der Ablauf zur aufgabenorientierten Ressourcenauswahl und Programmierung gliedert sich in eine Sequenz aus neun Schritten:

1. *Modellierung des Montageprozesses (Abschnitt 7.3.3)*: Ausgehend von Planungsdaten wird der durchzuführende Montageprozess als Verkettung von Montagefähigkeiten innerhalb des Frameworks erstellt. Der Anwender wählt hierzu Prozessschritte aus, welche dem in Kapitel 6 vorgestellten Konzept entsprechen. Es entsteht ein gerichteter Graph, der anlagenneutral beschrieben ist und sowohl sequentielle als auch parallele Arbeitsschritte enthält. Hierbei existieren drei Unterphasen:
 - *Automatisierter Aufbau des Graphen*: Zur Reduktion des Modellierungsaufwands können Ansätze wie bspw. Assembly-by-Disassembly genutzt werden, um automatisiert Fügeschritte abzuleiten.
 - *Manuelle Ergänzung und Verbesserung*: Der erstellte Graph wird durch den Anwender verbessert und um nicht-automatisiert bestimmbare Schritte ergänzt (z. B. Mess- und Prüfaufgaben, Sonderoperationen, Anzeige von Prozesswerten).
 - *Spezifikation der Prozessschritte*: Für jeden Prozess definiert der Anwender weitergehende Parameter bzw. Anforderungen, wie etwa Taktzeit oder Messgenauigkeit.
2. *Suche nach Gerätekombinationen (Abschnitt 7.3.5)*: Dies geschieht auf Basis der angegebenen Schnittstellen im Digital Twin. Für bereits verbundene Systeme werden die sich ergebenden Merkmale und Fähigkeiten des Verbunds direkt berechnet, noch nicht verbundene Ressourcen werden als „Virtuelle Verbünde“ eingetragen – eine Aufforderung zur Kombination geschieht durch das Framework nach einer eventuellen Auswahl.
3. *Festlegung geeigneter Ressourcen (Abschnitt 7.3.5)*: Auf Basis der definierten Prozesssequenz und deren Parameter werden für jeden Prozessschritt geeignete Geräte aus dem digitalen Fabrikabbild gesucht. Der Benutzer wählt aus einer Liste an geeigneten Geräten den bevorzugten Kandidat aus.
4. *Generierung Materialflüsse (Abschnitt 7.3.5)*: Nachdem prozessnahe Ressourcen zur Ausführung bestimmt worden sind, generiert ein Algorithmus die notwendigen

- Pfade zwischen den Stationen (→AP2). Dies schließt die geometrische Zuordnung von Transportressourcen zu Materialpfaden mit ein.
5. *Festlegung geeigneter Transportgeräte (Abschnitt 7.3.5)*: Analog zu Schritt 3 werden geeignete Transportressourcen durch das System bestimmt (→AP3). Der Nutzer bekommt die zugehörigen Transportpfade und deren Merkmale (z. B. Transportzeit) präsentiert und kann eine für ihn optimale Kombination auswählen (→AP4).
 6. *Generierung der Handhabungs- und Transportprozesse (Abschnitt 7.3.5)*: Ressourcen besitzen u. U. verschiedene Fähigkeitensequenzen für den Transport. Nachdem Materialfluss und Transportgeräte in den vorherigen Schritten bestimmt worden sind, kann das Framework die zugehörigen Fähigkeiten aus den Digital Twins laden und in den Montageprozess einfügen.
 7. *Ressourcenparametrierung (Abschnitt 7.3.5)*: Geräte können u. U. ihre Eignung erst bestimmen, nachdem eine Parametrierung durchgeführt worden ist. Gleiches gilt für einen Einsatz im Produktivbetrieb. Dieser Aspekt wird mittels bekannter Parametrierungsstandards (Abschnitt 2.2.1) umgesetzt oder geschieht auf Basis der Setup-Dienste im Digital Twin (→AP8).
 8. *Iteration*: Wurden durch den Materialflussalgorithmus nicht zugewiesene Prozessschritte eingebracht (z. B. Bestimmung der Produktpose) wird für diese Schritte erneut eine Ressourcensuche durchgeführt. Gleiches gilt für Ressourcen, die sich nach einer Parametrierung als nicht mehr geeignet gezeigt haben. Der Prozess wird solange wiederholt, bis alle Prozessschritte zugewiesen worden sind.
 9. *Codegenerierung und Export (Abschnitt 7.3.6)*: Nach abgeschlossener Geräteauswahl wird der Montageprozess in Steuerungscode umgewandelt und an die Steuerungseinheiten geschickt. Hierzu werden die in Abschnitt 7.2 definierten Konfigurationsdienste verwendet.

7.3.2 Architektur des TOMAP-Frameworks

Das zu erstellende Software-Framework der aufgabenorientierten Ressourcenauswahl und Programmierung (engl. Task-oriented Matchmaking and Programming, TOMAP) wird aufgrund des Umfangs als Schichtenarchitektur entworfen. Ähnlich wie bei dem in Abschnitt 2.2.1 vorgestellten ISO/OSI-Modell sind Schichten innerhalb der Software für dezidierte Aufgaben verantwortlich und weisen nach außen hin klar definierte Schnittstellen auf. Hierdurch steigt die Modularität und Wiederverwendbarkeit der

Softwarebestandteile (RICHARDS 2015). Nach RICHARDS (2015, S. 1 f.) bestehen die meisten Konzepte aus vier dezidierten Schichten, die auch hier Anwendung finden:

Die *Präsentationsschicht* enthält die Eingabemasken zur Steuerung der Software. Neben der Modellierung des Montageablaufs kann der Anwender die Geräteeignung testen, Materialflüsse bestimmen und den finalen Ablauf auf die Steuerungssysteme laden lassen.

In der *Geschäftsschicht* befinden sich die Algorithmen des Systems. Das *Modelling Framework* erlaubt die Beschreibung und Persistierung von Montageprozessen, es wird in Abschnitt 7.3.3 näher vorgestellt.

Kernstück des Frameworks stellt das *Blackboard-System* dar. Durch die hohe Heterogenität bei der Lösungsstrategie (Domäne, Hardware, Planungszustand etc.), muss ein Architekturmuster gefunden werden, welches mit nur eingeschränkt vorab beschreibbaren Lösungswegen umgehen kann – genau für diesen Einsatz eignet sich nach BUSCHMANN ET AL. (1996, S. 71 ff.) die Blackboardarchitektur.

Der *Codegenerator* wandelt das finale Modell des Montageablaufs in kompatiblen Steuerungscode. Zielsysteme können hierbei reale Steuerungen oder Simulationssysteme sein.

Die Softwarebausteine der Geschäftsschicht beziehen ihre Daten über die *Datenzugriffsschicht*, welche die Kommunikationstechnologien und Strukturen der *Datenschicht* abstrahiert. Pro technologische Ausprägung existiert ein Zugriffsobjekt, das zur Geschäftsschicht eine einheitliche Schnittstelle bereitstellt und so einen einfachen Technologiewechsel ermöglicht. Das TOMAP-Framework lädt den aktuellen Fähigkeitenkatalog aus einer Datenbank. Dieser kann bspw. als offener Standard im Internet bereitgestellt werden. Zweite Datenquelle für das Framework sind Produktinformationen, wie etwa CAD-Daten oder physikalische Merkmale, die aus der PDM-Lösung des Unternehmens gewonnen werden. Das Blackboard-System nutzt diese, um Fragestellungen bez. der Geräteauswahl beantworten zu können (etwa Traglast > Produktgewicht?).

Größtes Softwaremodul in dieser Schicht stellt der *Collection Client* dar. Dieser sammelt die aktuellen Digital-Twin-Daten von allen Geräteservern und stellt der Geschäftsschicht so eine Kopie des Anlagenzustands bereit. Dieser Ansatz fußt auf zwei Gegebenheiten: 1) Ähnlich dem zyklischen Ablauf einer SPS ist es notwendig von allen „Datenpunkten“ ein möglichst simultanes Abbild zu bekommen. Ansonsten können Inkonsistenzen in der Planung auftreten, falls sich etwa der Gerätezustand bei mehreren Zugriffen ändert. 2) Die Problemlösung im Blackboard schickt mitunter eine große

Zahl an Einzelanfragen an das Gerät, was in einem hohen Kommunikationsoverhead und damit einer niedrigen Systemperformance mündet. Eine Offline-Kopie reduziert so die Serverlast.

Die Aktualisierung des Anlagenabbilds wird jeweils vor Beginn der Planung im Blackboard-System durchgeführt oder kann jederzeit bei Bedarf ausgelöst werden. Neben Online-Geräten sammelt der Collection Client auch Offline-Digital-Twins, ohne synchronisierte Realgeräte. Hierdurch wird es möglich, physisch (noch) nicht verfügbare Geräte mit aufzunehmen, was bspw. einen virtuellen Vorabtest von Feldgeräten erlaubt (→AR8).

Abbildung 7.2 zeigt die vorgestellte Architektur. Aus Sicht der aufgabenorientierten Programmierung stellt das Modelling Framework das Aufgabenmodell (**A**) dar, das PDM-System bildet zusammen mit dem Digital Factory Twin das Ressourcenmodell (**R**) und das Blackboard-System das Planungsmodul (**P**).

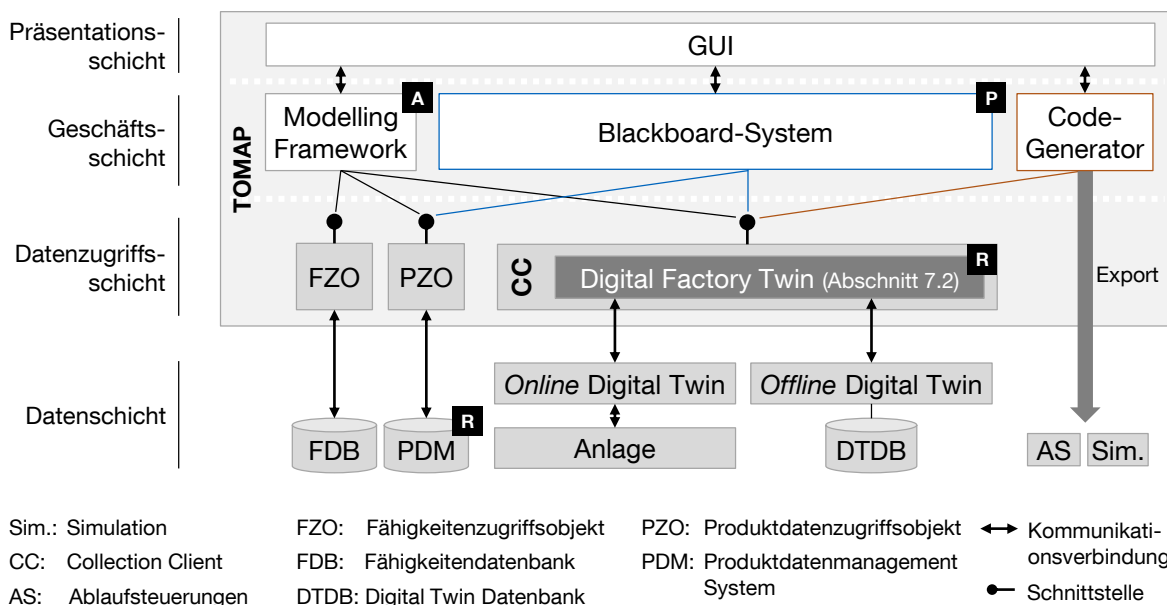


Abbildung 7.2: Architektur des TOMAP-Frameworks. Der aktuelle Fabrikzustand kann aus realen oder virtuellen Geräten bestehen. Gleiches gilt für die spätere Komponentensteuerung.

7.3.3 Modelling Framework

In Anlehnung an existierende Beschreibungsmittel aus der Prozessplanung (z. B. Montagevorranggraph oder Montageablaufplan vgl. Abschnitt 3.2.2), geschieht die Modellierung der Montageaufgabe grafisch, um so eine leichte Übertragbarkeit zu schaffen (→API). Auf einer Zeichnungsebene wird ein Montageprozess erstellt, indem Anwender (oder Programme) Knoten erzeugen und zu einem Ablauf verknüpfen, wie beispielhaft in Abbildung 7.3 zu sehen.

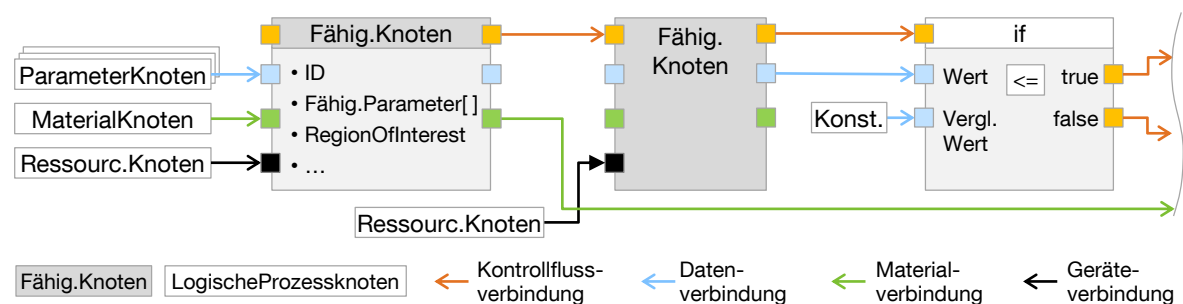


Abbildung 7.3: Modellierungsbeispiel des Montageablaufs. Fähigkeitenknoten werden über verschiedene Verbindungsarten mit ihresgleichen und logischen Prozessknoten verknüpft.

Hierbei kommen zwei unterschiedliche Knotenarten zum Einsatz: *Fähigkeitenknoten* und *logische Prozessknoten*, welche alle weiteren prozessrelevanten Informationen abbilden, wie etwa true/false-Bedingungen. Als Verbindungsmechanismus zwischen zwei Knoten wurden insgesamt vier Typen identifiziert:

1. *Ablaufverbindungen*: beschreiben den logischen Programmablauf, d. h. die Abarbeitungsreihenfolge der Prozessschritte. Über Kontrollfluss-Knoten (z. B. if-Bedingung) kann der Ablauf modifiziert werden, um so auf verschiedene Zustände reagieren zu können.
2. *Datenverbindungen*: sind notwendig, um Werte zwischen Fähigkeitsknoten zu transferieren (→AP6). Dies können sowohl vorgegebene Eingangsparameter (über Parameterknoten) sein als auch die Weitergabe des Ergebnisses an nachfolgende Knoten (z. B. Messergebnis eines Kamerasystems).
3. *Materialflussverbindungen*: Der Materialtransport kann sich unabhängig vom Programmablauf gestalten, wenn etwa logische Schleifen existieren oder mehrere Fähigkeiten an einem Ort im Montagesystem ausgeführt werden (→AP5). Über Materialknoten kann ausgewählt werden, welche Bauteile zu dem entsprechenden Prozessschritt zugeführt werden sollen.

4. *Geräteverbindungen*: geben eine Verbindung zu einem Ressourcenknoten an, der den verbundenen Prozessschritt umsetzen soll. In dem Ressourcenknoten kann der Anwender Geräte auswählen und durch das Blackboard-System auf Eignung überprüfen lassen.

Jeder Prozessknoten besitzt eine unterschiedliche Anzahl und Typen von Schnittstellen, diese Eigenschaft wird in einer generischen Schnittstellen-Klasse abgebildet, die von allen Prozessknoten genutzt wird (vgl. Abbildung 7.4).

Weiterer Bestandteil des Modells ist die Produktmodell-Klasse, welche dem Anwender Bauteilinformationen zur Verfügung stellt, vorrangig Geometriedaten zur Angabe der ROI.

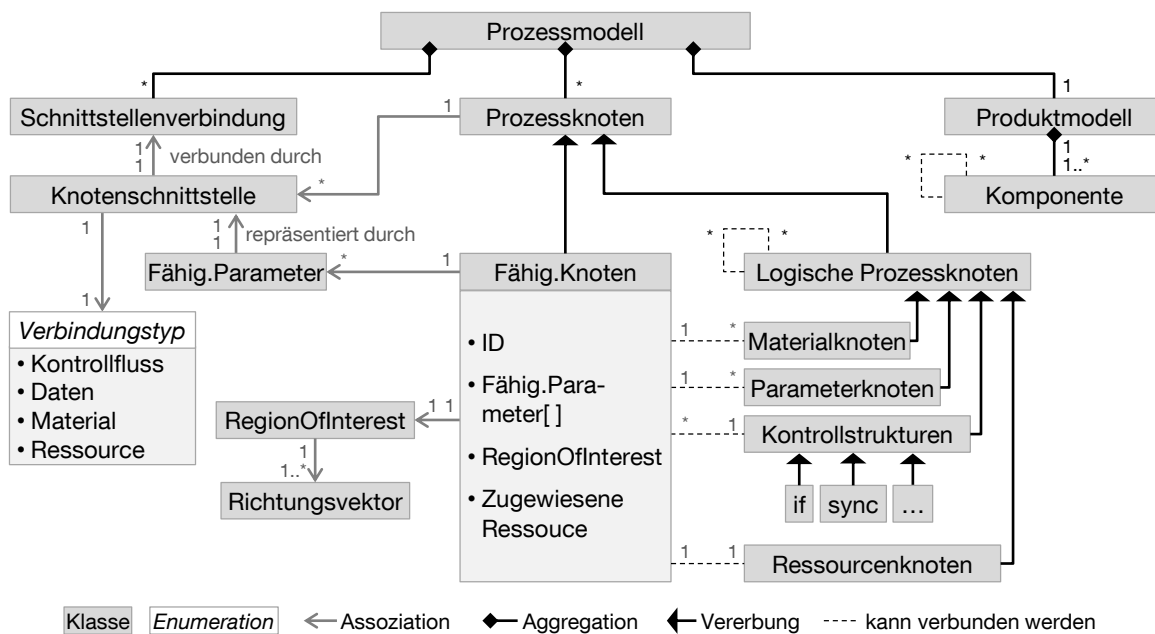


Abbildung 7.4: Architektur des Prozessmodells, dargestellt als Klassendiagramm. Aufgrund des visuellen Umfangs werden nur ausgewählte Eigenschaften gezeigt, ein detaillierteres Bild wird in Anhang A.5 gegeben.

7.3.4 Blackboard-System

Grundlagen

Das Blackboard-System stellt die Planungskomponente des Konzepts dar. Neben einer Architekturspezifikation gilt es den Problemlösungsprozess, die auftretenden Problemarten sowie die Problemlösungskomponenten zu definieren.

In Anlehnung an die Definitionen von BUSCHMANN ET AL. (1996, S. 75 f.) existieren folgende Elemente in vorliegender Architektur:

- *Blackboard*: zentraler Datenspeicher, der alle aktuellen und bisherigen Lösungen, Hypothesen und Problemstellungen enthält.
- *Experten*¹: unabhängige Softwaremodule, die spezifische Teile eines Gesamtproblems lösen.
- *Probleme*: stellen zu lösende Anfragen an das System dar und werden von Experten behandelt.
- *Hypothesen*: sind partielle oder finale Lösungen von Problemen, die noch nicht auf Gültigkeit überprüft worden sind. Hypothesen werden durch Experten aufgestellt, pro Problem können mehrere Hypothesen gelten. Abgelehnte Hypothesen werden vom Blackboard gelöscht.
- *Lösung*: ist das aus einer Menge an Hypothesen ausgewähltes Ergebnis. Die Lösungsauswahl kann durch den Anwender oder einen Blackboard-Experten geschehen.
- *Blackboard-Controller*: verwaltet die Lösungsfindung im Blackboard-System, indem gestellte Probleme an Experten weitergegeben, zurückgelieferte Hypothesen verglichen und verschiedene Sackgassen im Problemlösungsprozess erkannt und behoben werden.

Systemarchitektur und Ablauf der Problemlösung

Um der Anforderung →AR3 nach einer Problemzerlegung sowie der Erweiterbarkeit des Systems um neue Experten (→AR6) gerecht zu werden, wird die Problemlösung nicht nur zwischen verschiedenen Experten aufgeteilt, sondern auch innerhalb des Experten folgt eine Zerlegung in unterschiedliche Problemlösungsaufgaben. Immer wenn Experten externe Informationen oder Entscheidungen als Voraussetzung ihrer Hypothesenbildung benötigen, stellen sie Subprobleme an das Blackboard, die wiederum von Experten gelöst werden (→AR5). Ein Hauptproblem wird dadurch (in sich strukturell meist unterscheidende) Subprobleme zerlegt. Diese Dekomposition erlaubt die Definition kleinerer und damit stärker formalisierbarer Probleme, wodurch die

¹In der Literatur wird häufig der Begriff *Wissensquelle* verwendet. Nach Meinung des Autors klingt dies eher nach passivem Modell als nach einem aktiv entscheidenden Algorithmus, weshalb eine aktivere Bezeichnung gewählt wird.

Möglichkeiten zur automatisieren Verarbeitung steigen. Weiterhin ist mit zunehmender Dekomposition eine stärkere Parallelisierbarkeit der Problembearbeitung möglich, nach HUMBURGER (1998, S. 68) ist dies ein wesentlicher Faktor zur Reduktion der Problemlösungszeit. Ein Beispiel des sich hieraus ergebenden Problembaums für die Suche nach einem Materialflusspfad zeigt Abbildung 7.5.

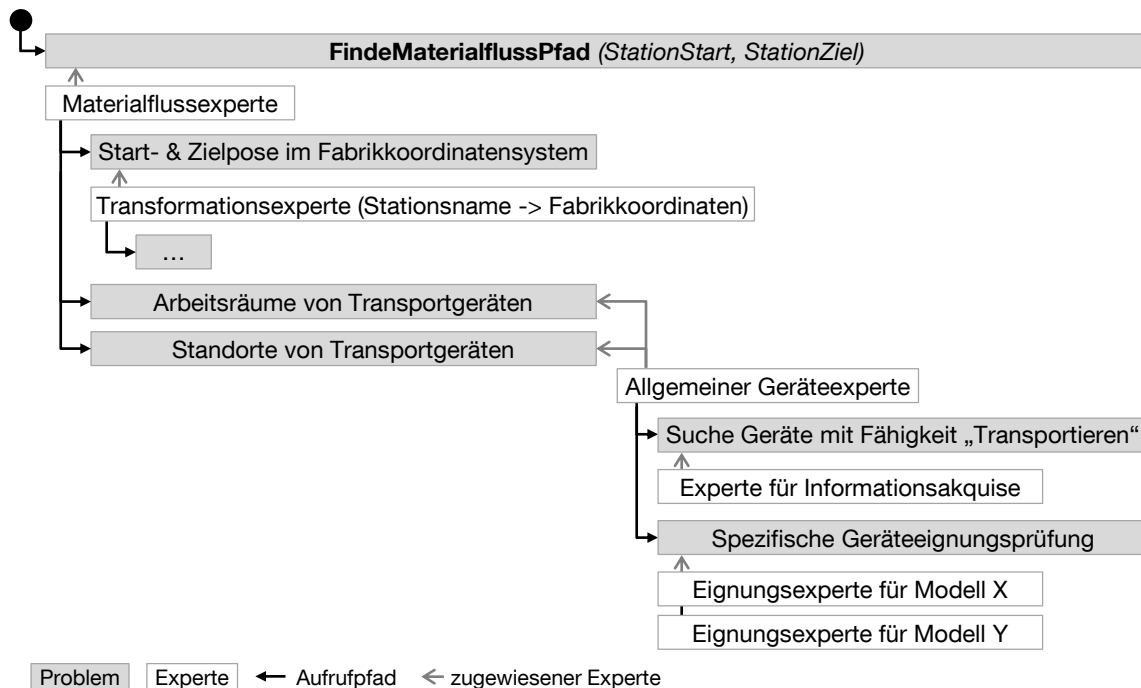


Abbildung 7.5: Beispiel eines Problembaums für die Bestimmung eines Materialflusspfads. Der Materialflussexperte erstellt eine Reihe von Subproblemen, welche am Blackboard parallel durch neue Experten bearbeitet werden.

Für eine Umsetzung des Dekompositionsansatzes wird neben den oben genannten Blackboard-Komponenten ein zusätzliches Element zur Strukturierung der Lösungsfindung eingesetzt, *BlackboardSlot* genannt (WEBER 2017, S. 61). Slots sind eigene Arbeitsbereiche für Experten, welche die aufgestellten Hypothesen und dafür notwendigen Subprobleme speichern (vgl. Abbildung 7.6). Jeder Experte bekommt für seine Problembehandlung einen eigenen Slot zugewiesen. Somit existieren für n Probleme auch n Slots, die von maximal n Experten bearbeitet werden. Slots verhindern bei einer parallelen Bearbeitung durch Experten, dass ein Merkmal am Blackboard durch mehrere Experten mehrfach und damit unkoordiniert geändert wird. Darüber hinaus ist durch die 1:1-Zuordnung von Experten zu Slots bei einem Fehlschlag der Problemlösung klar ersichtlich, welcher Experte keine Lösung finden konnte. Damit lassen sich Qualitätsmetriken definieren, was Voraussetzung für eine Auswahl konkurrierender Experten ist.

Experten besitzen einen Bedingungsteil, der beschreibt, unter welchen Bedingungen der Experte zur Problemlösung beitragen kann und einen Aktionsteil, der die eigentliche Problemlösung durchführt (BUSCHMANN ET AL. 1996, S. 76 f.). In der vorliegenden Architektur existieren hierfür zwei öffentliche Methoden: *ÜberprüfeProblem* nimmt einen spezifischen Problemtypen an und liefert zurück, ob dieser berechenbar ist (Bedingungsteil), *ErstelleHypothese* führt die Problemlösung aus (Aktionsteil).

Jedes Problem besitzt dabei einen Zustand, der anzeigt, ob es gerade bearbeitet oder bereits gelöst wurde und somit mindestens eine Hypothese zur Lösung enthält. Existieren mehrere konkurrierende Hypothesen, muss ein Experte nach einem festgelegten Zielkriterium die am besten geeignete auswählen (Abschnitt 7.3.5).

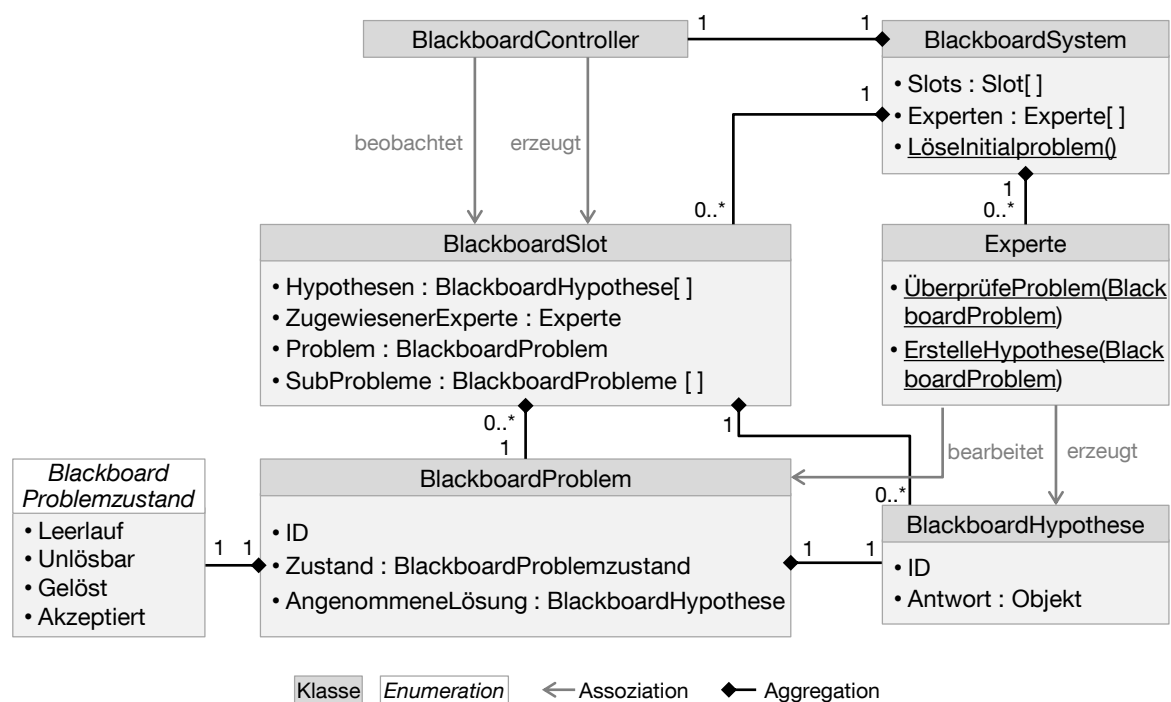


Abbildung 7.6: Klassendiagramm der Blackboard-Architektur. Erweiterte Darstellung in Anlehnung an WEBER (2017, S. 50)

Mit den vorherigen Erläuterungen ist es nun möglich, den generellen Gesamttablauf zur Problemlösung zu beschreiben, welcher in Abbildung 7.7 dargestellt ist.

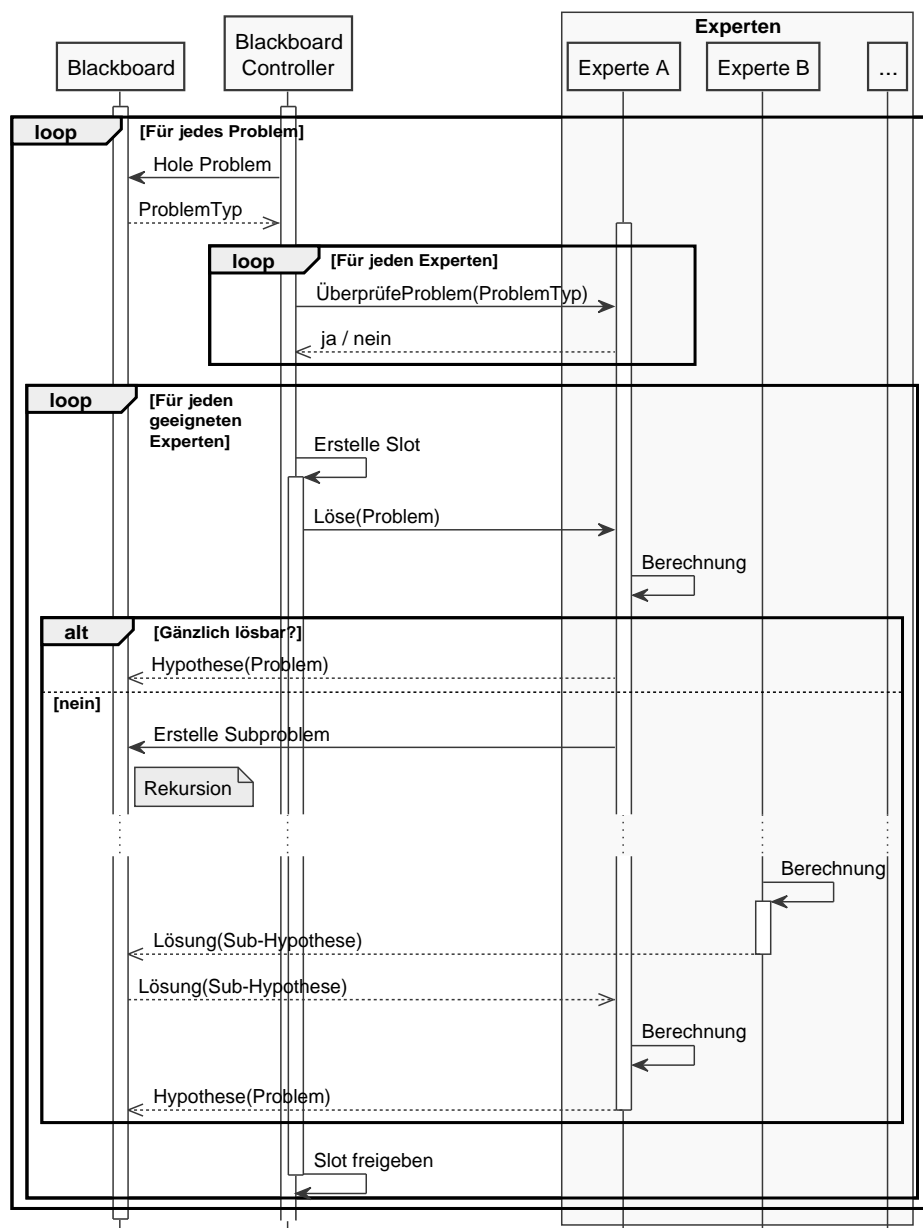


Abbildung 7.7: Genereller Ablauf der Problemlösung im Blackboard-System

Vokabular des Blackboards

Als *Vokabular* werden die Typen der im Blackboard verwendeten Elemente bezeichnet (BUSCHMANN ET AL. 1996, S. 75). Es besteht aus den Problemen und Hypothesen bzw. Lösungen. Nachdem es sich dabei um Fragestellungen und deren Antworten handelt, kann für eine strukturierte Herleitung die Sprachwissenschaft genutzt werden.

Grammatikalisch werden in der deutschen Sprache fünf Satztypen unterschieden (KÜRSCHNER 2016, S. 218 ff.), die in Tabelle 7.1 näher klassifiziert werden. Übertragen

auf Blackboard-Hypothesen (bzw. -Lösungen) kann zwischen *Aussagen* oder *Handlungsanweisungen* unterschieden werden. Blackboard-Probleme bestehen hingegen aus Fragen, die *offene Mengen* als Antwort liefern (z. B. numerische Werte) oder *geschlossene Mengen* (ja/nein/unbekannt). Dazu existiert als Sonderfall die *Wahlfrage*, bei der aus vorgegebenen Alternativen eine Auswahl getroffen wird.

Tabelle 7.1: Satztypen im allgemeinen Sprachgebrauch nach KÜRSCHNER (2016, S. 218 ff.) und deren Zuordnung zu Blackboard-Elementen

Klasse	Unterklasse	Beschreibung	Blackboard-Zuordnung	Beispiel
<i>Deklarativsatz</i>	–	Aussage	<i>Hypothese/Lösung</i>	Roboter X besitzt Traglast Y.
<i>Interrogativsatz</i>	Ergänzungsfrage	Offene Menge, gebildet aus „W-Fragewörtern“	<i>Blackboard-Problem</i>	Wie hoch ist die Traglast der Ressource X?
	Entscheidungsfrage	Geschlossene Menge (ja/nein/unbekannt)	<i>Blackboard-Problem</i>	Erfüllt Ressource X die Taktzeit Y?
	Wahlfrage	Besteht aus zwei oder mehr Entscheidungsfragen	<i>Blackboard-Problem</i>	Ist Weg A oder B kürzer?
<i>Imperativsatz</i>	–	Durchführung einer Interaktion	<i>Hypothese/Lösung</i>	Bewege Produkt von A nach B.
<i>Desiderativsatz</i>	–	Wunsch	Wird identisch zu Imperativsatz behandelt	Produkt soll von A nach B bewegt werden.
<i>Exklamativsatz</i>	–	Dringliche Aussage	Wird identisch zu Imperativsatz behandelt	Bewege Produkt von A nach B!

Fragegegenstand im Blackboard (entspricht Satzsubjekten oder -objekten) können alle Elemente des Montageprozesses sein, nach dem PPR-Modell sind dies *Produkt*, *Prozess* sowie *Ressourcen*. Letztere lassen sich weiter aufteilen in *Einzelgeräte*, die *Gesamtanlage* und die *Umgebung*.

Für eine Bestimmung der Blackboard-Problemarten erscheint die semantische Einteilung von Adverbien² zunächst nützlich. Hierbei wird in *lokale* (örtlich), *temporale* (zeitlich), *modale* (Art und Weise), *kausale* (Zusammenhänge) und *sonstige* Bestimmungsfragen unterschieden (KÜRSCHNER 2016, S. 200 ff.). Übertragen auf eine Problemstellung im Anwendungsfall zeigt sich jedoch, dass eine weitere Unterscheidung getroffen werden muss. Als Gedankenkonstrukt wird folgendes Beispiel einer lokalen Problemstellung verwendet: „Welchen Arbeitsraum besitzt Ressource X im Fabrikkoordinatensystem Y?“ – Hypothese: „Ressource X besitzt Arbeitsraum Z.“. Zur Bildung der Hypothese werden implizit (mindestens) drei weitere Subprobleme ausgelöst, die nun

²Satzglieder, die Angaben über die näheren Umstände von Ereignissen enthalten (GRANZOW-EMDEN & LUBER 2014, S. 293 f.).

ortsunabhängig sind: 1. Suche des Digital Twins der angeforderten Ressource 2. Akquise des benannten Werts (Arbeitsraum) 3. Umrechnung in das Fabrikkoordinatensystem. Das Beispiel macht ersichtlich, dass eine Problem-Unterscheidung stärker anhand der auftretenden Softwareaktivitäten gegliedert werden muss.

Generell kann zwischen *allgemeinen* und *domänenspezifischen Problemen* unterschieden werden. Allgemeine Probleme sind naturwissenschaftlicher Art und unabhängig von Anwendungsfall. Sie stellen Hilfsmittel zur Lösung domänenspezifischer Probleme dar. Beispiele sind mathematische Optimierungsverfahren, die Umrechnung von Einheiten oder die Berechnung physikalischer Größen (z. B. Schwerpunkt). Diese Problemarten sind stark formalisiert und können aus Grundlagenbüchern der Naturwissenschaften abgeleitet werden.

Für domänenspezifische Probleme werden im Betrachtungsraum Aktivitäten des Engineerings untersucht und anhand der PPR-Gebiete untergliedert. Die weitere Klassifizierung ist stark von der jeweils durchzuführenden Aktivität abhängig, bspw. „Planung der Fügereihenfolge“, „Prüfung der Geräteeignung“ etc. Für eine strukturierte Identifikation neuer Probleme bieten sich als Beschreibungsmittel Datenflussdiagramme an, in welchen die Beziehungen zwischen den auftretenden Engineeringaktivitäten abgebildet werden. Dieser Ansatz wurde bspw. von HUMBURGER (1998, S. 87 f.) erfolgreich zur Definition seiner Blackboard-Experten eingesetzt.

Eine Hybridklasse stellt die Übersetzung von Informationen zur Mensch-Maschine-Interaktion dar, wie etwa die Anzeige von Texten in einer spezifischen Sprache. Die Übersetzungsleistung kann generisch in unterschiedlichen Anwendungsfällen benutzt werden, die jeweils zu übersetzenden Elemente sind jedoch häufig fachspezifisch. Abbildung 7.8 zeigt die identifizierten Problemstellungen in der Übersicht (→AR4).

Der Problemkatalog wird als Basis genutzt, um Experten für den Referenzanwendungsfall zu entwickeln. Er stellt keine abschließende Aufzählung dar und kann mit der in Abbildung 7.9 dargestellten Methode leicht erweitert werden.

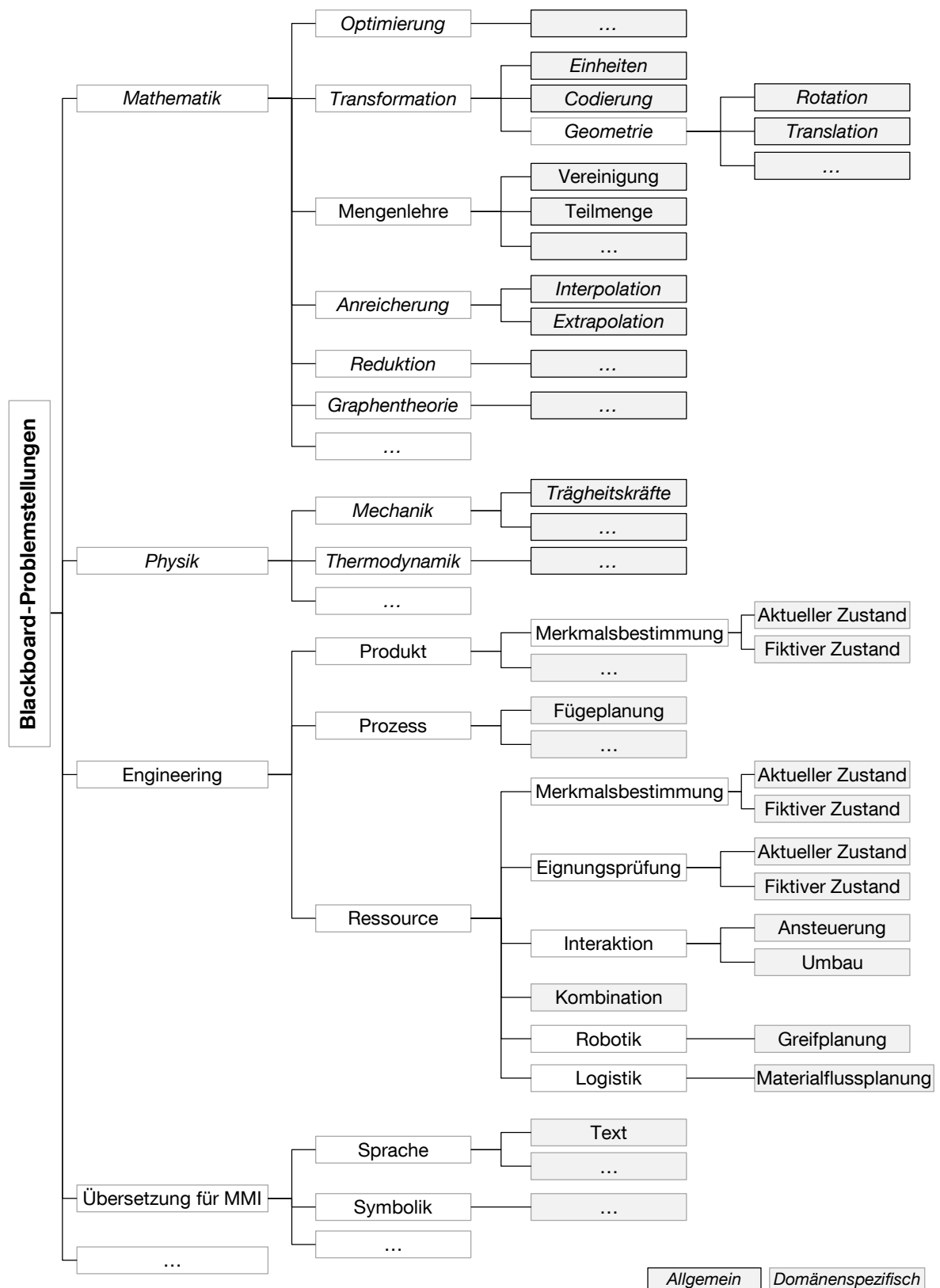


Abbildung 7.8: Identifizierte Blackboard-Problemstellungen

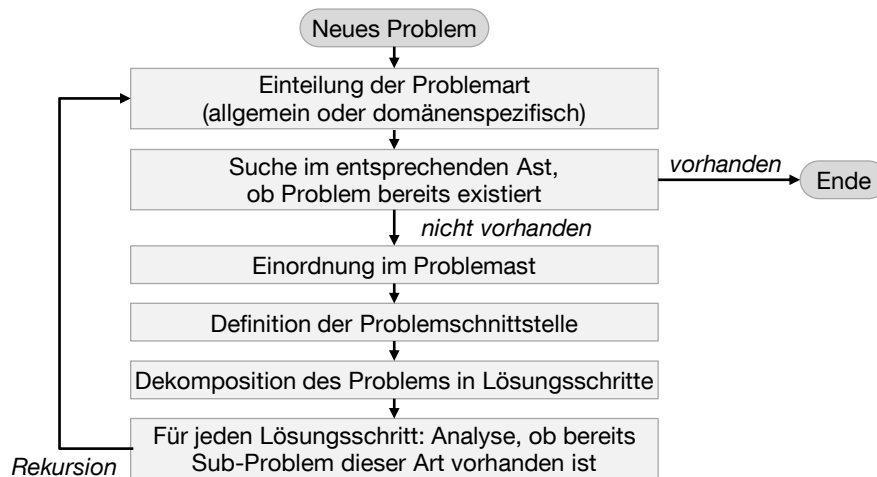


Abbildung 7.9: Methode zur Definition und Einordnung neuer Problemstellungen

Sollten ähnliche Problemstellungen in unterschiedlichen Ästen eingeordnet werden, ist dies für das Blackboard-System kein Hindernis, da Experten eine beliebige Zahl von Problemen lösen können (vgl. Abbildung 7.6, S.123). Für ähnliche Problemstellungen würde sich somit der Bedingungsteil zur Aktivierung des Experten auf das neue Problem ausdehnen – die Logik im Aktionsteil zur Problemlösung bliebe davon größtenteils unberührt.

Blackboard-Problemstellungen werden informationstechnisch als eigenständige Klassen modelliert, mit spezifischen Eigenschaften als Eingangsgröße (Grammatik) des Problems. Es wird festgelegt, dass eine Problemstellung auch die gegebene Antwortstruktur implizit mit definiert, hierdurch ergibt sich pro Problemklasse eine definierte Menge von Ein- und Ausgangseigenschaften, vgl. Abbildung 7.10.

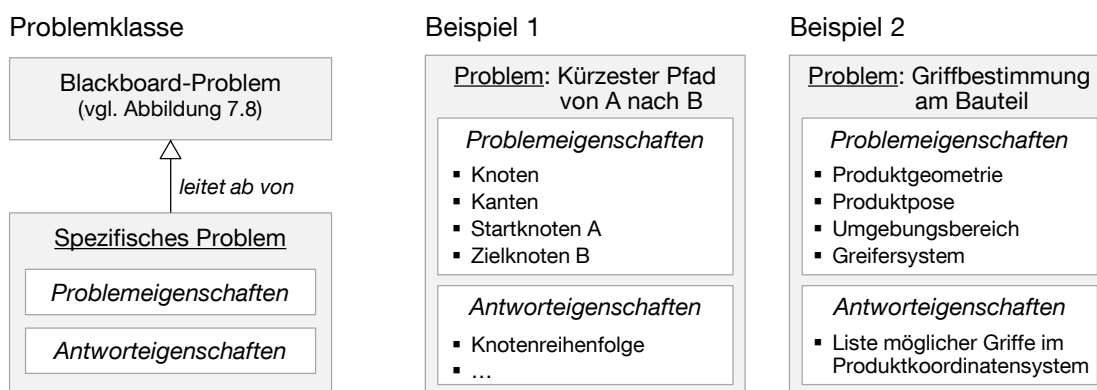


Abbildung 7.10: Aufbau der spezifischen Problemklasse

Experten

Grundsätzlich lösen Experten für sie behandelbare Problemstellungen, und stellen die Lösung als Hypothesen an das Blackboard. Für eine Klassifizierung der Experten wurden folgende Kriterien identifiziert:

Aus einer Datenverarbeitungssicht können Experten entweder neues Wissen *erzeugen* (z. B. Berechnung einer Trajektorie) oder auf Basis von vorhandenem Wissen *Entscheidungen treffen* (z. B. Auswahl des kostengünstigsten Betriebsmittels).

Dabei operieren Experten auf verschiedenen Abstraktionsebenen: *Allgemeine* Experten behandeln prozessunabhängige Probleme, z. B. Optimierungsaufgaben. *Prozessspezifische* Experten sind verantwortlich für die Ausgestaltung spezifischer Aufgaben, wie etwa die Berechnung von Handhabungsvorgängen. Sie interagieren mit Ressourcen hardwareunabhängig mittels Fähigkeiten (z. B. Bewege Produkt von A nach B). *Ressourcenspezifische* Experten stellen die bestimmteste Form dar und wurden für einzelne Hardwaremodelle oder -instanzen entworfen. Sie beantworten meist technologische Anfragen.

Experten können *lokal* aus einer Unternehmensdatenbank in das Framework geladen werden oder über den *Digital Twin* eines erkannten Geräts (vgl. Kapitel 5, etwa bei ressourcenspezifischen Experten). Als dritte Möglichkeit steht der Anschluss an eine cloudbasierte *Experten-Plattform* zur Verfügung. Dies ermöglicht Drittanbietern, Experten für häufig angefragte Probleme zu verfassen.

Als Verfasser von Experten kann das *Unternehmen* auftreten (Know-how-Schutz), der *Ressourcenhersteller* (zusätzliche digitale Dienstleistung), ein *Drittanbieter* (Geschäftsmodell) oder eine *Online-Community* im Rahmen einer Open-Source-Lösung.

Die Berechnungen im Aktionsteil des Experten können *lokal* im Framework oder remote ausgeführt werden. Dies kann innerhalb einer *Simulation* geschehen (z. B. für die Ermittlung von Kollisionen), innerhalb der *Gerätehardware* (Know-how-Schutz) oder in einer entfernten *Recheneinheit* (z. B. Cloudanbieter). Tabelle 7.2 zeigt die Kategorien in der Übersicht.

Die Definition des Funktionsumfangs von Experten orientiert sich an dem in Abbildung 7.8 vorgestellten Problembaum, da Experten spezifische Problemklassen lösen. Nachdem der Problembaum aus erforderlichen Engineeringaktivitäten besteht, muss

Tabelle 7.2: Kategorien zur Einteilungen von Blackboard-Experten

<i>Wissensverarbeitung</i>	Erzeugend	Entscheidend		
<i>Spezialisierung</i>	Allgemeiner Experte	Prozessspezifischer Experte	Ressourcenspez. Experte	
<i>Quelle</i>	Lokal (Import aus Datenbank)	Ressource (Import aus Digital Twin)	World Wide Web (Online-Server)	
<i>Verfasser</i>	Unternehmensintern	Ressourcenhersteller	Drittanbieter	Online Community
<i>Verarbeitung</i>	Lokal: Innerhalb TOMAP	Remote: Simulation	Remote: Gerätehardware	Remote: Internetdienst

für jedes auftretende Problem ein behandelnder Experte verfügbar sein, um eine Automatisierung des Engineerings zu erreichen. Hieraus folgt, dass für jedes Blatt im Problembaum ein Experte vorhanden sein muss. Experten behandeln teilweise mehrere Problemstellungen oder erstellen zur Lösung Subprobleme anderer Art, wodurch ein gegenseitiger Aufruf von Experten geschaffen wird. Hierbei existiert eine gewisse hierarchische Richtung, welche sich in vier aufgabenseitige Ebenen definieren lässt:

Auf der untersten Ebene, der *Dienstleistungsebene*, sind vorrangig Experten zu finden, welche spezifische Probleme adressieren. Hierunter fallen alle mathematischen und physikalischen Fragestellungen, die Informationsakquise sowie stark wirkprinzipabhängige Probleme, wie etwa die Greifplanung. Die Dienstleistungsebene unterstützt Experten der *Planungsebene*, die umfassende Problemstellungen aus dem Engineeringbereich lösen. Dies beläuft sich hauptsächlich auf die Einplanung fehlender Füge-, Kontroll- und Materialflussprozesse. Weiterhin behandeln Experten in dieser Ebene den statischen und dynamischen Zusammenschluss von Ressourcen, wie etwa die Kombination eines Handhabungsgeräts mit einem Endeffektor.

Experten der *Kontrollebene* überwachen den Planungsprozess, in dem bestehende Abläufe optimiert werden. Dies kann etwa die Optimierung erstellter Prozesse im Montageablauf sein, z. B. Minimierung der Anzahl an Greiferwechseln etc. In diese Ebene fallen auch Experten zur Bewertung konkurrierender Hypothesen der unterlagerten Ebenen.

Auf höchster Ebene steht der *Blackboard-Controller*, welcher die Ausführung aller Experten und den Fortschritt der gesamten Problemlösung überwacht. In der Klassifizierung

nach HUMBURGER (1998, S. 116 ff.) stellt er damit einen Experten der *Strategieebene* dar.

Abbildung 7.11 zeigt zu den vorgestellten Ebenen relevante Experten in der Montage.

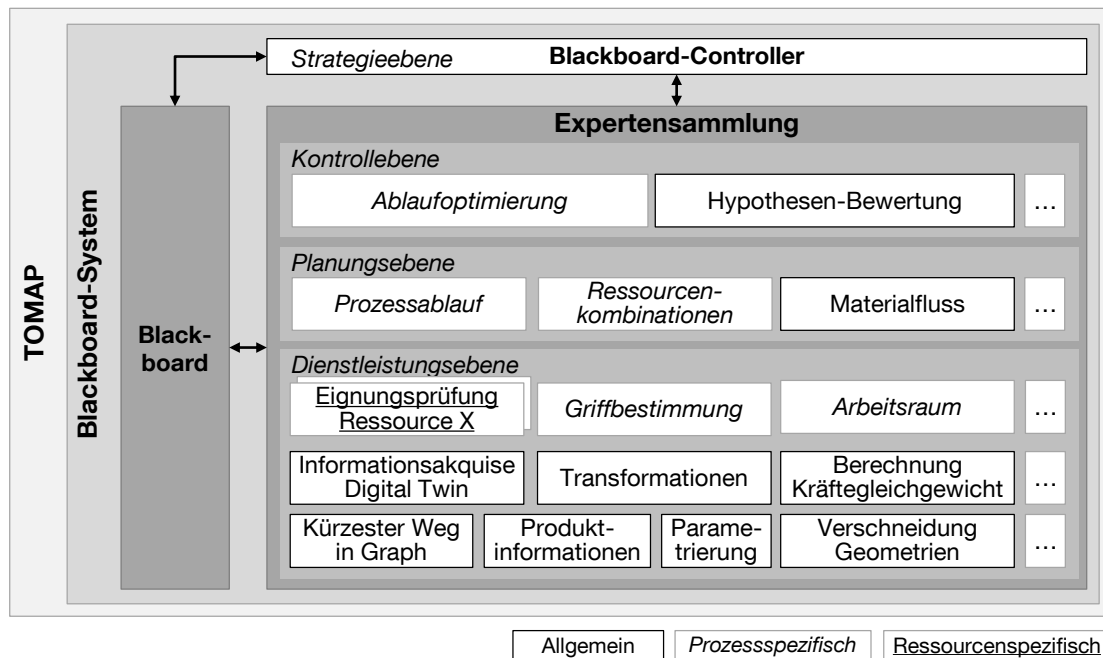


Abbildung 7.11: Übersicht relevanter Experten in der Montagetechnik und ihre Einordnung in vier Abstraktionsebenen.

Auch für den Menschen können stellvertretende Expertenmodule geschrieben werden, wodurch er mit seinem Fachwissen in die Problemlösung eingebunden werden kann (→AR7). Die Expertenmodule leiten hierbei im Aktionsteil die Anfragen an das GUI des TOMAP-Frameworks weiter. Damit entsteht eine Fall-Back-Strategie, die auch zu einer Lösung führen kann, wenn kein geeigneter Software-Experte existiert.

7.3.5 Beschreibung ausgewählter Experten für die Montagetechnik

Nachfolgend werden ausgewählte Experten zur Problemlösung im Betrachtungsraum des Referenzanwendungsfalls beschrieben. Probleme der Mathematik- und Physikdomäne stehen nicht im Fokus, da das Wissen zur Problemdefinition und -lösung durch eine Vielzahl an Literatur bereits beschrieben ist. In den nachfolgenden Abschnitten wird deshalb auf die Beschreibung von Experten der Engineeringdomäne eingegangen.

Strategieebene: Blackboard-Controller

Der Blackboard-Controller muss sicherstellen, dass die Problemlösung zu einem Ziel führt und einzelne fehlgeschlagene Experten nicht die gesamte Problemlösung bedrohen.

Dabei erfolgt die Problemlösung parallelisiert, um die erforderliche Bearbeitungszeit zu verkürzen (HUMBURGER 1998, S. 85 f.). Existieren für ein Problem mehrere Experten, wird jeder ihnen zugewiesene Slot in einem eigenen Softwareprozess (Thread) ausgeführt, um so trotz evtl. fehlerhafter Problemzweige einer Gesamtlösung fortfahren zu können. Dabei muss auf zwei Charakteristika eingegangen werden: Festgefahrene (Deadlocks) oder sich unendliche wiederholende (Lifelocks) Problemlösungen.

Deadlocks entstehen, wenn Experten in der Ausführung ihres Aktionsteils blockiert sind, entweder durch interne Programmfehler oder weil sie auf die nicht eintreffende Lösung eines Subproblems warten. Zur Behebung wird eingeführt, dass jeder Experte seine ungefähre Ausführzeit dem Blackboard-Controller mitteilt. Während der Problemlösung überwacht ein Softwaremodul die angegebene Ausführungszeit und markiert den Lösungsprozess als fehlgeschlagen, falls dieser seine Zeit überschreitet (WEBER 2017, S. 56). Hiervon ausgenommen muss die Zeit zur Lösung der Subprobleme sein, da diese vorab nicht geschätzt werden kann. Die Zeitmessung wird somit bei einem Warten auf Sublösungen unterbrochen. Aufgrund des rekursiven Ansatzes besitzen die ausführenden Subexperten wiederum eigene Timer. Die angegebenen Zeiten müssen dabei nicht starr sein, sondern können bei jeder Problemlösung kontinuierlich durch das System evaluiert und aktualisiert werden.

Bei *Lifelocks* produziert eine Reihe von Experten Hypothesen, welche erneut das zu lösende Problem enthalten, wodurch eine Art Endlosschleife entsteht. Dies führt zu einer stetig wachsenden Problembaumtiefe, jedoch zu keiner Lösung. Lifelocks sind schwieriger zu bestimmen als Deadlocks, da es valide ist, wenn Experten dasselbe Problem mehrmals behandeln: Agiert der Experte nicht deterministisch oder sind Randbedingungen zum initialen Problem unterschiedlich, kann der erneute Aufruf durchaus eine Hypothese generieren. Zur Lösung von Lifelocks sind zwei Ansätze denkbar (WEBER 2017, S. 58): 1.) Für jeden Expertenaufruf wird der Problembaum auf sich wiederholende Konstellationen durchsucht und ggf. abgebrochen. Dies setzt jedoch die vollständige Beobachtbarkeit und Definition des Problembaums sowie deterministisches Verhalten der Experten voraus. 2.) Es wird die Aufruftiefe im Problembaum beobachtet. Ist

diese zu groß, wird von einem Lifelock ausgegangen und der jeweilige Problemzweig abgebrochen. Letzterer Ansatz benötigt wesentlich weniger Rechenressourcen und Modellierungsaufwände, kann jedoch bei sehr langwierigen Problemzweigen False-Positives erzeugen.

Existiert kein softwareseitiger Experte zur Lösungsfindung, so wird das Problem an den „Mensch-Experten“ als universelle Problemlösungsinstanz weiterdelegiert und von diesem bearbeitet. Durch die Einteilung der Problemklassen ist hierbei eine Spezialisierung der Anwenderrollen denkbar, Nutzer könnten bspw. als Füge- oder Materialflussexperten auftreten.

Kontrollebene: Ablaufoptimierung

Bei mehreren Geräte- oder Materialflussalternativen ist es notwendig, eine Optimierung des zugewiesenen Montageablaufplans durchzuführen, um so eine optimale Ausführreihenfolge und Hardwarekombination zu erreichen. Eingangsparmeter für die Ablaufoptimierung sind somit die finale Montagefähigkeitensequenz und geeignete Ressourcen(-alternativen).

Eine Lösungsmöglichkeit stellt die Gesamtkostenberechnung der Ausführung auf Basis von Maschinenstundensätzen dar. Dies gilt unter der Annahme, dass die vom Anwender innerhalb spezifizierten Qualitätsanforderungen durch geeignete Ressourcen gänzlich erfüllt werden. Über ressourcenspezifische Experten müssen zuerst die jeweiligen Ausführzeiten bestimmt werden, die anschließend ein allgemeiner Kostenexperte weiterverrechnet. Der Experte benötigt als Eingangsparmeter die aktuellen Rohstoffpreise sowie die energetischen und stofflichen Ressourcenverbräuche, welche aus dem Digital Twin gewonnen werden (Abschnitt 7.2).

Anhand der Fähigkeitensequenz kann ein gerichteter Stationsgraph gebildet werden, dessen Knoten Ausführungskosten der prozessnahen Ressourcen enthalten und Kanten Transportkosten zwischen diesen widerspiegeln. Ressourcenalternativen können mit mehreren Knoten im Graph dargestellt werden (R_3 und R_4), verschiedene Materialflussalternativen über Mehrfachkanten ($R_5 \rightarrow R_6$, vgl. Abbildung 7.12, links). Die unterschiedlichen Kostenarten können im Anschluss zu kombinierten Kantenkosten zusammengefasst werden (Abbildung 7.12, rechts). Der sich so ergebene gerichtete Multigraph kann nun mit einem modifizierten Dijkstra-Algorithmus (vgl. BISWAS ET AL. 2013) minimiert werden, um so die kostengünstigste Anlagenkonfiguration zu finden.

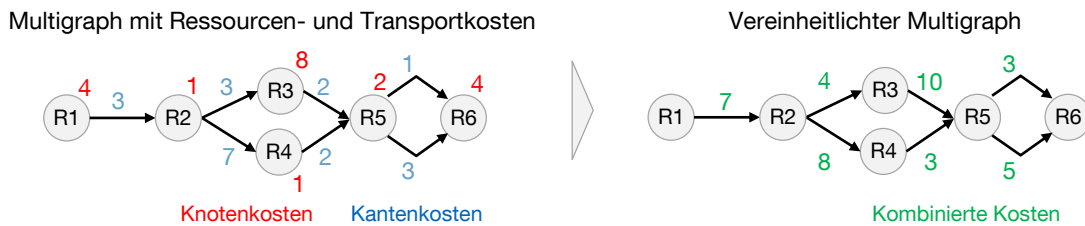


Abbildung 7.12: Darstellung eines beispielhaften Montageablaufplans als gerichteten Multigraph

Kontrollebene: Hypothesen-Bewertung

Erzeugte Hypothesen müssen als Lösung angenommen werden, um mit der Problemlösung fortfahren zu können. Hierfür werden spezifische Experten geschrieben, welche Hypothesen auf Plausibilität prüfen.

Um die Expertensammlung möglichst gering halten zu können, wird definiert, dass jeder Experte vor Veröffentlichung zuerst intern seine Hypothese auf Plausibilität kontrollieren muss (soweit möglich). Beispielsweise muss eine erzeugte Trajektorie innerhalb des Gerätearbeitsbereichs liegen. Existiert nur eine Hypothese, so wird diese automatisch als Lösung des jeweiligen Problems angenommen.

Existieren konkurrierende Hypothesen, die etwa durch unterschiedliche Experten aufgestellt wurden, so muss eine Auswahl geschehen. Hierbei wurden vier Lösungsansätze identifiziert: *Primitive Auswahlverfahren* nehmen die erste eingegangene Hypothese an oder wählen aus allen eingegangenen ein Element zufällig aus. Sie sind leicht zu implementieren und liefern die schnellste Problemlösung.

Alternativ können die *Qualität von Hypothesen* oder *Experten* bewertet werden. Da eine Formalisierung und Berechnung von entsprechenden Metriken (z. B. Gültigkeit der Aussage) komplex ist, bieten sich heuristische Verfahren an, die das Gesamtergebnis auf Basis eines zufällig ausgewählten Subsets von Hypothesen/Experten iterativ bewerten (vgl. GERDES ET AL. 2004).

Planungsebene: Prozessablauf

Der Softwareexperte zur Planung des Prozessablaufs erstellt und verfeinert den vom Nutzer vorgegebenen Montageprozess. Hierzu nutzt er bspw. Assembly-by-Disassembly-Ansätze, um Fügereihenfolgen aus Produktdaten zu bestimmen (vgl. Abschnitt 3.2.2).

Falls keine geeigneten Betriebsmittel gefunden werden, zerteilt der Experte zusammengesetzte Fähigkeiten in Elementarfähigkeiten, anhand des in Abschnitt 6.2.4 beschriebenen Schablonenkonzepts. Diese können in einer Datenbank hinterlegt oder durch einen Experten behandelt werden, welcher für die Dekomposition der jeweiligen zusammengesetzten Fähigkeit verantwortlich ist.

Treten in der Prozesssequenz redundante, unvollständige oder fehlerhaft modellierte Prozesse auf, so besteht die Aufgabe des Experten darin, diese Fehler zu beheben. Die Ausführung kann bspw. als Assistenzfunktion für den Anwender geschehen, indem Fehler im Modelling Framework (Abschnitt 7.3.3) grafisch hervorgehoben werden.

Planungsebene: Kombinierte Ressourcen

Neben Einzelressourcen sind Montageanlagen aus einer Vielzahl an kombinierbaren Standardgeräten aufgebaut. Eine Verbindung kann *statisch* bereits während der Gerätevernetzung existieren oder *dynamisch* durch das Softwareframework erzeugt werden.

Statische Kombinationen werden durch die Plug&Produce-Software identifiziert (vgl. Abschnitt 5.2.3) und als Gerätehierarchie im Informationsmodell angezeigt (Abschnitt 5.3).

Für *dynamische* Kombinationen wird jede verfügbare Ressource im Digital Factory Twin auf dessen Eingangsschnittstellen überprüft. Diese bestehen bspw. aus mechanischen oder elektrischen Anschlüssen (vgl. Abschnitt 7.2). Ist eine Schnittstelle nicht belegt, wird nach dem kompatiblen Gegenpart innerhalb des Digital Factory Twins gesucht. Für alle möglichen Kombinationen werden „fiktive“ Geräteverbünde angelegt, welche von anderen Softwaremodulen genutzt werden können. Für jede verbundene Komponente wird ein Kombinationsproblem ausgelöst, auf Basis dessen bei Bedarf eine Merkmalsänderung innerhalb der Ressource berechnet werden kann (z. B. die Reduktion der Tragfähigkeit eines Industrieroboters). Das Problem wird durch ressourcenspezifische Experten bearbeitet. Im Rahmen von Subproblemen der „Merkmalsbestimmung“ fordern diese meist physikalische Größen an, wie etwa das Endeffektorgewicht.

Wird solch eine virtuelle Kombination letztlich zur Durchführung der Fähigkeit ausgewählt, erzeugt der Experte ein Interaktionsproblem, das den Ressourcenumbau enthält. Dieser kann manuell oder automatisiert (z. B. Roboter mit Greiferwechselsystem) ausgeführt werden.

Die Eignungsprüfung von Gerätekombinationen wird innerhalb der ressourcenspezifischen Experten behandelt (vgl. nachfolgender Abschnitt).

Dienstleistungsebene: Eignungsprüfung Ressourcen

Für die Bestimmung einer *prozessseitigen* Geräteeignung fließen hardwarespezifische Informationen in die Entscheidungsfindung ein. Da dies meist internes Know-how des Komponentenherstellers darstellt, wird davon ausgegangen, dass dieses Wissen nicht frei verfügbar ist. Demnach wird für die meisten Gerätequalifizierungen ein ressourcenspezifischer Experte befragt, dessen Verarbeitung remote gekapselt stattfindet (vgl. Tabelle 7.2).

Der Geräteexperte behandelt im Rahmen eines *Eignungsprüfungsproblems* die Frage, ob der geforderte Prozessschritt PS mit der Menge an Eingangsparametern $\{p_1; \dots; p_n\} =: P_E$ durch die Ressource erfüllt werden kann und stellt das Berechnungsergebnis $\{p'_1; \dots; p'_n\} =: P'_E$ als Angebot der Fähigkeit F bereit ($\rightarrow AR2$). Liegt P'_E vollständig innerhalb von P_E , ist die Ressource geeignet.

Während der Eignungsberechnung können mehrere Fälle auftreten:

- *Fehlende Informationen*: Der Experte erstellt in diesem Fall ein Subproblem. Bei Fragestellungen welche analytisch/simulativ nur schwer beschreibbar sind (etwa der Reflektionsgrad eines metallischen Bauteils) kann bspw. der Parametrierungsexperte einen Teach-in-Vorgang zur Wertakquise durchführen (vgl. nachfolgende Abschnitte).
- *Eingangsparameter $p_n \in P_E$ ist unabhängig von anderen Parametern*: Die Berechnung kann im Experten einfach ausgeführt werden. Ein Beispiel hierfür ist die Ausführungszeit eines primitiven Sensors.
- *Mehrere Eingangsparameter $\{p_i; \dots; p_j\} \subseteq P_E$ sind voneinander abhängig*: Nachdem Anwender stets einen Wertebereich $W_{p_n} := [\text{Min}_{p_n}, \text{Max}_{p_n}]$ pro Parameter vorgeben (vgl. Abschnitt 6.2.5), entsteht hierbei ein multikriterielles Optimierungsproblem, welches mit bekannten Verfahren aus der Mathematik gelöst werden kann (vgl. PAPPAGEORGIOU ET AL. 2015). Als vereinfachend erweist sich der Umstand, dass die Zielfunktionen der erforderlichen Ressourcengenauigkeit $f_{gen.}(p_{gen.})$ sowie die Ausführzeit $f_t(p_t)$ stets minimiert werden müssen, um keine unnötigen Betriebsmittelkosten entstehen zu lassen.

Das Ergebnis P'_E stellt stets eine Menge von Skalaren dar, die als Hypothese zurückgeliefert und durch einen Experten mit den geforderten Größen verglichen werden. Die letztliche Annahme des Ergebnisses kann automatisiert gelöst werden oder in der Verantwortung des Anwenders liegen.

Neben Einzelressourcen können auch Gerätekombinationen auftreten, deren Verhalten durch die momentanen Schnittstellengrößen (z. B. Spannung, Druckluft) bestimmt wird. Demnach muss der Ressourcenexperte auch Problemstellungen L behandeln, welche neben den Fähigkeitenparametern P_E auch von den Werten der Eingangsschnittstellen S_E abhängig sind: $L = f(P_E, S_E, \dots)$

Eine Kombination kann hierbei eine wechselseitige Merkmalsänderung auslösen (*beeinflussende Kombination*) oder *autark* pro Komponente stattfinden.

Der Informationsfluss einer statischen, *autarken* Kombination wird am Beispiel eines Förderbands erläutert, das über einen Motor, Getriebe und Frequenzumrichter an eine Steuerung angeschlossen ist (Abbildung 7.13). Hierbei stellt das Förderband die Fähigkeit Transportieren bereit.

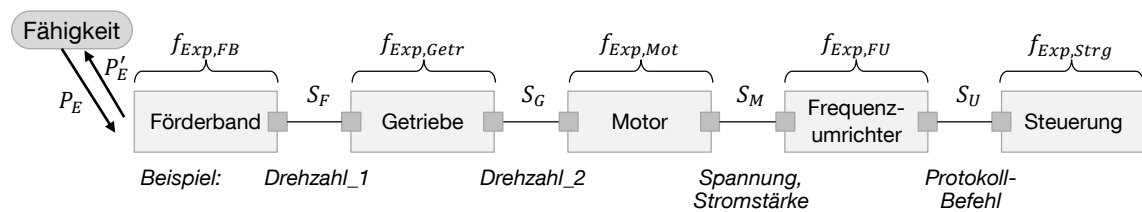


Abbildung 7.13: Schnittstellenverkettung einer beispielhaften autarken Gerätekombination

Im Rahmen der Eignungsüberprüfung wird der ressourcenspezifische Experte des Förderbands aufgerufen. Der Experte kann diesbezüglich jedoch nur eine Aussage treffen, wenn die entsprechenden Eingangsschnittstellenwerte (hier: S_F) bekannt sind. Hierzu wandelt er P_E in für das Gerät zulässige Wertebereiche von S_F um. Ähnlich zu der vorherigen Fähigkeitenanfrage wird anschließend ein Subproblem erzeugt, ob der verbundene Partner die geforderten Schnittstellen-Wertebereiche anbieten kann. Dies wird pro Teilnehmer als Problemkaskade so lange fortgesetzt, bis die Schnittstellenanfrage die letzte Komponente erreicht (Steuerung). Der ursprüngliche Wertebereich kann dabei aufgrund der Komponentenlimitierungen von jedem Experten sukzessive eingeschränkt werden. Es ergibt sich folgende Informationsverkettung:

$$S_U = f_{\text{Exp,FU}}(S_M) \circ f_{\text{Exp,Mot}}(S_G) \circ f_{\text{Exp,Getr}}(S_F) \circ f_{\text{Exp,FB}}(S_E) \quad (7.1)$$

Die Steuerung kann schließlich bestimmen, ob sie den geforderten Wertebereich S_U absenden kann, woraufhin die stufenweise Lösung der Problemkaskade geschieht, bis im ursprünglichen Experten der durch die Vorgänger mögliche Schnittstellenbereich S_F verfügbar ist und schließlich die Aussage P'_E getroffen werden kann (\rightarrow AR9).

Da die Fähigkeit nur logisch dem Förderband zugeordnet ist, informationstechnisch aber im Proxy-Server der Steuerung läuft, erfolgt die Ansteuerung als inverse Kaskade, dies ist in Abschnitt 5.2.3 erläutert.

Für den Anwendungsfall einer dynamischen *beeinflussenden* Kombination wird die zusammengesetzte Fähigkeit „Pick-and-Place“ als Beispiel verwendet. Hierfür sucht der Prozess-Experte einen Verbund, der Sichern und Bewegen anbietet, und bekommt bspw. eine „Greifer + Roboter“-Kombination vorgeschlagen. Im Rahmen der Eignungsprüfung des Greifers muss bspw. die erforderliche Haltekraft berechnet werden, hierzu ist aufgrund einer möglichen Bewegung des Geräts u. a. die Bahnbeschleunigung des Handhabungssystems und die Masse m des Produkts notwendig:

$$P'_{E,greifen} = f(P_{E,greifen}, P_{E,Roboter}, m_{Produkt}, \dots) \quad (7.2)$$

Wie im autarken Fall wird die Anfrage als kaskadierende Subproblem gestellt, in diesem Fall entlang des mechanischen Verbindungspfads (vgl. Abbildung 7.14). Für die Akquise der Bahnbeschleunigung wird bei dem Robotersystem angefragt, welches hierfür zuerst die Gesamtmasse des Endeffektors benötigt. Anhand der mechanischen Verbindung wird zuerst eine Anfrage an den Greifer gestellt und anschließend an das Produkt. Die Trajektorie kann im Falle einer bereits spezifizierten Bewegens-Fähigkeit direkt berechnet werden, falls keine Spezifikation vorhanden ist, werden die Bahnparameter als Subproblem abgefragt. Nach Erhalt der Bahnbeschleunigung fragt der Greifer-Experte die Masse des Produkts ab – dieses Mal mit der Zielstellung der Eignungsberechnung (z. B. Festigkeit).

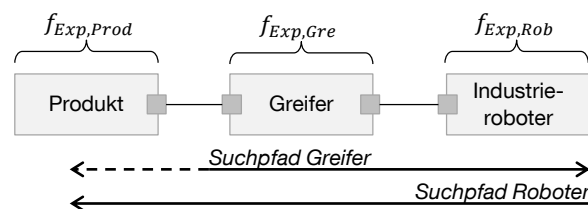


Abbildung 7.14: Schnittstellenverkettung einer beispielhaften beeinflussenden Gerätekombination

Neben einem Abgleich der prozesseitigen Parameter muss auch die *geometrische* Domäne auf Eignung überprüft werden. Diese Aufgabe wird als Subproblem an den *Arbeitsraumexperten* delegiert, der im folgenden Abschnitt vorgestellt wird.

Dienstleistungsebene: Arbeitsraum

Der Arbeitsraumexperte stellt die geometrische Erreichbarkeit einer Ressource zur angegebenen ROI auf dem Produkt sicher. Als Problemparameter benötigt er den Arbeitsraum und Pose der Ressource, sowie die aktuelle Produktgeometrie und angegebene ROI der Fähigkeit. Über geometrische Verschneidungsverfahren wird berechnet, ob a) das Produkt innerhalb des Arbeitsraums liegt und b) das Gerät freie Zugänglichkeit zur ROI hat. Analog zur ROI sind im Arbeitsraum ein oder mehrere Richtungsvektoren definiert, die mit dem Aktionsvektor der ROI abgeglichen werden. Bei Aktoren geben Arbeitsraumvektoren mögliche Orientierungen der Ressource an einem Punkt an. Dieser Ansatz wird bspw. von ZACHARIAS (2012) verwendet, um Roboterposen zu beschreiben. Bei Sensoren wird hierüber der Messfehler innerhalb des Arbeitsraums beschrieben, so weisen bspw. optische Sensoren an den Sichtfeldrändern wesentlich höhere Ungenauigkeiten auf (Verzeichnung) und dürfen somit weniger von der idealen Orientierung abweichen.

Lassen sich im Bereich der ROI Arbeitsraumvektoren finden, die parallel zu dem ROI-Aktionsvektor stehen, so erfüllt die Ressource die geometrischen Anforderungen der Fähigkeit, die Einhaltung von a) und b) vorausgesetzt. Für eine Lösung kann der Experte die Geometrieverschneidung als Subproblem stellen, welches bspw. durch einen Simulationsexperten übernommen wird (vgl. Tabelle 7.2).

Da Arbeitsraum und ROI zwei Mengen darstellen, wird als Ergebnis eine Menge an Produktposen M_{Produkt} bezogen auf den Arbeitsraum zurückgeliefert. Diese können vom Materialflussexperten genutzt werden, um erreichbare Transportressourcen zu finden.

Planungsebene: Materialfluss

Der Materialflussexperte (MFE) erstellt Lösungen zum Produkttransport in der Montageanlage (\rightarrow AP2). Dies ist in drei Phasen gegliedert. Phase I sucht geeignete Materialflussposen:

- I-1. *Suche nach Haltekombinationen:* Als Voraussetzung für eine Ausführung von Fähigkeiten wird definiert, dass das Produkt in einer eindeutigen Position gehalten werden muss, d. h., alle rotatorischen und translatorischen Freiheitsgrade (engl. Degrees of Freedom, DOF) fixiert sind. Ein solcher Ort wird als „Product Shape

Pose“ (PSP) bezeichnet, welche die eindeutige Raumpose einer Produktgeometrie beschreibt. Der MFE sucht nach Ressourcen, welche eine PSP bereitstellen können. Ausgangsgröße stellt die Menge an berechneten, möglichen Produktposen im Arbeitsraum zugewiesener Geräte dar ($M_{Pos,Produkt}$, vgl. Arbeitsraumexperte). Diese wird mit dem Arbeitsraum von Ressourcen verschnitten, welche eine Sichern-Fähigkeit anbieten und eine kompatible mechanische Schnittstelle zum Produkt aufweisen. Für jede Komponente wird eine Eignungsprüfung als Subproblem gestellt. Ergebnis ist eine Menge an Raumposen $M_{PSP,v1}$, an denen das Produkt sowohl gehalten als auch von der jeweiligen prozessnahen Ressource „bearbeitet“ werden kann: $M_{PSP,v1} := M_{Pos,Produkt} \cap M_{Pos,Halten}$

Hierbei wird für jede Ressource überprüft, wie viele DOFs sie einschränkt. Ist diese Zahl kleiner als 12 (sechs Freiheitsgrade in positiver und negativer Richtung), wird nach unterstützenden Komponenten gesucht, welche die restlichen DOFs einschränken. Dies gilt sowohl für die Ausgangsschnittstelle (zum Produkt) als auch die Eingangsschnittstelle (Verbindung zur Umgebung bzw. zu anderen Geräten).

- I-2. *Suche nach Transportkombinationen:* Der MFE sucht nach Ressourcen mit Transport-Fähigkeit. Anschließend wird die prozesseitige und geometrische Eignung der Geräte überprüft. Ergebnis ist die Menge geeigneter Ressourcen $M_{Transport}$.
- I-3. *Erzeugung erreichbarer PSPs:* Arbeitsräume geeigneter Transportgeräte werden mit $M_{PSP,v1}$ verschnitten und ergeben so eine Menge an PSPs, die zusätzlich durch einen Materialflusspfad erreicht werden können: $M_{PSP,v2} := M_{PSP,v1} \cap M_{Transport}$

In Phase II wird der Materialflussgraph im Montagesystem erstellt:

- II-1. *Erzeugung ungerichteter Materialflussgraph:* Die Arbeitsräume aller geeigneten Transportressourcen von $M_{Transport}$ werden gegenseitig verschnitten, um mögliche Übergabebereiche und damit Knoten im Materialflussgraphen zu erzeugen. Dazu werden die transportseitig erreichbaren Prozessposen $M_{PSP,v2}$ eingefügt, wodurch ein ressourcenübergreifendes Transportnetz entsteht. Kanten stellen die jeweiligen Transportmittel dar. Da mitunter mehrere Ressourcen eine Start- und Endpose im Raum erreichen können, existieren parallele Kanten in einem Multigraph.
- II-2. *Erzeugung gerichteter Materialflussgraph:* Für jede Ressource wird geprüft, ob sie einen uni- oder bidirektionalen Materialfluss bewirken kann oder sonstigen Beschränkungen (z. B. Stellplätze) unterliegt. Dies wird als Richtungs- und semaphorische Information in den Graphen mit aufgenommen.

Die Auswahl der Transportwege und -mittel wird in Phase III realisiert:

- III-1. *Berechnung Transportwege*: Auf Basis des modellierten Montageablaufs und der zugewiesenen Geräte kann eine Reihenfolgebildung im Materialflussgraph geschehen. Es werden alle möglichen Materialflusspfade gesammelt und ihre zugehörigen Transportgrößen (z. B. Zeitdauer und Kosten) aus dem Digital Twin gelesen und summiert.
- III-2. *Präsentation Transportwege*: Dem Anwender werden die unterschiedlichen Transportmöglichkeiten (Pfade, respektive Transportmittel) präsentiert (→AP4). Über die gegebenen Metriken kann dieser eine Auswahl innerhalb der Menge an geeigneten Geräten treffen (→AP3). Alternativ kann über Optimierungsverfahren automatisiert nach einer oder mehreren Größen optimiert werden (vgl. Abschnitt „Ablaufoptimierung“, S. 7.3.5).
- III-3. *Einfügen Materialflussschritte*: Da jede Ressourcenart eine andere Abfolge von Transportschritten benötigt (vgl. Industrieroboter und Förderband), werden aus jedem Digital Twin die dem Prozess vor- und nachgelagerten Transportschritte (Halten, Bewegen, Anwesenheit prüfen, Position messen etc.) geladen und im Montageablauf entsprechend eingefügt. Sind Fähigkeiten noch nicht durch Ressourcen besetzt, wie etwa Anwesenheit prüfen, müssen für diese Komponenten analog dem initialen Vorgehen Geräte gefunden werden. Hierdurch ergibt sich eine Iteration, bis alle Fähigkeiten eine Umsetzung auf Ressourcenseite erlangen.

Dienstleistungsebene: Parametrierung

Die angegebenen Fähigkeitenparameter sind unter Umständen nicht ausreichend für eine Befehlsausführung innerhalb der Ressource, etwa wenn weitere hardwarespezifische Parameter notwendig sind. Der Ressourcenexperte kann fehlende Parameter über Subprobleme anfordern. Benötigt eine Ressource bspw. Materialkennwerte, so ist über die ROI bekannt, an welchem Ort bzw. Bauteil diese Information erforderlich ist. Der produktseitige Merkmalsexperte kann diese Information bereitstellen.

Häufig besitzen Ressourcen inhärente Einrichtungsmechanismen („Teach-in“), mit denen Größen, die analytisch nur aufwendig zu beschreiben wären, anhand der realen Umgebung eingemessen werden können. Beispiele hierfür sind die Festlegung des Werkzeugmittelpunkts bei Industrierobotern oder eine Bestimmung des produktseitigen Reflektionsverhaltens bei Lichttastern. In solch einem Fall befindet sich das Gerät im

Betriebszustand Not Configured (vgl. Abschnitt 7.2 und Abbildung 6.7). Während der Überprüfung der Geräteeignung erkennt dies der ressourcenspezifische Experte und erstellt ein Subproblem vom Typ *Setup*.

Der allgemeine Setup-Experte nimmt sich der Problemlösung an (\rightarrow AP8), indem er den Setup-Dienst im Digital Twin der Ressource ausführt. Dieser startet einen generischen MMI-Ablauf: Der Digital Twin sendet Anweisungen, wie der Einstellungsschritt durchzuführen ist, evtl. zusammen mit weiteren Informationen (z. B. Schaubilder, Diagramme). Der Experte reicht dies an das GUI des Frameworks weiter, wo der Anwender die Anweisungen ausführt und ggf. angeforderte Informationen zurückliefert. Nach Rückmeldung an das Gerät beginnt der Interaktionsablauf von vorne, solange bis die Einrichtung aus Ressourcensicht abgeschlossen ist oder fehlschlägt. Abbildung 7.15 gibt den Ablauf als Sequenzdiagramm wieder.

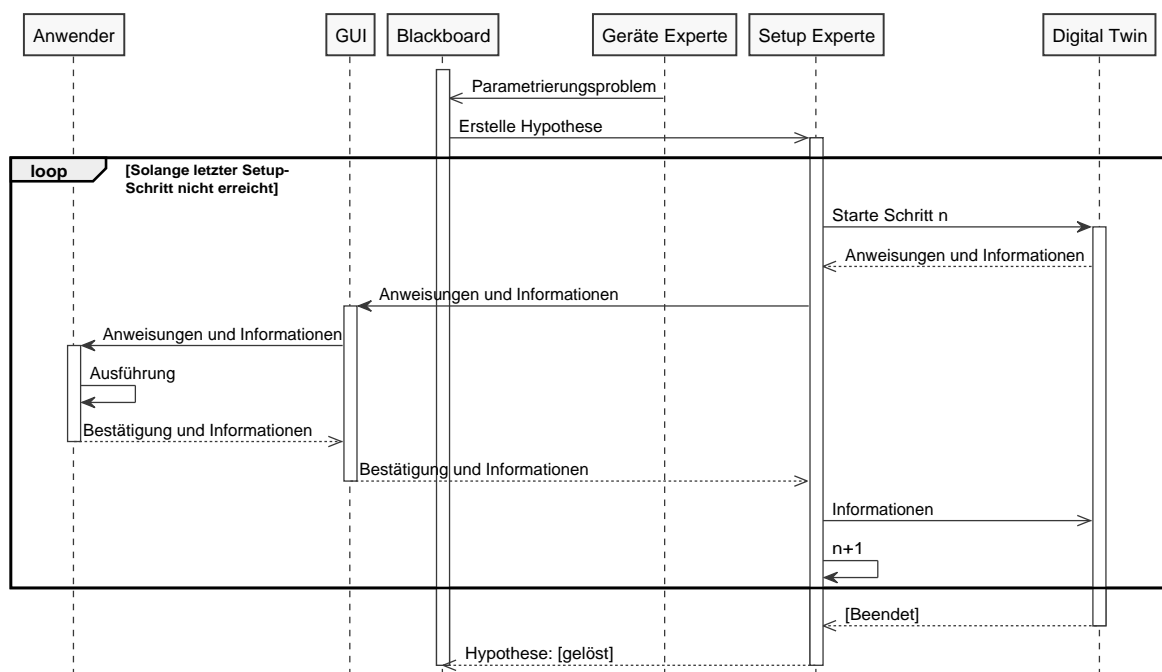


Abbildung 7.15: Ablauf der assistierenden Geräteeinrichtung (Setup). Zur Reduktion der Komplexität wird die Blackboard-Controller-Interaktion nicht dargestellt.

7.3.6 Codegenerator

Der Codegenerator übersetzt den grafischen Montageablaufplan des Prozessmodells in ausführbare Programme der Steuerungseinheiten (\rightarrow AP7). Hierbei werden folgende Schritte durchlaufen:

1. *Erstellung Ablaufplan:* Das grafische Modell wird in das textuelle SFC-Format des PLCopen-Standards überführt.
2. *Suche nach Steuerungen:* Für jede im Prozessmodell zugewiesene Ressource wird deren verbundene Steuerungseinheit gesucht. Hierzu wird die generierte Hierarchie der Ressourceninformationsmodelle genutzt (Abschnitt 5.3), um über die informationstechnischen Schnittstellen die zugehörige Steuerungen zu lokalisieren.
3. *Bildung von Steuerungsgruppen:* Fähigkeiten, die Ressourcen mit gleicher Steuerungseinheit umsetzen, werden gebündelt. Der erstellte SFC wird anhand der Steuerungsgruppen aufgetrennt und Sub-SFCs für jede Gruppe erstellt. Der Gesamt-SFC dient im Anschluss vorrangig der gruppenübergreifenden Koordination und direkten Ansteuerung Smarter Feldgeräte.
4. *Konvertierung der Sub-SFCs:* SPSen können aufgrund der Unterstützung des PLCopen-Standards SFCs ohne Anpassungen importieren. Für andere Steuerungsdomänen (Robotik, BVS) wird der Ablauf in das entsprechende Zielformat gewandelt. Ist das Roboter- oder Bildverarbeitungssystem monolithisch ausgelegt, können die Ressourcen auch als Smarte Feldgeräte gesehen werden, deren Einbindung über den benutzten SOA-Standard (z. B. OPC UA) mittels der Gesamt-Ablaufsteuerung realisiert wird.
5. *Import ressourcenseitiger Steuerungsdaten:* Bis zu diesem Schritt bestanden die erzeugten SFCs aus einem steuerungsspezifischen Fähigkeitenablauf, dessen Fähigkeiten jedoch keinen Inhalt hatten. Nun werden je Fähigkeitsaufruf die korrespondierenden Logikbefehle aus den Digital Twins der Ressourcen geladen und in die Fähigkeiten der Sub-SFCs integriert. Befindet sich die Ressource außerhalb des Steuerungsclusters wird stattdessen ein entsprechender Dienstauftrag eingefügt. Informationen zur kaskadierten Ansteuerung sind in Abschnitt 5.2.3 verfügbar.
6. *Export der Sub-SFCs:* Je Steuerung geschieht dies über den im Digital Twin definierten Download-Dienst (vgl. Abschnitt 7.2).
7. *Start der Steuerungen:* Alle beteiligten Steuerungssysteme werden in den Aktivzustand geschaltet und der Montageablauf beginnt.

Abbildung 7.16a zeigt das zugehörige Strukturbild des Codegenerators und eine beispielhafte Architektur der Anlagensteuerung (Abbildung 7.16b).

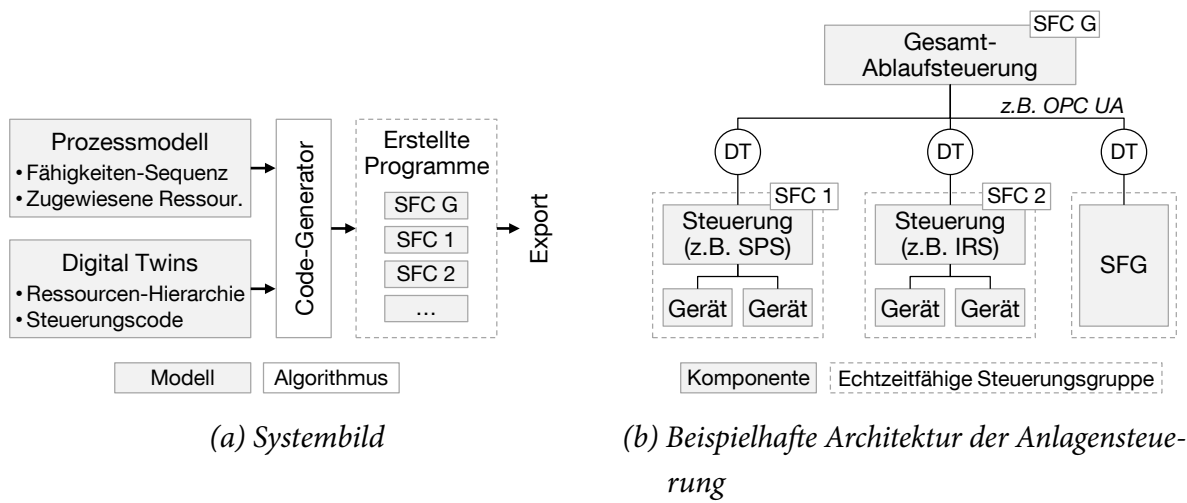


Abbildung 7.16: Codegenerator

8 Evaluation

Für die Evaluation der beschriebenen Konzepte werden zwei Softwareprogramme erstellt: Das Tool zur automatisierten Peripherieanalyse und Konfiguration (TAPAKO) sowie das Framework für das aufgabenorientierte Matchmaking und die Programmierung (TOMAP).

Beide Programme werden gemeinsam anhand verschiedener Anwendungsfälle getestet. In Abschnitt 8.1 wird hierfür zuerst der verwendete Montagedemonstrator beschrieben und dessen Systemintegration mit den erstellten Tools durchgeführt. Da weitere Rekonfigurationen Untermengen der initialen Einrichtung darstellen (vgl. Abschnitt 2.1.2), erfolgt hierauf aufbauend die Beschreibung von drei Rekonfigurationsszenarien und deren Anpassungen zur Ausgangskonfiguration. Der nachfolgende Abschnitt 8.2 gibt eine quantitative Bewertung im Rahmen einer Probandenstudie und einer Wirtschaftlichkeitsbetrachtung. Abschließend werden in Abschnitt 8.3 weitergehende Potentiale der erstellten Lösung aufgezeigt.

8.1 Technische Bewertung

8.1.1 Demonstrator

Als Evaluationsobjekt wird eine modulare Montageanlage verwendet, welche im Rahmen des Forschungsprojekts **AK[⊗]MI** (Automatisierte Konfiguration in der Mikrosystemtechnik) entstanden ist. Die Anlage besteht aus drei kombinierbaren Montagestationen, welche zwei Varianten von LED-Lampen montieren. Jede Lampe besteht aus einem Unterteil mit Spannungswandler, worauf ein Oberteil gefügt wird, das die Platine mit bestückten LEDs enthält.

Der Montageprozess orientiert sich an dem Referenzanwendungsfall aus Abschnitt 4.2. In Modul ① werden leere Werkstückträger (WST) angeliefert und von einem Werker

mit Lampenunterteilen bestückt. Der WST fährt in eine Messposition, in der eine Kamera die eingelegte Lampenvariante prüft und über einen angebrachten Barcode die WST-Nummer ausliest. Die Paarung wird für eine spätere Verwendung in einer Datenbank abgelegt. Im Anschluss fährt der WST mittels Doppelgurt-Förderbändern automatisiert in das Modul ②. An der Fügeposition wird er durch einen Haltezylinder fixiert und die Barcodenummer wird gescannt. Aus der Datenbank wird die korrespondierende Produktvariante geladen, zu der ein Bildverarbeitungssystem das passende Lampenoberteil in einer Lagermatrix ermittelt. Die Position und Orientierung werden als Koordinaten an einen stationären 6-Achs-Industrieroboter übertragen, der mittels eines am Flansch befestigten 3-Finger-Greifers den Handhabungs- und Fügeprozess durchführt. Nach dem Fügevorgang fährt der WST in die Prüfstation von Modul ③, wo dieser erneut fixiert und gescannt wird. Eine Kontaktierstation setzt das Leuchtmittel mittels am WST angebrachter Kontakte unter Spannung. Auf Basis der geladenen Produktvariante wählt ein Bildverarbeitungssystem das entsprechende Prüfprogramm aus und misst die Helligkeit des Leuchtmittels. Für eine Rückverfolgbarkeit wird anschließend das Prüfergebnis zusammen mit der WST-Nummer in einer Datenbank abgespeichert. Liegt das Ergebnis der Messung außerhalb des Toleranzrahmens, wird dies auf einer Signalleuchte an der Anlage für den Bediener entsprechend angezeigt, um eine Nacharbeit einzufordern.

Abbildung 8.1 zeigt ein Foto des Gesamtaufbaus. Weitere Detailansichten des Anwendungsfalls können in Anhang A.6 eingesehen werden.

8.1.2 Softwareimplementierung und Ausgangskonfiguration

Automatisierte Vernetzung

Die entwickelte Plug&Produce-Software TAPAKO ermittelt mit dem in Kapitel 5 vorgestellten Konzept Steuerungssysteme im Netzwerk (Abbildung 8.2a). Die Implementierung erfolgte in C#, die Ausführung auf einem handelsüblichen Desktop-PC.

Der Anwender startet die Suche nach Host-Systemen und SFGs im Ethernet-Netzwerk. Ein Klick auf das entsprechende Gerät durchsucht mit dem beschriebenen rekursiven Suchalgorithmus die jeweilige Steuerungsdomäne. Soweit der Kommunikationskanal

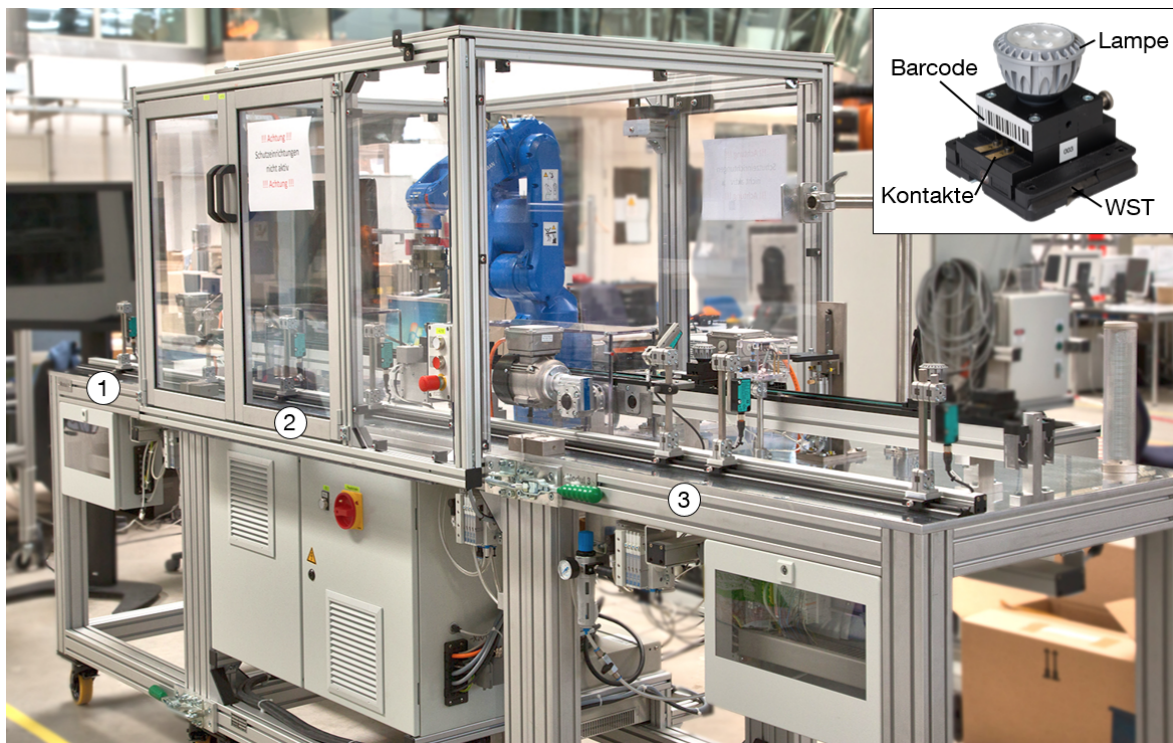
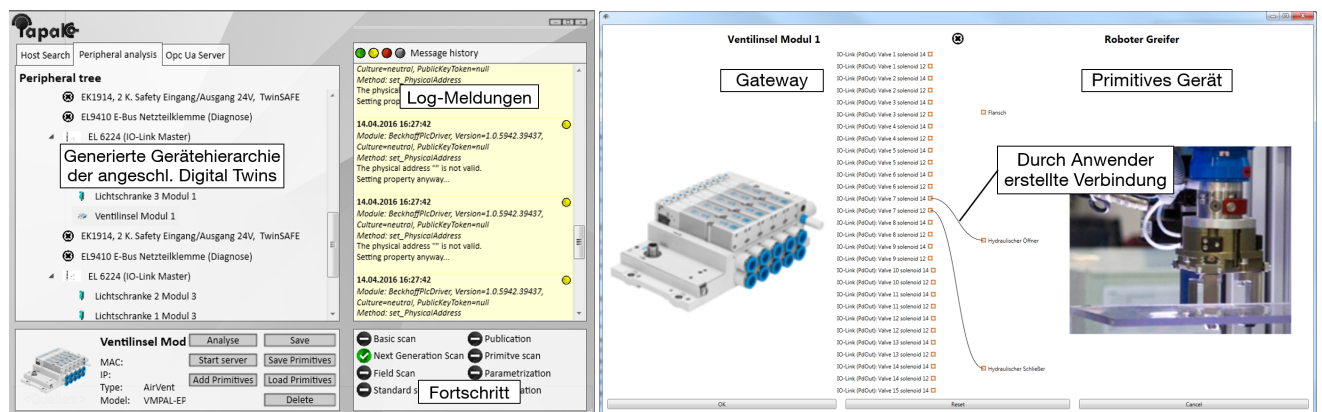


Abbildung 8.1: AKOMI-Montagedemonstrator als Evaluationsobjekt, bestehend aus Modulen ① – ③.

technologisch eine automatisierte Erkennung von Geräten ermöglicht, wird dies erfolgreich automatisiert ausgeführt (AV2✓)¹. Hierbei wurden sowohl KKTs zur Stellvertretersuche über die Schnittstellen der Steuerungen genutzt (ETHERCAT, POWERLINK, IO-Link) als auch bekannte Protokolle selbst implementiert (Ethernet, USB), um steuerungsunabhängig zu sein (AV5✓). Bei primitiven Kommunikationskanälen erscheint der Suchbefehl auf einem GUI, auf diesem unterlagerte Geräte aus einer Datenbank ausgewählt werden können. Nach Laden des Gerätetreibers werden die Schnittstellen des Host- und Peripheriegeräts angezeigt und können durch den Anwender entsprechend den realen Gegebenheiten nachgebildet werden (Abbildung 8.2b). Dies wurde für analoge Geräte (Kontaktierstation in Modul 3) und primitive Druckluftgeräte (Haltezyylinder, Greifer) erfolgreich evaluiert (AV6✓).

Aus den geladenen Gerätetreibern werden die zugehörigen Beschreibungsdateien zur Kommunikation geladen und automatisiert in die Steuerungen integriert, worauf eine echtzeitfähige Ansteuerung erfolgen kann (AV1✓, AV7✓, AV3✓). Getestet wurde dies

¹Eine Übersicht aller gestellten Anforderungen kann Anhang A.1 entnommen werden.



(a) Übersicht der Software

(b) Maske zur Anbindung primitiver Geräte

Abbildung 8.2: Plug&Produce-Software TAPAKO

mit SPSen der Firmen BECKHOFF und B&R, welche neben SIEMENS zu den bedeutendsten Steuerungsherstellern gehören (ROTHHÖFT 2014, S. 118).

Das generierte Informationsmodell wird automatisch publiziert, enthält den Digital Twin aller gefundenen Geräte und stellt diese analog der real vorkommenden Topologie dar (AV8✓). Über einen Rückkanal der Steuerung erfolgt eine Synchronisation des Digital Twins mit dem realen Zustand des Geräts (AV9✓). Als technologische Ausprägung wird das OPC UA-Protokoll verwendet, da OPC UA die bis dato am meisten verbreitete Umsetzung einer SOA in der Automatisierungstechnik darstellt (vgl. WGP 2016 S. 264, PLATTFORM I4.0 2016b, S. 6).

Im Bereich der Robotik erfolgt die Geräteintegration auf Basis des ROS-Frameworks, Distribution „Indigo“. Als beispielhafte Umsetzung wird ein „MH5-F“-Industrieroboter der Firma YASKAWA verwendet. Der Ablauf erfolgt analog des SPS-Konzepts, gefundene Hardware wird als Knoten in das ROS-System geladen und mit einer entwickelten OPC UA-Brücke verknüpft, welche alle aktuellen Zustandsdaten (z. B. Achswinkel) in dem Digital Twin anzeigt und entsprechende Dienstaufträge entgegennimmt. Weitere Informationen sind in HAMMERSTINGL ET AL. (2016) zu finden.

Die Bildverarbeitungsaufgaben werden auf Basis der kommerziellen Software „Halcon“ der Firma MVTec durchgeführt. Über diese erfolgt auch eine Stellvertretersuche, in welcher drei unterschiedliche Kamerasysteme über USB- und GigEVision-Protokolle identifiziert und deren Digital Twins über OPC UA publiziert werden.

Das System wurde erfolgreich mit allen Feldgeräteklassen aus der Referenzarchitektur (Abschnitt 5.1.1) getestet. Es nutzt etablierte Standardkomponenten und Technolo-

gien, um eine möglichst breite Anwendbarkeit zu besitzen. Ein vollautomatisiertes Plug&Produce-Konzept ist je nach Gerätetechnologie und Kommunikationskanal nicht immer möglich, durch die gebotenen Assistenzfunktionen (z. B. Verbindungstool) kann jedoch auch in diesen Fällen eine Erleichterung der Integration geboten werden. Eine Erweiterung um neue Protokolle oder Ressourcen ist aufgrund der Modularisierung der Kommunikationstechnologien und des Gerätezugriffs effizient und unabhängig von bereits vorhandenen Technologien zu bewerkstelligen (AV4✓).

Eine Veröffentlichung des Vernetzungskonzepts erfolgte in HAMMERSTINGL & REINHART (2015b), HAMMERSTINGL ET AL. (2016) und REINHART ET AL. (2017).

Fähigkeiten zur Prozessmodellierung

Es wurden zwei umfangreiche Fähigkeitentaxonomien entwickelt, mit denen der Montageprozess des Demonstrators (Abschnitt 8.1.1) nachgebildet wurde (AF1✓). Durch Orientierung an bekannten Normen ist hierbei eine grundlegende Akzeptanz bei Anwendern zu erwarten. Insbesondere der wichtige Bereich der Kontrolloperationen (PLATTFORM I4.0 2016a, S. 17) war bis dato unterrepräsentiert und konnte umfassend strukturiert werden (AF6✓).

Darüber hinaus wurden ein Vorgehen zur Ermittlung von Fähigkeitenparametern entwickelt, übergreifende Standardparameter definiert und für eine Vielzahl an spezifischen Fähigkeiten festgelegt (AF4✓, AF3✓, AF5✓). Fähigkeiten und Parameter wurden genutzt, um Ressourcen im Demonstrator mit verschiedenen Montageprozessen erfolgreich abzugleichen und diese über die Fähigkeiten anzusteuern. Hierzu wurden aus allen Hauptkategorien der Aktorik, Sensorik, Sonderoperationen, MMI und Datenaustausch Fähigkeiten exemplarisch auf Ressourcen implementiert (AF7✓). Der Fokus lag hierbei auf elementaren Fähigkeiten im Betrachtungsbereich Montage. Fähigkeiten im ur- und umformenden Bereich gilt es darauf aufbauend zu entwickeln.

Durch eine Definition und Auflistung zusammengesetzter Fähigkeiten wurde eine stärkere Unterscheidung zu elementaren Fähigkeiten getroffen (AF2✓) und ein erster Schritt in Richtung einer informationstechnischen Beschreibung zusammengesetzter Ressourcen unternommen. Nachdem in diesem Bereich Emergenzeffekte² auftreten,

²Ein Beispiel ist das Verbinden zweier Lichtschranken mit der Fähigkeit Anwesenheit prüfen, wodurch sich die neue Fähigkeit Höhe messen ergibt.

d. h. sich gänzlich neue Fähigkeiten und Merkmale durch die Komposition von Subsystemen ausbilden, kann eine rein analytische Herleitung oft nicht mehr erfolgen. Hierzu müssen weitere Konzepte untersucht werden, bspw. auf Basis von Inferenzmethoden oder Ansätzen der künstlichen Intelligenz.

Auch Fähigkeiten und Merkmale des Menschen können über die entwickelten Taxonomien informationstechnisch abgebildet und eingesetzt werden (vgl. Abschnitt 8.1.2). In vorliegender Arbeit stand die Definition von Fähigkeiten und deren Parameter im Fokus, eine Methodik zur Quantifizierung menschlicher Fähigkeitenmerkmale („Wie genau kann Person X Abstände messen?“) war nicht Gegenstand der Untersuchungen, wird durch das Voranschreiten der Digitalisierung jedoch zukünftig von zunehmender Bedeutung sein.

Die Veröffentlichung der Fähigkeiten erfolgte in HAMMERSTINGL & REINHART (2018), REINHART ET AL. (2017) und HAMMERSTINGL ET AL. (2015).

Aufgabenorientierte Ressourcenauswahl und Programmierung

Das Prozessmodell wird durch eine Desktop-Anwendung erstellt, in der Anwender die definierten Fähigkeiten aus einer Liste auswählen und zu einem Prozessablauf verknüpfen (Abbildung 8.3). Die Umsetzung orientiert sich an bekannten Modellierungstools wie bspw. LabVIEW (National Instruments) oder Simulink (MathWorks) (AP1✓). Die Angabe von Prozessparametern erfolgt auf Basis von Parameterknoten. Ablauf- (orange), Material- (grün) und Datenflüsse (blau) können getrennt spezifiziert werden, ebenso wie die Verbindung mit Ressourcen (schwarz). Im vorliegenden Anwendungsfall wird bspw. das Ergebnis der Helligkeitsprüfung in Modul ③ als Eingangsgröße für die Anzeige-Fähigkeit der Signallampe verwendet (AP5✓, AP6✓).

Auf Basis des Prozessmodells können geeignete Montageressourcen gefunden werden. Das zugehörige TOMAP-Framework wurde in C# implementiert und einem handelsüblichen Desktop-PC ausgeführt. Es ermöglicht die Lösung von unterschiedlichen Problemklassen durch Expertenmodule. Auch die Erzeugung von Subproblemen ist erfolgreich, welche dann durch das Blackboard angenommen und an geeignete Experten weitergeleitet werden. Der sich hierbei ergebende Problembaum wird in der Software grafisch visualisiert und ist auszugsweise in Abbildung 8.4 dargestellt (AR3✓, AR5✓).

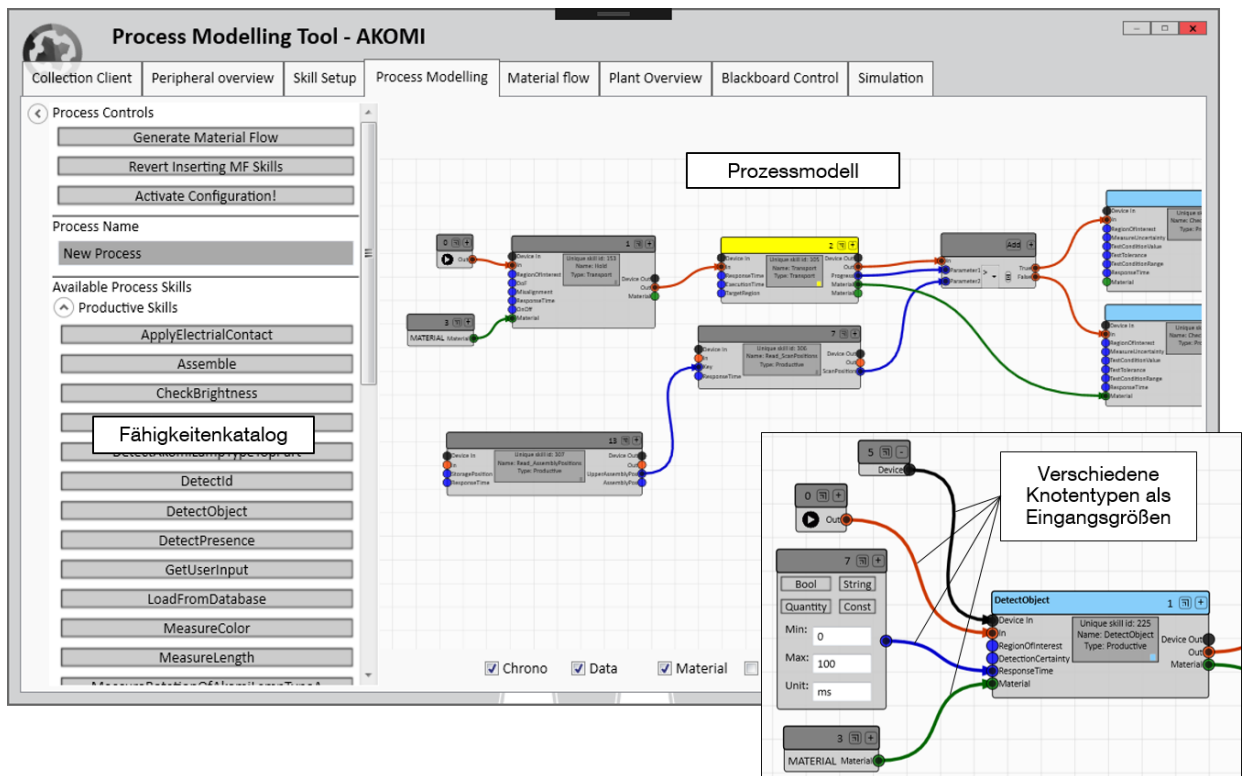


Abbildung 8.3: GUI zur Prozessmodellierung im TOMAP-Framework. Fähigkeitenparameter werden durch unterschiedliche Knotentypen (Ressource, Ablauf, Größe, Material) befüllt.

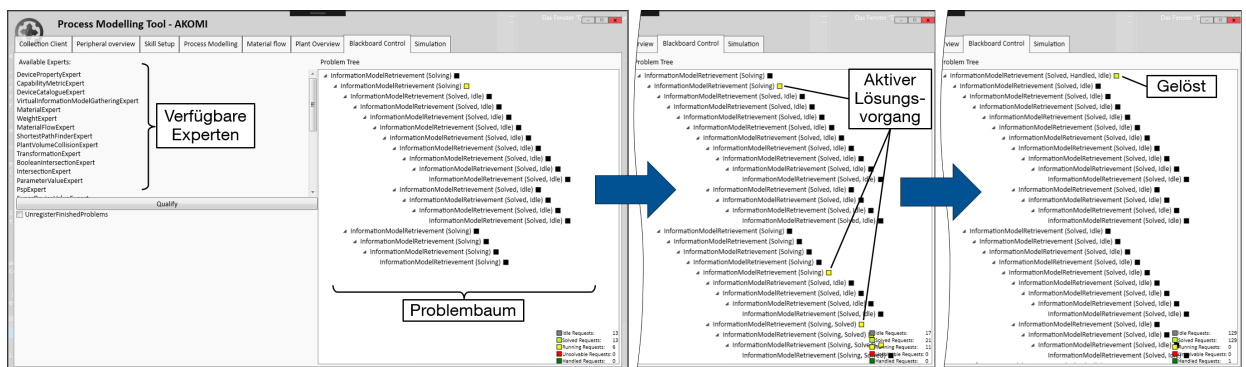


Abbildung 8.4: Visualisierung des erzeugten Problembaums und dessen schrittweiser Lösung

Im Rahmen der Evaluation werden 14 allgemeingültige, 5 prozessspezifische und 9 ressourcenspezifische Experten exemplarisch umgesetzt, welche die in Abbildung 7.8 aufgelisteten Problemarten behandeln, u. a. zur Informationsakquise, Eignungsprüfung, Ressourcenkombination, Koordinatentransformation, Graphensuche (AR4✓). Die Dauer zur Erzeugung und Lösung der Probleme ist stark abhängig von der Problemart, Implementierung des Experten-Aktionsteils sowie dessen Ausführungsumgebung

(lokal, remote etc.). Ein Großteil der Probleme entstammt der Kategorie „Informationsakquise“, deren Einzellösungen meist im Sub-Sekunden-Bereich liegen. Die Gesamtzahl an gelösten Problemen liegt für das vorliegende Aufgaben- und Umweltmodell des Demonstrators (Abschnitt 8.1.1) in der beispielhaften Implementierung bei ca. 400.

Schlägt die Problemlösung eines Experten fehl, ist dies grafisch im Problembaum ersichtlich und Anwender können bei Bedarf eine Modifikation des Experten anstreben. Eine Erweiterbarkeit um neue Problemstellungen und Experten ist durch die verwendete Architekturform per se gegeben (AR6✓). Wird für ein Problem kein (informationstechnischer) Experte gefunden, erfolgt eine Weiterleitung an den Menschen als universellen Problemlösungsexperten. Hierzu erscheint die Problemstellung auf dem GUI und erlaubt die Eingabe der in der Problemklasse definierten Antwortparameter (AR7✓).

Das Vorkommen von Dead- und Lifelocks wird untersucht, indem gezielt fehlerhafte Experten implementiert werden. Für den ersten Fall wird ein Experte erstellt, der keine Antwort liefert, da dessen Prozess blockiert ist. Nach Überschreiten der angegebenen Ausführungszeit wird der zugehörige Slot durch ein Überwachungsmodul („Watchdog“) beendet. Zur Behandlung von Lifelocks wird als Methode die Überwachung der Aufruftiefe festgelegt. Eine vollständige Vorhaltung des gesamten Problembaums und aller seiner Zustände würde sehr hohe Anforderungen an die Rechnerressourcen stellen und hierdurch die Problemlösungszeit stark ansteigen lassen. Die Aufruftiefe wird mit 300 definiert und im Testszenario bei sich gegenseitig aufrufenden Experten bei dieser Zahl erfolgreich abgebrochen. Im vorliegenden Anwendungsfall wird eine Tiefe größer 30 nicht erreicht.

Ressourcenkombinationen werden anhand der in Abschnitt 7.3.5 beschriebenen Beispiele des Förderband- und Roboterverbunds untersucht. Jede Komponente besitzt einen eigenen Experten, der im Testszenario die angefragten Schnittstellen- oder Merkmalsinformationen berechnet bzw. weiterleitet (AR9✓). Im Falle des Förderbands werden die eingegebenen Transportparameter schrittweise über die angeschlossenen Subkomponenten eingeschränkt, während im Falle des Roboterverbunds dessen Experte die Traglast aufgrund des Verbundpartners entsprechend reduziert.

Für die Berechnung der Ressourceneignung wird eine Reihe von exemplarischen, hardwarespezifischen Experten erstellt, u. a. für Greifer, Roboter, Förderband, Lichttaster, Barcodescanner, Kameras. Über den Ressourcenknoten im Prozessmodell lässt sich eine Suche starten (Abbildung 8.5). Das Framework berechnet über die Experten eine

Eignung und liefert zurück, ob die angegebenen Parametermengen eingehalten werden können (AR1✓, AR2✓). Daraufhin kann der Anwender eine Auswahl treffen.

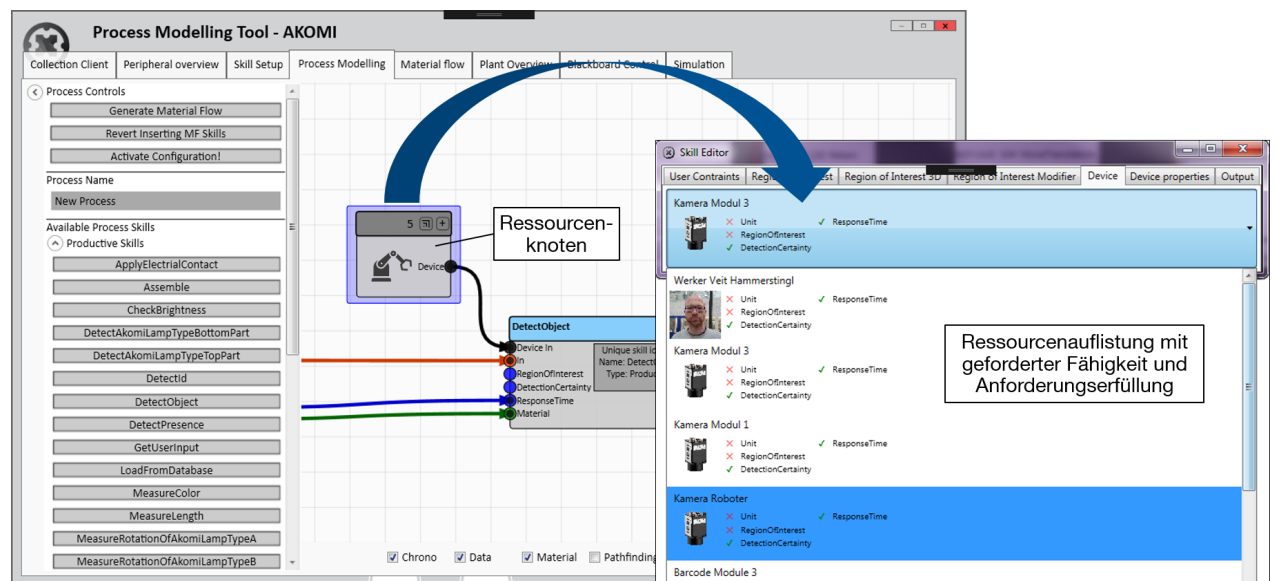


Abbildung 8.5: Über Ressourcenknoten können Anwender eine Eignungsprüfung starten und Geräte zuweisen. Die Einhaltung der geforderten Prozessparameter wird durch das Framework einzeln aufgelistet.

Zur Beschreibung der Arbeitsräume und Produktdaten wird ein voxelbasierter Ansatz gewählt, um eine effektive Arbeitsraumverschneidung und leichtere ROI-Eingabe zu erreichen. Abbildung 8.6 zeigt die aus den Digital Twins automatisch erzeugte Darstellung. Über den Arbeitsraumexperten werden Verschneidungs-, Richtungs- und Sichtbarkeitsabgleiche durchgeführt – befindet sich eine Komponente außerhalb des Toleranzbereichs, wird sie erwartungsgemäß aussortiert.

Eine Erweiterung der Montagesequenz um Transportprozesse wird durch den implementierten Materialflussexperten erreicht. Analog dem in Abschnitt 7.3.5 beschriebenen Vorgehen, wird aus den bestehenden Digital-Twin- und PSP-Daten ein Anlagengraph erstellt und es werden geeignete Transportpfade für den Anwender visualisiert (AP2✓, AP3✓). Für alle Materialflusspfade können Detailinformationen eingeblendet werden (vgl. Abbildung 8.7). Auf Basis der ermittelten Pfad-Metriken (z. B. Strecke, Transportzeit) kann der Anwender gewünschte Materialflusswege und -transportkombinationen auswählen (AP4✓). Nach Beendigung der Aktivität werden erforderliche Handhabungsprozesse automatisiert in das Prozessmodell eingefügt.

Weiterhin wurde zur Lösung nicht automatisiert bestimmbarer Geräteparameter ein Setup-Experte implementiert, der über die entsprechenden OPC UA-Gerätedienste

8.1 Technische Bewertung

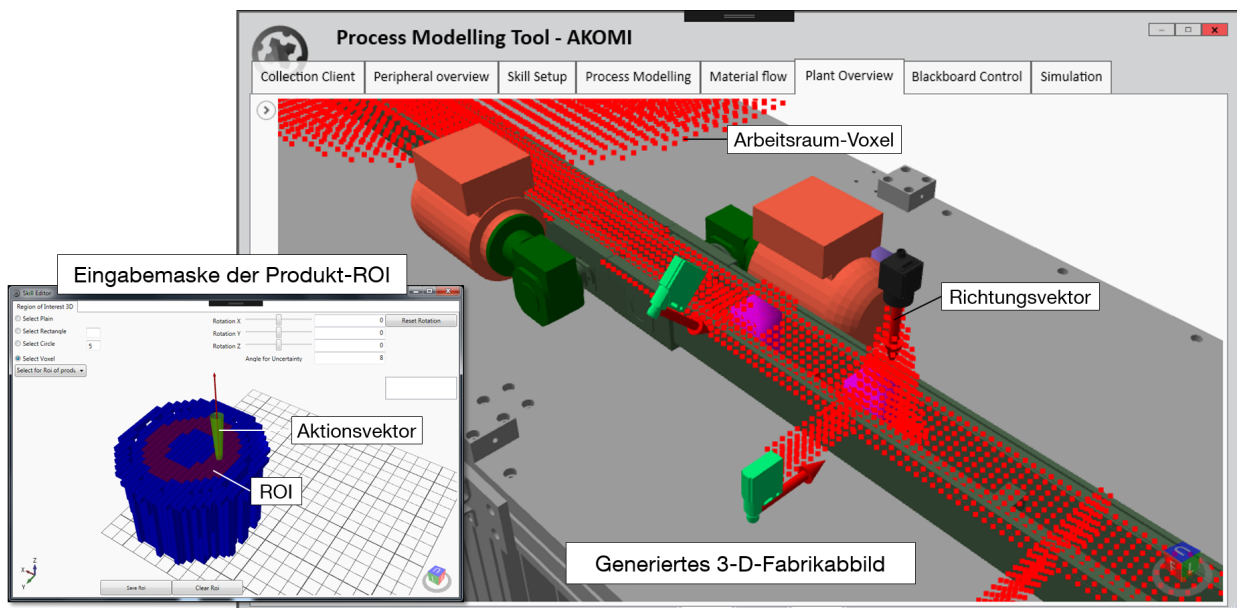


Abbildung 8.6: Das Arbeitsraum- und Produktmodell werden mit Voxeln abstrahiert. Anwender können produktseitige Voxel markieren, um ROIs zu erstellen. Richtungsvektoren erlauben eine orientierungsabhängige Eignungsberechnung.

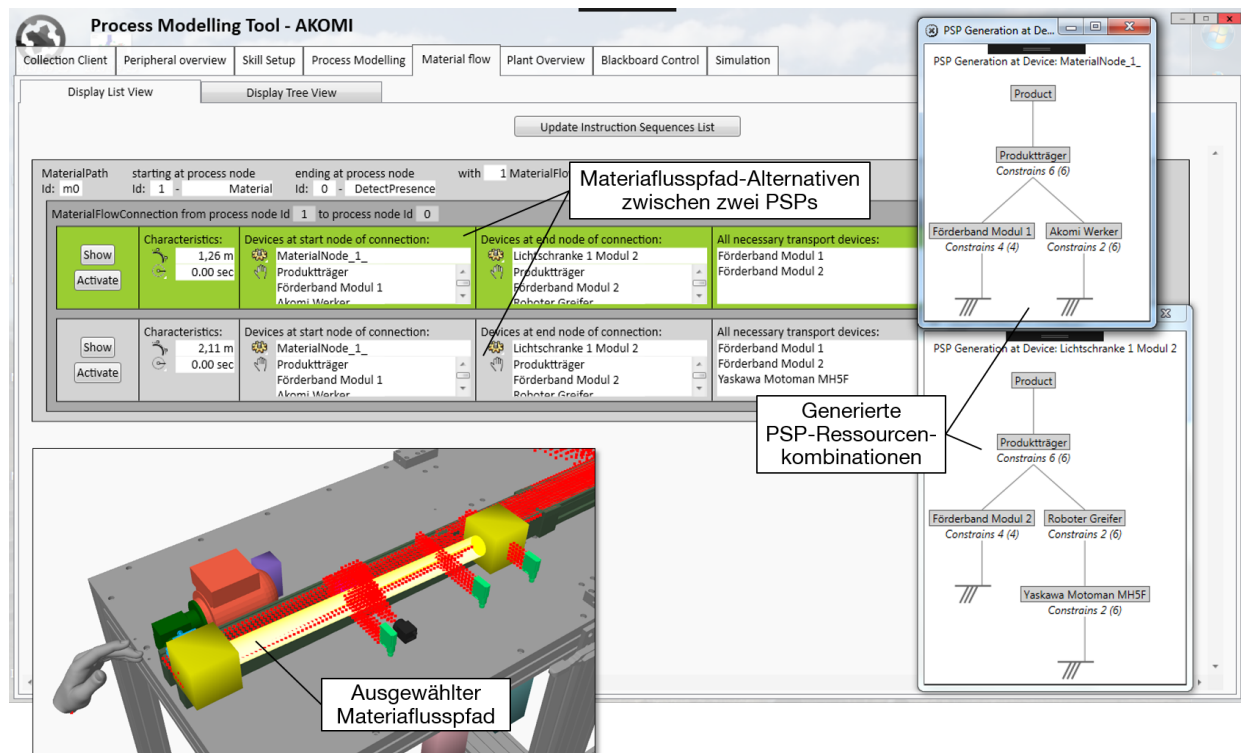


Abbildung 8.7: Hauptfenster: Darstellung zweier Materialflusspfade mit zugehörigen Merkmalen (Distanz etc.). Untere Detailansicht: Ausgewählte Materialflusspfade werden grafisch im Anlagenlayout hervorgehoben. Rechte Detailansichten: Darstellung generierter PSP-Ressourcenkombinationen auf Basis beschränkter DOF und verbundener Schnittstellen.

eine Mensch-Maschine-Interaktion bereitstellt. Geräte übertragen hierbei ein Set an Bild- und Textanweisungen sowie die Strukturen der benötigten Antwortdaten. Nach Aktionsdurchführung und Quittierung durch den Anwender wird das nächste Set ausgetauscht, solange bis die Einrichtsequenz abgeschlossen ist oder abgebrochen wird. Die Routine wird am Demonstrator erfolgreich eingesetzt, um die Fähigkeiten des Bildverarbeitungssystems einzurichten (Helligkeit, Maßstab etc.; AP8✓).

Der Codegenerator erzeugt aus der modellierten Fähigkeitensequenz samt zugewiesenen Geräten einen SFC im PLCopen-Format. Der Export findet auf zwei SPSen vom Typ „CX-2030“ der Firma BECKHOFF statt. Nach dem Export werden die Steuerungen automatisiert in den Betriebsmodus versetzt und der Montageprozess des Demonstrators beginnt entsprechend dem Modell (AP7✓). Die Codegenerierung für das Robotik- und Bildverarbeitungssystem wird als nicht erkenntnisfördernd gesehen, hierzu existieren bereits zahlreiche Arbeiten (z. B. RIDGE ET AL. 2017 oder FRIEDRICH 2010). Aus diesem Grund werden die Komponenten als Smarte Feldgeräte implementiert und kommunizieren mit den Steuerungssystemen über OPC UA.

Die beschriebenen Ansätze wurden auszugsweise in LEIBER ET AL. (2018) und HAMMERSTINGL & REINHART (2017) veröffentlicht.

8.1.3 Rekonfigurationsszenarien

Für die initiale Systemintegration des Demonstrators können bereits zahlreiche Bestandteile der beschriebenen Konzepte angewandt werden (vgl. Abschnitt 8.1.2). Dies wird als Anwendungsfall 0 (engl. Use Case, UC 0) bezeichnet. Hiervon ausgehend, werden nachfolgend weitere Rekonfigurationsszenarien entwickelt und deren softwareseitige Abläufe beschrieben.

Rekonfiguration bedeutet eine Anpassung aufgrund geänderter Randbedingungen. Diese können nach EILERS (2015, S. 17 f.) Veränderungen am Produkt, der Stückzahl oder den eingesetzten Montagetechnologien sein. Unter Berücksichtigung der möglichen Flexibilitätsarten in Montagesystemen (vgl. Abschnitt 2.1.3) ergeben sich hieraus drei Anwendungsfälle:

Im *ersten* Anwendungsfall (UC 1) tritt eine starke Absatzerhöhung auf, weshalb eine zusätzliche, roboterbasierte Fügestation eingebracht wird. Beide Stationen werden parallel betrieben, um der hohen Stückzahl gerecht zu werden. Dies erfordert nach EVERSHEIM

(1983, S. 27) eine zeitpunktorientierte *Umsteuerflexibilität*.

Im *zweiten* Anwendungsfall (UC 2) soll eine zusätzliche Produktvariante montiert werden. Diese besitzt geänderte geometrische Abmaße. Hierbei ist zu überprüfen, ob das neue Produkt mit den vorhandenen Betriebsmitteln noch immer montiert werden kann und bei Erfolg ist das System entsprechend neu zu programmieren und parametrieren (*Umstell-* und *Umrüstflexibilität*).

Im *dritten* Anwendungsfall (UC 3) fällt das Kamerasystem im Bestückungsmodul ① aus. Um die Stillstandszeit gering zu halten, soll die Aufgabe der Lampenzuweisung eine menschliche Arbeitskraft mit Smart Device übernehmen. Dies entspricht einer *Störungs-* und *Umbauflexibilität*.

Alle Anwendungsfälle decken sowohl physische als auch logische Anpassungen des Systems nach ELMARAGHY (2006, S. 270 f.) ab. Abbildung 8.8 zeigt sie schematisch in der Übersicht.

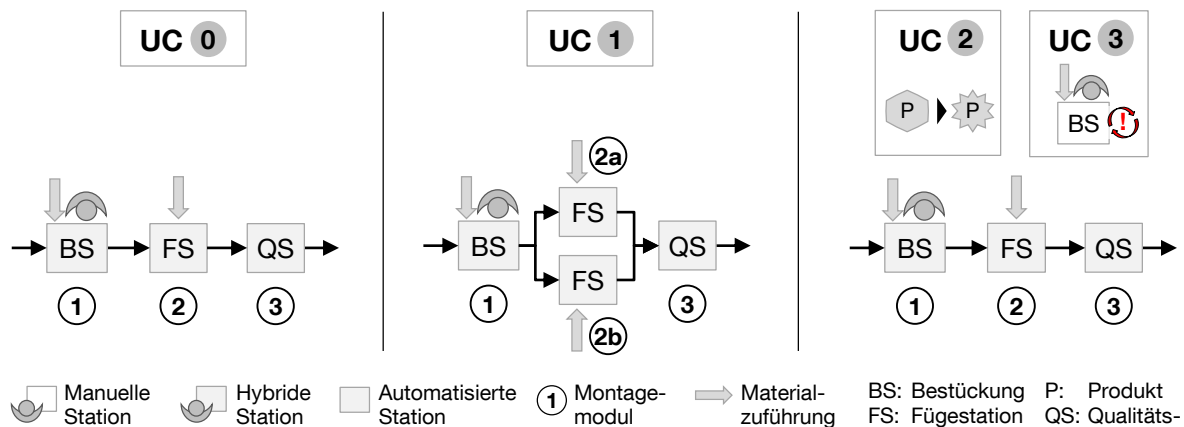


Abbildung 8.8: Schematische Übersicht der Stationsverkettung für Anwendungsfälle UC 0 – 3.

Bezogen auf die Ausgangskonfiguration wird in UC 1 ein paralleler Materialfluss eingebracht. Hierbei ist die modellierte Montagesequenz anzupassen, indem ein zweiter Fügeprozess als Fähigkeit integriert und die Materialflussverbinder entsprechend angelegt werden. Die Abfrage, welche der beiden Fähigkeiten zu nutzen ist, wird mithilfe der Logikknoten beschrieben. Nachdem im Demonstrator nur ein Robotersystem zur Verfügung steht, erfolgt die Umsetzung der zweiten Station simulativ (Abbildung 8.9). Hierzu werden die Digital Twins der virtuellen Komponenten aus einem lokalen Repository geladen und in das TOMAP-Framework integriert. Bezüglich des Gesamttablaufs hat dies keinerlei Auswirkungen, da das Framework rein auf den Digital-Twin-Modellen operiert, unabhängig von deren Herkunft. Für das zweite Fügemodul kann der Anwender im Prozessmodell die virtuellen Aktoren und Sensoren auswählen, deren Eignungs-

und Materialflussprüfungen identisch zu realen Ressourcen ablaufen. Wird die Ablaufsteuerung des realen Demonstrators gestartet, lädt die Simulationsumgebung dynamisch alle zugewiesenen Ressourcen und erstellt ein Simulationsmodell samt zugehörigem OPC UA-Server zur Digital-Twin-Publikation. Im Codegenerator treten virtuelle Geräte als SFGs auf, weshalb diesbezügliche Steuerungsaufrufe als OPC UA-Befehle an die Simulation weitergeleitet und dort im Verhaltensmodell der Geräte ausgeführt werden. Durch den Ansatz lassen sich rein virtuelle Geräte innerhalb des realen Anlagenverbunds auf Eignung prüfen und deren Zusammenspiel bereits vor einer realen Verbauung testen (AR8✓).

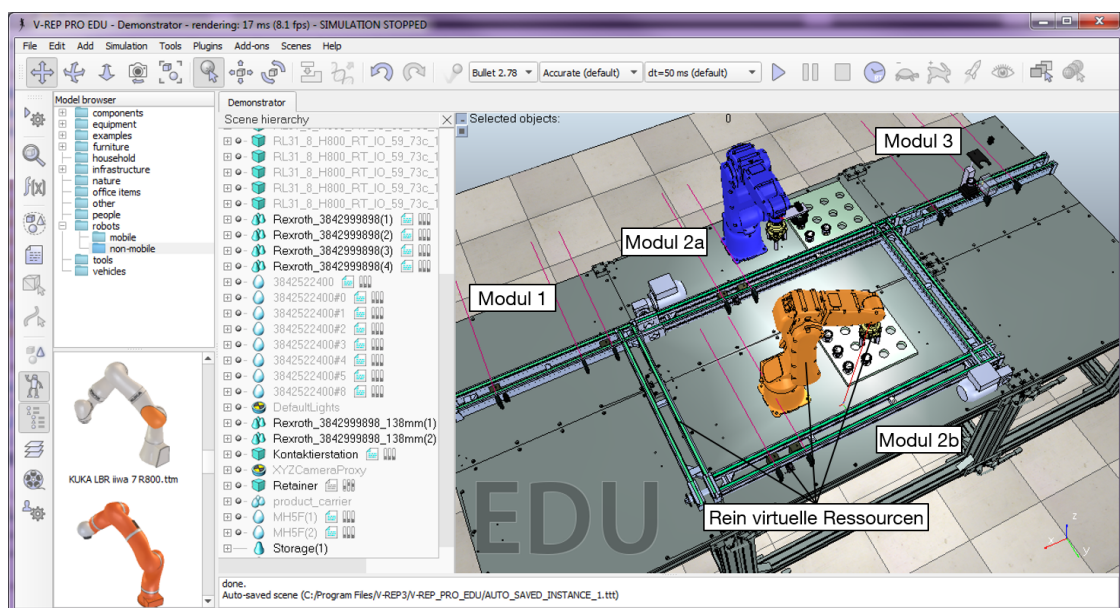


Abbildung 8.9: Beschreibung der zweiten Montagestation als rein virtuelle Ressourcen innerhalb der 3D-Kinematiksimulation „V-REP“ (COPPELIA ROBOTICS). Über einen verknüpften OPC UA-Server sind diese als SFGs durch die Ablaufsteuerung kontrollierbar.

In UC 2 tritt eine Änderung der Produkteigenschaften auf (größere Abmaße, neues Außenmaterial). Der modellierte Montageablauf kann weitestgehend bestehen bleiben, nachdem Produktdaten automatisiert aus dem PDM-System geladen werden. Treten auch Prozessanpassungen auf, wie etwa neue Spaltmaßtoleranzen, müssen diese Anforderungen in der Fähigkeitensequenz entsprechend aktualisiert werden. Dies trifft auch auf die ROI-Angaben zu, welche durch die geänderten Abmaße neu eingegeben werden müssen. Das anschließende Matchmaking und die Codegenerierung laufen analog der Ausgangskonfiguration ab. Hierbei sind u. U. andere Ressourcen als in der ersten Variante geeignet. Kann die neue Produktvariante nicht montiert werden, sind die verantwortlichen Merkmale in der Eignungsanzeige je Ressource ersichtlich (Abbil-

dung 8.5). Anwender können so entsprechend das Produktdesign oder die Ressource(n) anpassen.

In UC ③ bleibt der modellierte Prozess unverändert, lediglich die auszuführende Ressource für die Identifikation und Barcodemessung ist anzupassen. Über ein handelsübliches Tablet ist der Werker mit dem Framework verbunden, welches seine Fähigkeiten in der entsprechenden Digital-Twin-Struktur bereitstellt. Hierdurch kann das Matchmaking identisch zu technischen Ressourcen erfolgen, der gewünschte Arbeitsbereich des Werkers wurde zuvor im Tablet eingetragen. Im Framework selektiert der Anwender für betroffene Fähigkeiten anstatt des defekten Geräts neue Ressourcen – in diesem Fall das Smart Device des Werkers. Nach der Codegenerierung startet der Montageablauf, die Steuerung ruft das Tablet als SFG an den entsprechenden Prozessschritten auf. Das Smart Device wandelt den Aufruf in hinterlegte visuelle und akustische Informationen, die durch den Werker ausgeführt werden (hier: Eingabe des Barcodes). Nach der Quittierung wird das Ergebnis zurück an die Steuerung gesendet und der Ablauf fortgeführt.

8.1.4 Fazit anhand der Veränderungsbefähiger

Durch das beschriebene Konzept der *automatisierten Vernetzung* können Anwender ohne Expertenwissen und in geringer Zeit neue Geräte in Steuerungssysteme integrieren – auch für deterministische Anwendungsfälle. Die Abstraktion auf eine Kommunikationstechnologie (hier: OPC UA) erlaubt eine Ansteuerung unabhängig von Hersteller und Kommunikationsweg. Bezogen auf die Wandlungsbefähiger von WIENDAHL (Abschnitt 2.1.3) steigert das Konzept die *Skalierbarkeit* und *stufenweise Automatisierbarkeit* von Montagesystemen (bedingt durch eine einfache Komponentenintegration) und verbessert die *Kompatibilität* der Betriebsmittel (durch eine standardisierte Digital-Twin-Technologie).

Innerhalb des *Digital Twins* führt die Standardisierung von dessen Semantik und Struktur zu einer Verbesserung der *Modularität* von Montageressourcen. Hierzu zählt insbesondere das *Fähigkeitenmodell* als prozesseitige Abbildung der Ressourcenfunktionalitäten.

Das Framework zur *aufgabenorientierten Ressourcenauswahl und Programmierung* erlaubt eine effizientere Rekonfiguration der Montageanlage, indem sowohl engineering-

bezogene Aktivitäten (Matchmaking) als auch softwareseitige Handlungen (Codegenerierung) automatisiert werden. Übertragen auf die Wandlungsbefähiger wird vorrangig die *Universalität* der Ressourcen gesteigert, da effizient ein Abgleich auf neue Anforderungen und Aufgaben durchgeführt werden kann. Dies wirkt sich auch indirekt auf die *Mobilität* der Ressourcen aus, da durch die verschiedenen Ansätze zur Eignungsprüfung, Einrichtung und Programmierung Geräte leichter als bisher in neue Umgebungen eingebracht werden können.

8.2 Quantitative Bewertung

8.2.1 Probandenstudie

Zur quantitativen Bestimmung des Nutzens wurde eine Probandenstudie mit 9 Teilnehmern durchgeführt, in der die Zeiten des herkömmlichen Vorgehens und der entwickelten Ansätze verglichen wurden. Die Personen sind mehrheitlich erfahrene Anwender im Bereich der Automatisierungstechnik (vgl. Abbildung 8.10) und wurden zuvor in den Aufbau und Funktionsweise des Demonstrators aus Abschnitt 8.1.1 eingewiesen.

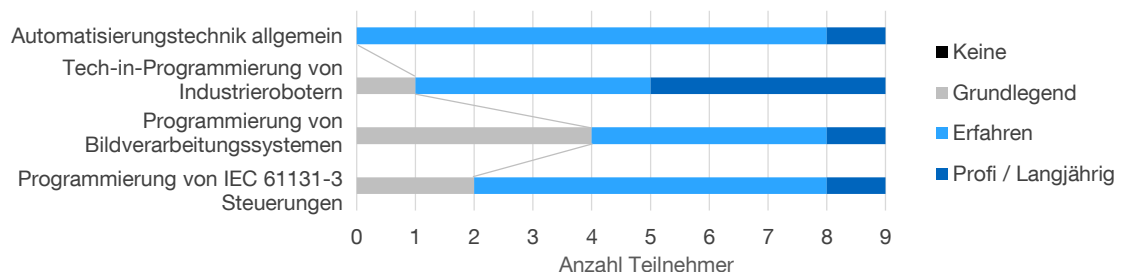


Abbildung 8.10: Selbsteinschätzung der Probanden bez. Kenntnisse in der Automatisierungstechnik

Ausgangskonfiguration

Im ersten Fall wurde die Zeitdauer der initialen Systemintegration des Demonstrators (UC ①) ermittelt. Hierbei wurde zwischen der reinen Einrichtung (Kommunikation, Parametrierung) und der Programmierung des Systems unterschieden. Für das konventionelle Vorgehen wurde die Konfigurationsdauer auf durchschnittlich 20,1 Stunden und die erforderliche Programmierzeit auf 63,2 Stunden geschätzt. Nach Einweisung in die erstellten Tools (TAPAKO und TOMAP) wurden die neuen Zeiten mit 1,4 Stunden und 6,5 Stunden angegeben, was einer zusammengefassten Zeitersparnis von 90,5 %

entspricht (Abbildung 8.11). Aufwände entstehen im neuen System durch Modellierungszeiten für primitive Geräte in TAPAKO sowie das Anlegen des Montageprozesses in TOMAP. Hinzu kommen Rechenzeiten der jeweiligen Softwares.

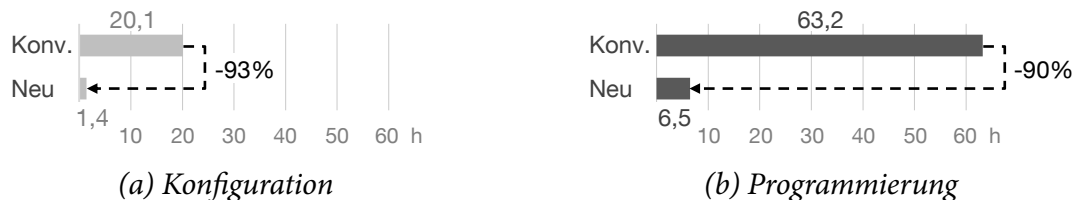


Abbildung 8.11: Zeiten zur initialen Systemintegration des Demonstrators (UC 0)

Als nächster Aspekt wurden die erforderlichen Kenntnisse zur Durchführung der Systemintegration befragt. Als Tenor wurde identifiziert, dass sich durch das Konzept spezifisches Wissen zur Programmierung in den einzelnen Steuerungsdomänen stark reduzieren lässt (Abbildung 8.12a). Generelle Kenntnisse in der Automatisierungstechnik können zwar verringert werden, besitzen jedoch nach wie vor eine hohe Bedeutung (Abbildung 8.12b). Unverändert wichtig bleibt die Anforderung eines umfassenden Verständnisses über den Montageprozess (Abbildung 8.12c). Dies erscheint nachvollziehbar, da Anwender auch mit dem neuen Konzept Prozessanforderungen definieren müssen. Hardwarespezifisches Wissen kann somit reduziert werden, während der Anwender nach wie vor als Prozessspezialist bestehen bleibt und sich verstärkt der Aufgabendefinition widmet. Eine Übersicht aller gegebenen Antworten kann Anhang A.7 entnommen werden.

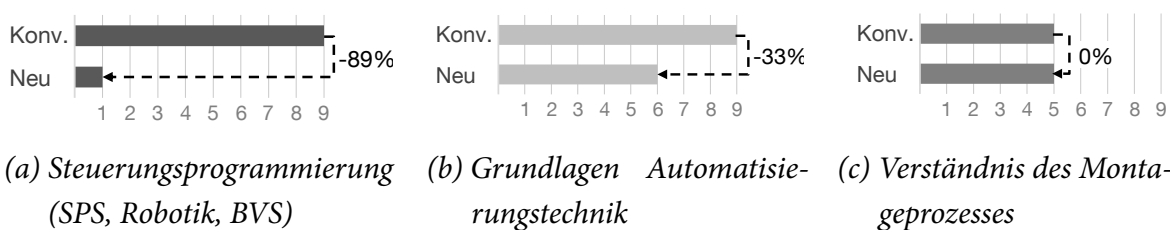


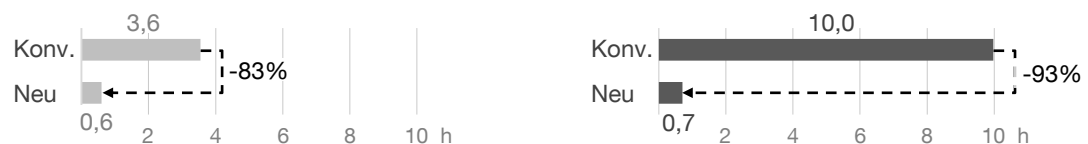
Abbildung 8.12: Notwendige fachliche Kenntnisse zur Durchführung des Systementwurfs und der Systemintegration (Anzahl Nennungen)

Rekonfigurationsszenarien

Innerhalb der Rekonfigurationsszenarien wurden die Probanden zu UC 2 und UC 3 befragt. UC 1 wurde aufgrund der Ähnlichkeit zu UC 0 nicht gesondert untersucht.

Für den Fall einer neuen Produktvariante (UC 2) wurden die Zeiten für unterschiedliche Merkmalsanpassungen erhoben (z. B. Änderung der Geometrie, neues Material) und im Anschluss gemittelt, um eine vom einzelnen Merkmal unabhängige Metrik zu erhalten. Die Einzelzeiten können Anhang A.7 entnommen werden.

Unterschieden wurde zwischen der erforderlichen Zeit, um eine *Aussage* über die Montierbarkeit des geänderten Produkts treffen zu können (Eignung des Produktionssystems) und der tatsächlichen *Umprogrammierung* des Systems, um die geänderten Anforderungen bewältigen zu können (Abbildung 8.13). Die hierfür erforderlichen Zeiten sinken von 3,6 h auf 0,6 h bzw. von 10,0 h auf 0,7 h. Bei Einsatz des TOMAP-Frameworks unterscheiden sich die Zeiten zur Machbarkeitsprüfung und Programmierung nur kaum, da für eine Machbarkeitsaussage bereits ein Matchmaking stattgefunden haben muss – die anschließende Programmierung wird automatisiert durch das Framework erledigt.



(a) Dauer, bis Aussage über Montierbarkeit (b) Dauer zur Anpassung der Programmierung getroffen werden kann

Abbildung 8.13: Anpassung der Ausgangskonfiguration auf neue Produktvarianten

Neben geplanten Rekonfigurationen wurden mit UC 3 auch ungeplante Rekonfigurationen (Geräteausfälle) untersucht. Hierzu wurden die geschätzten Zeiten für den Austausch unterschiedlicher Ressourcentypen abgefragt und gemittelt, um eine geräteunabhängige Metrik zu erhalten (Einzelzeiten siehe Anhang A.7).

Es sinken die temporalen Aufwände von 6,2 h auf 1,1 h, was einer Verbesserung von 82,3 % in den IT-seitigen Stillstandszeiten entspricht (Abbildung 8.14).

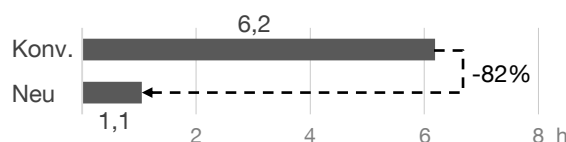


Abbildung 8.14: Gemittelte Zeiten für den Austausch eines nicht-baugleichen Ersatzgeräts

Zusammenfassend ist ein großes Potential in der Automatisierung von Tätigkeiten des Systementwurfs und der Systemintegration ersichtlich. Die ermittelten Zeiten werden

im nachfolgenden Abschnitt genutzt, um das monetäre Potential der Konzepte zu bestimmen.

8.2.2 Wirtschaftlichkeitsrechnung

Neben der vorangegangenen nicht-monetären Bewertung soll das Konzept auch auf seine Wirtschaftlichkeit überprüft werden. Da die Durchführung einer Rekonfiguration keine direkte wertschöpfende Tätigkeit darstellt, wird eine Normkostenvergleichsrechnung angestrebt. Hierzu werden die informationstechnischen Lebenszykluskosten des Montagesystems zwischen dem herkömmlichen Vorgehen („konv.“) und dem beschriebenen Ansatz verglichen („neu“).

Da in beiden Szenarien identische Aufwände für die mechanische und elektrische Anpassung auftreten, werden diese nicht einbezogen. Die Berechnung geschieht anhand der in Abschnitt 8.1.3 vorgestellten Szenarien Anlagenänderung, Produktänderung und Geräteausfall.

Die Lebenszykluskosten $K_{\text{Sys,IT,LCC}}$ setzen sich hierbei wie folgt zusammen:

$$K_{\text{Sys,IT,LCC}} = K_{\text{Invest}} + K_{\text{zyklisch}} = K_{\text{Inv}} + K_{\text{zykl,fix}}(t) + K_{\text{zykl,var}}(n, t) \quad (8.1)$$

$$K_{\text{Sys,IT,LCC,konv}} = K_{\text{Inv,konv}} + K_{\text{Wartung}}(t) + K_{\text{Rekonfig,gepl,konv}}(n, t) + K_{\text{Rekonfig,ungepl,konv}}(n, t) \quad (8.2)$$

$$K_{\text{Sys,IT,LCC,neu}} = K_{\text{Inv,neu}} + K_{\text{Betrieb}}(t) + K_{\text{Wartung}}(t) + K_{\text{Rekonfig,gepl,neu}}(n, t) + K_{\text{Rekonfig,ungepl,neu}}(n, t) \quad (8.3)$$

t steht hierbei für jährliche Kosten, n für häufigkeitsbezogene Kosten. Die einzelnen Summanden setzen sich wie folgt zusammen:

$$K_{\text{Inv}} = K_{\text{InitialModellierung}} + K_{\text{InitialProgrammierung}} = K_{\text{IM}} + K_{\text{IP}} \quad (8.4)$$

$$K_{\text{Betrieb}}(t) = K_{\text{TreiberLizenzen}} + K_{\text{Rechner}} \quad (8.5)$$

$$K_{\text{Wartung}}(t) = 0,1 \cdot K_{\text{Inv}} \text{ (jährliche Software-Wartung und -Anpassungen)} \quad (8.6)$$

$$K_{\text{Rekonfig,gepl}} = K_{\text{SysIntegration}} + K_{\text{Stillstand,gepl}} \quad (8.7)$$

$$K_{\text{Rekonfig,ungepl}} = K_{\text{Komp.tausch}} + K_{\text{Stillstand,ungepl}} \quad (8.8)$$

Es wird angenommen, dass die Produktionsmittel voll ausgelastet sind. Eine Rekonfiguration stellt deshalb einen Produktionsstillstand dar, dessen Kosten $K_{\text{Stillstand}}$ sich

aus den entgangenen Erlösen ergeben. Hinzu kommen die Mitarbeiterkosten K_{Pers} zur Durchführung der Rekonfiguration:

$$K_{\text{SysIntegration}} = t_{\text{Rekonfig,gepl}} \cdot K_{\text{Pers}} \quad (8.9)$$

$$K_{\text{Stillstand,gepl}} = t_{\text{Rekonfig,gepl}} \cdot (E_{\text{Stück}} \cdot \text{Anz}_h) \quad (8.10)$$

$$t_{\text{Rekonfig,gepl}} = t_{\text{Planung}} + t_{\text{Konfig}} + t_{\text{Programmierung}} \quad (8.11)$$

$$K_{\text{Komp.tausch}} = t_{\text{Komp.tausch}} \cdot K_{\text{Pers}} \quad (8.12)$$

$$K_{\text{Stillstand,ungepl}} = t_{\text{Komp.tausch}} \cdot (E_{\text{Stück}} \cdot \text{Anz}_h) \quad (8.13)$$

Zur Bestimmung konkreter Geldwerte werden folgende technische Festlegungen getroffen:

- Kosten der Rechnerhardware K_{Rechner} sind als Infrastrukturdienstleistung in der Cloud umgesetzt. Sie entstehen dadurch jährlich und erfahren keine Abschreibung.
- Kalkulatorische Zinsen $K_{\text{Zins,kal}}$ werden als Opportunitätskosten den Betriebskosten $K_{\text{Betrieb,neu}}(t)$ zugeschlagen.
- Die Bereitstellung der Gerätetreiber erfolgt durch die jeweiligen Hersteller als digitale Dienstleistung und wird mit einer jährlichen Gebühr pro Gerät veranschlagt.

Diese führen in Kombination mit dem angenommenen Montagesystem aus Abschnitt 8.1.1 zu folgenden Größen:

$$K_{\text{Pers}} = 45 \text{ €/h}$$

$$t_{\text{Takt}} = 8 \text{ s}$$

$$K_{\text{IM,konv.}} = 0 \text{ €}$$

$$\text{Erlös}_{\text{Stück}} = 0,30 \text{ €}$$

$$K_{\text{IM,neu}} = (100 \text{ h} \cdot 2 \text{ Pers.}) \cdot 45 \text{ €/h}$$

$$K_{\text{Zins,kal}} = 6 \%$$

$$K_{\text{TreiberLizenzen}} = 40 \text{ Geräte} \cdot 100 \text{ €/Gerät p. a.}$$

$$K_{\text{Rechner}} = 4000 \text{ € p. a. (Rechnerkosten und Lizenzen für Expertenmodule)}$$

Ausführzeiten der Rekonfiguration t_{Rekonfig} werden aus den abgefragten Anwendungsfällen (Abschnitt 8.2.1) entnommen und gemittelt, um so eine ressourcen- und anwendungsfallunabhängige Größe zu erreichen. Zusätzlich müssen noch zwei ergänzende Zeiten bestimmt werden:

- t_{konfig} : Pro Rekonfigurationsfall werden 15 % der Ressourcen hardwareseitig geändert. Zeiten ergeben sich prozentual entsprechend der abgefragten Gesamtkonfigurationszeit ($t_{\text{Rekonfig,gepl,konv.}} = 3,02 \text{ h}$, $t_{\text{Rekonfig,gepl,neu}} = 0,21 \text{ h}$).

- $t_{\text{Programmierung,neu}}$: Im Falle eines ungeplanten Geräteausfalls wird von 50 % der Zeiten für Systemintegration und Programmierung ausgegangen, wie für die Gesamtanlage erforderlich ist ($t_{\text{SysIntegration,neu}} + t_{\text{Programmierung,neu}} = 0,7 \text{ h} + 0,35 \text{ h}$).

Hiermit sind alle Einzelgrößen berechnet. Nachdem sich die Lebenszykluskosten $K_{\text{Sys,IT,LCC}}$ abhängig von der Häufigkeit der geplanten und ungeplanten Rekonfigurationen gestalten, werden diese anhand von drei Szenarien bestimmt:

- *Konservatives Szenario*: Es wird *quartalsweise* eine Rekonfiguration notwendig. Etwaige Geräteausfälle treten *nicht* auf oder werden während geplanter Anpassungen behoben. In diesem Szenario rechnet sich der Einsatz des Konzepts für den beschriebenen Anwendungsfall nach ca. 2,5 Jahren.
- *Moderates Szenario*: Geht von einer *monatlichen* Rekonfiguration aus, pro Jahr tritt *ein* ungeplanter Geräteausfall & Produktionsstillstand auf. In diesem Szenario wird der Break-Even-Point bei Einsatz des Konzepts nach ca. 12 Wochen erreicht.
- *Progressives Szenario*: Hier werden wöchentliche Änderungen des Montageprozesses geplant, es wird von 50 Rekonfigurationen p. a. ausgegangen, bei *zwei* Geräteausfällen pro Jahr. Dadurch ist das Konzept bereits nach ca. 3 Wochen rentabler als der konventionelle Fall.

Abbildung 8.15 stellt die vorgestellten Szenarien grafisch dar, zugehörige Werte sind der Tabelle 8.1 zu entnehmen. Das Konzept weist höhere Invest- und Betriebskosten auf als der konventionelle Fall, kann dies jedoch durch geringere Aufwände und Stillstandzeiten pro Rekonfiguration ausgleichen. Innerhalb von zwei Jahren ergeben sich im moderaten Szenario Einsparungen von ca. 46 000 € für das Unternehmen.

Voraussetzung für die genannten Einsparungen ist, dass entsprechende Gerätetreiber und Expertenmodule am Markt verfügbar sind und nicht durch den Anwender selbst entwickelt werden müssen. Weiterhin ist durch Formeln 8.10 und 8.13 ersichtlich, dass die Kosten für Produktionsstillstände maßgeblichen Einfluss auf die Wirtschaftlichkeit des Konzepts haben.

Neben den vorgestellten Kosteneinsparungen werden weitere positive kaufmännische Effekte erzeugt, welche sich nur schwer quantifizieren lassen. Beispielhaft sei der reduzierte Maschinenstundensatz genannt, aufgrund einer besseren Auslastung der Produktionsmittel, bedingt durch deren höhere Einsatzflexibilität. Weiterhin können sich Absatzsteigerungen für das Unternehmen ergeben, da die Produktion insgesamt eine schnellere Reaktionsfähigkeit auf neue Marktanforderungen aufweisen kann.

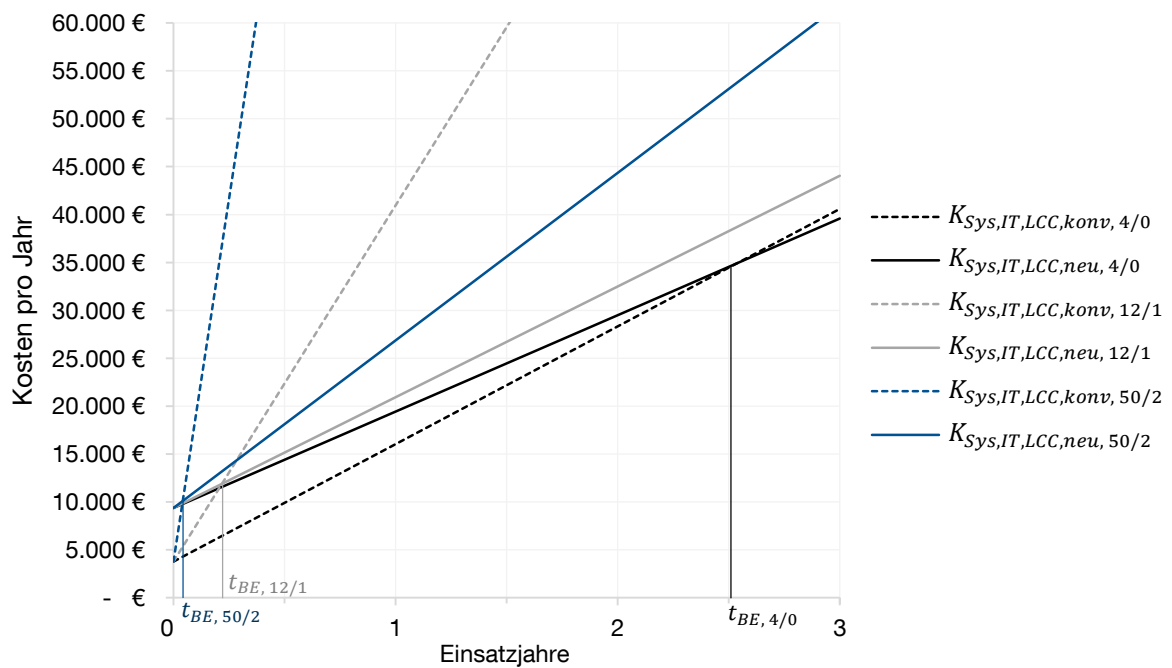


Abbildung 8.15: Vergleich der herkömmlichen Rekonfigurationskosten mit dem vorgestellten Konzept. Dargestellt sind die drei Break-Even-Punkte t_{BE} .

Tabelle 8.1: Kostentabelle auf Basis der getroffenen Annahmen für ausgewählte Rekonfigurations-szenarien im Anwendungsfall

Rekonf./ Ausfälle	Ansatz	Invest	Jahr 1	Jahr 2	Jahr 3	Jahr 4	Jahr 5
4/0	$K_{sys,IT,LCC,konv}$	3.750 €	16.031 €	28.313 €	40.594 €	52.875 €	65.156 €
	$K_{sys,IT,LCC,neu}$	9.356 €	19.426 €	29.497 €	39.568 €	49.639 €	59.709 €
12/1	$K_{sys,IT,LCC,konv}$	3.750 €	40.958 €	78.166 €	115.374 €	152.582 €	189.790 €
	$K_{sys,IT,LCC,neu}$	9.356 €	20.926 €	32.496 €	44.066 €	55.636 €	67.206 €
50/2	$K_{sys,IT,LCC,konv}$	3.750 €	155.182 €	306.614 €	458.046 €	609.478 €	760.909 €
	$K_{sys,IT,LCC,neu}$	9.356 €	26.859 €	44.363 €	61.866 €	79.370 €	96.873 €

8.3 Weitere Potentiale der Lösung

Für Unternehmen, die das erstellte Konzept zur Rekonfiguration nutzen, ergeben sich mehrere Vorteile: Im Fall einer geplanten Anlagenadaption reduziert das Konzept die erforderliche Zeit und das Expertenwissen zur Systemintegration von Ressourcen (vgl. Abschnitt 8.2.1). Damit verringern sich Rekonfigurationsaufwände, wodurch Unternehmen ihre Produktionsmittel leichter an geänderte Randbedingungen anpassen können.

Dies steigert die Produktivität und erlaubt es, in einem turbulenten Marktumfeld besser zu bestehen.

Ähnliche Vorteile ergeben sich bei einer ungeplanten Rekonfiguration, z. B. einem Geräteausfall. Über den synchronen Digital Twin kann eine entsprechender Ausfall detektiert werden, inklusive einer Lokalisierung des Geräts innerhalb der informationstechnologischen Topologie. Der Digital Twin erlaubt den Download der zugehörigen Dokumentation, über die Konfigurationsdienste der Ressourcen wird eine Möglichkeit zum Self-Service gegeben, wodurch insgesamt eine Fehlerbehebung effektiver ausgeführt wird. Durch den fähigkeitsbasierten Ansatz können andere verfügbare Komponenten mit gleichen Fähigkeiten ermittelt und temporär oder dauerhaft als Ersatz verwendet werden. Existieren bereits geeignete Ressourcenalternativen in der Anlage, erlaubt die automatisierte Eignungsprüfung und Materialflussplanung eine entsprechende Umsteuerung. Durch all diese Aspekte sinkt die notwendige Zeit zur Wiederherstellung des Montagesystems (Mean-Time-To-Repair, MTTR), wodurch die technische Verfügbarkeit der Anlage steigt (vgl. LOTTER & WIENDAHL 2012, S. 339 ff.).

Die Zuweisung geeigneter Ressourcen ist dabei nicht auf Automatisierungskomponenten beschränkt. Auch Arbeitskräfte können über Smart Devices (z. B. Tablets) als Ressourcen mit Fähigkeiten eingeplant werden³. Hiermit lassen sich Konzepte einer stufenweisen Automatisierung des Montageprozesses realisieren, je nach Anforderung können über das Framework Werker oder technische Ressourcen eingeplant werden. Auch bei einem Geräteausfall kann der Mensch als flexibler Springer agieren und die entsprechenden Prozessschritte kurzfristig übernehmen.

Zusätzlich zu Ressourcenanpassungen lässt sich das Planungsframework auch für produktbezogene Aussagen verwenden. Soll bspw. eine Produktänderung eingeführt werden, erlaubt das Matchmaking festzustellen, ob die Anpassung noch immer auf den aktuellen Produktionsmitteln montierbar ist. Ist dies nicht der Fall, sind die verantwortlichen Prozessschritte und deren fehlgeschlagene Anforderungen für den Nutzer klar erkennbar. Somit kann eine geeignete Ressource beschafft oder das Produkt entsprechend angepasst werden.

³Auf Shopfloor-Ebene existieren heute bereits primitive analoge Merkmalslisten, welche bspw. die schweißtechnischen Fähigkeiten von Mitarbeitern für eine entsprechende Schichtplanung beschreiben.

Neben Endanwendern birgt das Konzept auch für *Systemintegratoren* Vorteile. Die semantische Standardisierung und Hardwareabstraktion reduziert technische Unwägbarkeiten, wodurch eine präzisere Angebotserstellung ermöglicht wird. Die hohe Interoperabilität des Ansatzes erlaubt eine bessere Wiederverwendbarkeit von erstellten Kundenapplikationen. Beide Aspekte senken die internen Kosten, wodurch Systemintegratoren größere Gewinnmargen erzielen oder wettbewerbsfähiger anbieten können.

Für *Hersteller* von Automatisierungsgeräten ergeben sich mit dem vorgestellten Konzept folgende Synergieeffekte: Zusätzlich zur eigentlichen Hardware können der für eine automatisierte Vernetzung zugrunde liegenden Treiberbaustein und Digital Twin als digitale Dienstleistung verkauft werden. Durch die geschaffene Möglichkeit, Ressourcen leichter austauschen zu können, werden Eintrittsbarrieren für innovative Hersteller in gesättigte Märkte reduziert. Schließlich können auch neue Kundenkreise erschlossen werden, wenn das Planungsframework Anwendern Ressourcen vorschlägt, die bisher noch nicht installiert waren, aber eine (bessere) Eignung für einzelne Prozessschritte bieten als bisherige Komponenten. Dies kann beispielsweise über eine Online-Plattform erfolgen. Über diese liese sich außerdem analysieren, welche Prozessanforderungen von Anwendern häufig gefordert werden und gezielt hierfür entsprechende Ressourcen entwickeln.

Generell abstrahiert das Konzept allgemeine technologische oder herstellereigentliche Merkmale. Hierdurch verringern sich für den Mitarbeiter monotone und nur gering wertschöpfende Aufgaben, wie etwa die Suche nach Werten in Datenblättern. Der Anwender kann sich somit mehr auf höherwertige Aufgaben der Entscheidung und Gestaltung konzentrieren, wodurch die *Arbeit im Produktionsumfeld* aufgewertet wird. Diesen Umstand sieht WGP 2016 (S. 9) als wichtige Zielstellung von Industrie 4.0.

Die automatisierte Anbindung von Feldgeräten sowie die synchronisierte und einheitliche Publikation des Digital Twins über OPC UA, schaffen darüber hinaus eine *vertikale Vernetzung*, die es Unternehmen einfacher als bisher erlaubt, Produktions- und Ressourcendaten im großen Maßstab zu verarbeiten, bspw. in Cloud-Anwendungen. Hierdurch können neue Produktivitätspotentiale auch jenseits der Rekonfiguration erzielt werden (Data Analytics), zusätzlich lassen sich Stillstandszeiten durch Ressourcenausfälle verringern (Predictive Maintenance).

Weiterhin bietet der Dreiklang aus automatisierter vertikaler Vernetzung, standardisierter Fähigkeitenbeschreibung und Auswahlframework die Möglichkeit, dass Unter-

nehmen ihre Produktionsmöglichkeiten einheitlich und global anbieten können. Dies schafft die Grundlage für sogenannte *Produktionsplattformen* (Production-as-a-Service), bei denen Kunden online das zu fertigende Produkt angeben und (teil-)automatisiert Produktionsdienstleister mit geeigneten Ressourcen mit der Herstellung beauftragt werden.

9 Zusammenfassung

In der Montage lässt sich die Produktivität durch Automatisierung nur steigern, wenn Montagesysteme leichter als bisher auf geänderte Randbedingungen adaptierbar sind. Für jede dieser Rekonfigurationen sind dabei langwierige, voneinander abhängige Planungs- und Engineeringaktivitäten durchzuführen. Die vorliegende Arbeit adressiert dies, indem mit einem holistischen Konzept Aufwände des Systementwurfs und der Systemintegration verringert werden. Als Lösungsansatz werden Standardfeldgeräte befähigt, die Eigenschaften Cyber-physischer Systeme (*Selbstvernetzung*, *-beschreibung* und *-konfiguration*) zu erreichen:

Über ein Plug&Produce-Konzept lassen sich angeschlossene Feldgeräte in heterogenen Netzwerken automatisiert erkennen und echtzeitfähig in übergeordnete Steuerungssysteme integrieren. Dies ist auch für Gerätehierarchien und primitive Geräte möglich. Pro Gerät wird eine Selbstbeschreibung (Digital Twin) publiziert, welche als Kern die produktionstechnischen Fähigkeiten enthält.

Fähigkeiten werden genutzt, um Montageprozesse lösungsneutral zu beschreiben. Sie stellen somit ein Bindeglied zwischen der Prozess- und Ressourcendomäne dar und befähigen eine aufgabenorientierte Ressourcenauswahl und Programmierung. Aus diesem Grund wurde eine umfangreiche Herleitung und Klassifizierung von Fähigkeiten und deren Parameter in der Montage gegeben. Weiterhin wurde ein Prozessmodell definiert, welches die Beschreibung von Montageabläufen auf Grundlage der ermittelten Fähigkeitsklassen erlaubt.

Für die komplexe Aktivität der Ressourcenauswahl wurde ein Software-Framework entwickelt, das eine iterative Lösung heterogener Probleme aus dem Engineeringbereich ermöglicht. Als Grundlage dient eine Blackboard-Architektur, welche um die Lösung hierarchischer Teilprobleme erweitert wurde. Hierfür wurden sowohl auftretende Problemklassen in der Montage als auch die Struktur und Interaktion von Experten zu

deren Lösung definiert. Das Ergebnis ermöglicht eine freie Modellierung von Montageprozessen sowie eine automatisiert bestimmte Auswahl möglicher Ressourcen und Programmierung derer.

Durch die beschriebenen Konzepte kann bei Rekonfigurationen effektiv ermittelt werden, ob geeignete Produktionsmittel zur Verfügung stehen und im positiven Fall dann eine hardware- und softwareseitige Anpassung effizient geschehen. Hierbei stellen die entwickelten Konzepte teilautomatisierte Assistenzsysteme für den Anwender dar, welche die Aktivitäten des Systementwurfs und der Systemintegration automatisieren oder durch Informationsverdichtung eine schnellere Entscheidungsfindung liefern. Als Folge kann die erforderliche Rekonfigurationszeit reduziert werden. Gleichzeitig steigert der fähigkeitenbasierte Ansatz zur Prozessmodellierung die Flexibilität in Bezug auf die umzusetzenden Produktionsmittel.

Diese Aspekte tragen dazu bei, das Industrie-4.0-Zielbild einer wandlungsfähigen Produktion für geringe Losgrößen ein wenig mehr Realität werden zu lassen.

Literatur

ABELE & REINHART 2011

Abele, E.; Reinhart, G.: Zukunft der Produktion. Herausforderungen, Forschungsfelder, Chancen. München: Hanser, 2011. ISBN: 978-3-446-42595-8.

ABTS & MÜLDER 2017

Abts, D.; Müller, W.: Grundkurs Wirtschaftsinformatik. Wiesbaden: Springer Fachmedien. 2017. ISBN: 978-3-658-16378-5. DOI: 10.1007/978-3-658-16379-2.

ACATECH 2011

acatech: Cyber-Physical Systems. Innovationsmotoren für Mobilität, Gesundheit, Energie und Produktion. Acatech POSITION. Dordrecht: Springer, 2011. ISBN: 978-3-642-27567-8.

ANDERSEN ET AL. 2016

Andersen, R. S.; Schou, C.; Damgaard, J. S.; Madsen, O.: Using a Flexible Skill-Based Approach to Recognize Objects in Industrial Scenarios. In: Robotics in the era of digitalisation. 47th International Symposium on Robotics. International Symposium on Robotics (ISR) München. Berlin & Offenbach: VDE Verlag. 2016, S. 399–406. ISBN: 978-3-8007-4231-8.

ANDREN ET AL. 2012

Andren, F.; Strasser, T.; Zoitl, A.; Hegny, I.: A reconfigurable communication gateway for distributed embedded control systems. In: IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society Montreal, QC, Canada. IEEE. 2012, S. 3720–3726. ISBN: 978-1-4673-2421-2. DOI: 10.1109/IECON.2012.6389299.

ARAI ET AL. 2001

Arai, T.; Aiyama, Y.; Sugi, M.; Ota, J.: Holonic assembly system with Plug and Produce. Computers in Industry 46 (2001) (3), S. 289–299. DOI: 10.1016/S0166-3615(01)00111-7.

ARAI ET AL. 2003

Arai, T.; Izawa, H.; Maeda, Y.; Kikuchi, H.; Ogawa, H.; Sugi, M.: Real-time task decomposition and allocation for a multi-agent robotic assembly cell. In: Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2003 Besancon, France. IEEE. 2003, S. 42–47. ISBN: 0-7803-7770-2. DOI: 10.1109/ISATP.2003.1217185.

AUTOMATIONSPRAXIS 2013

Automationspraxis: Der Produktionsstandort Deutschland lebt - Automatisierung hält die Fertigung konkurrenzfähig. 2013. URL: <https://automationspraxis.industrie.de/news/der-produktionsstandort-deutschland-lebt/> (Letzter Zugriff am 07.07.2018).

BACKHAUS & REINHART 2015

Backhaus, J.; Reinhart, G.: Digital description of products, processes and resources for task-oriented programming of assembly systems. Journal of Intelligent Manufacturing (2015). DOI: 10.1007/s10845-015-1063-3.

BACKHAUS 2016

Backhaus, J.: Adaptierbares, aufgabenorientiertes Programmiersystem für Montagesysteme. Dissertation. Technische Universität München (TUM), 2016. München. URL: <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20160126-1271525-1-5> (Letzter Zugriff am 21.02.2018).

BAUERNHANSL ET AL. 2014

Bauernhansl, T.; Hompel, M.; Vogel-Heuser, B.: Industrie 4.0 in Produktion, Automatisierung und Logistik. Wiesbaden: Springer Fachmedien, 2014. ISBN: 978-3-658-04681-1. DOI: 10.1007/978-3-658-04682-8.

BENGEL 2008

Bengel, M.: Modelling objects for skill-based reconfigurable machines. In: Innovative production machines and systems. Third I*PROMS Virtual International Conference, 2nd - 13th July, 2007. I*PROMS Virtual International Conference; IPROMS; Virtual International Conference on Innovative Production Machines and Systems. Hrsg. von E. E. Eldukhri; D. T. Pham; A. J. Soroka. Dunbeath: Whittles Publ. 2008, S. 238–243. ISBN: 978-1-420-07347-8.

BENGEL 2009

Bengel, M.: Model-based Configuration – A Workpiece-centred Approach. In: Proceedings of the 2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots. ReMAR 2009 London. Piscataway, N.J.: IEEE. 2009, S. 689–695. ISBN: 1-876346-58-2.

BENGEL 2010

Bengel, M.: Workpiece-centered approach to reconfiguration in manufacturing engineering. Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik. Dissertation. Universität Stuttgart, 2010. Stuttgart.

BIRKHOFER 2011

Birkhofer, R.: Modellbasierte Beschreibung zur offenen Integration intelligenter Feldgeräte der Automatisierungstechnik. Lehrstuhl für Informationstechnik im Maschinenwesen. Dissertation. Technische Universität München (TUM), 2011. München.

BISWAS ET AL. 2013

Biswas; Siddhartha; Alam, B.; Doja M.N.: Generalization of Dijkstra's Algorithm for Extraction of Shortest Paths in Directed Multigraphs. Journal of Computer Science 9 (2013) (3), S. 377–382. DOI: 10.3844/jcssp.2013.377.382.

BITKOM 2015

Bitkom: Industrie 4.0 – Volkswirtschaftliches Potenzial für Deutschland. 2015.

BLESSING & CHAKRABARTI 2009

Blessing, L. T.; Chakrabarti, A.: DRM, a Design Research Methodology. London: Springer London, 2009. ISBN: 978-1-84882-586-4. DOI: 10.1007/978-1-84882-587-1.

BOHN ET AL. 2006

Bohn, H.; Bobek, A.; Golatowski, F.: SIRENA - Service Infrastructure for Real-time Embedded Networked Devices. A service oriented framework for different domains. In: International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06) Morne, Mauritius. IEEE. 2006, S. 43. ISBN: 0-7695-2552-0. DOI: 10.1109/ICNICONSMCL.2006.196.

BRECHER ET AL. 2004

Brecher, C.; Schröter, B.; Almeida, C. d. O.; Dai, F.; Matthias, B.; Kock, S.: Intuitiv

bedienbare Programmiersysteme zur effizienten Programmierung von Handhabungsaufgaben. In: Robotik 2004 : Leistungsstand - Anwendungen - Visionen - Trends; Tagung München, 17. und 18. Juni 2004 / VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik. Bd. 1841. VDI-Berichte. Düsseldorf: VDI-Verl. 2004, S. 303–310. URL: <http://publications.rwth-aachen.de/record/113897>.

BRECHER ET AL. 2010

Brecher, C.; Herfs, W.; Kolster, D.; Pleßow, M.; Jensen, S.: Feldgeräte heute - Geräteintegration und Benutzerschnittstellen. A&D (2010), S. 44–45. URL: http://www.emsa-visual.de/de/7151fdb5fc4f8a0dc125730c0027e55b/emsa_umfrage.pdf (Letzter Zugriff am 28. 02. 2017).

BRONNER 2003

Bronner, A.: Handbuch der Rationalisierung. 2., neu bearb. Aufl. Bd. 331. Renningen: expert Verlag, 2003. ISBN: 3-8169-2036-5.

BROY 2010

Broy, M.: Cyber-physical Systems. Innovation durch softwareintensive eingebettete Systeme. Bd. 0. Acatech DISKUTIERT. Berlin: Springer, 2010. ISBN: 978-3-642-14498-1.

BULLINGER 1986

Bullinger, H.-J.: Systematische Montageplanung. Handbuch für d. Praxis. Gibts im Ordner auch ohne OCR. München & Wien: Hanser. 1986. ISBN: 3446146067.

BMBF 2015

Bundesministerium für Bildung und Forschung: Bekanntmachung: 5G: Industrielles Internet. 2015. URL: <https://www.forschung-it-sicherheit-kommunikationssysteme.de/foerderung/bekanntmachungen/5g-industrielles-internet> (Letzter Zugriff am 25. 01. 2018).

BÜSCHER ET AL. 2013

Büscher, C.; Kuz, S.; Ewert, D.; Schilberg, D.; Jeschke, S.: Kognitive Planungs- und Lernmechanismen in selbstoptimierenden Montagesystemen. In: Jeschke, S.; Isenhardt, I.; Hees, F.; Henning, K. (Hrsg.): Automation, Communication and Cybernetics in Science and Engineering 2011/2012. Berlin, Heidelberg: Springer, 2013, S. 595–605, ISBN: 978-3-642-33388-0. DOI: 10.1007/978-3-642-33389-7_47.

BUSCHMANN ET AL. 1996

Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M.: Pattern-Oriented Software Architecture, A System of Patterns. 1. Aufl. Wiley Software Patterns Series. Chicester: Wiley, 1996. ISBN: 0471958697.

CIRP 2015

Butala, P.; Mporfu, K.: Assembly Systems. In: The International Academy for Production Engineering; Laperrière, L.; Reinhart, G. (Hrsg.): CIRP Encyclopedia of Production Engineering. Berlin, Heidelberg: Springer, 2014, S. 1–5, ISBN: 978-3-642-35950-7. DOI: 10.1007/978-3-642-35950-7_16789-1.

CAVIN & LOHSE 2014

Cavin, S.; Lohse, N.: Multi-level skill-based allocation methodology for evolvable assembly systems. In: 2014 12th IEEE International Conference on Industrial Informatics (INDIN) Porto Alegre RS, Brazil. IEEE. 2014, S. 532–537. ISBN: 978-1-4799-4905-2. DOI: 10.1109/INDIN.2014.6945569.

CAVIN ET AL. 2013

Cavin, S.; Ferreira, P.; Lohse, N.: Dynamic skill allocation methodology for evolvable assembly systems. In: 2013 11th IEEE International Conference on Industrial Informatics (INDIN) Bochum, Germany. IEEE. 2013, S. 218–223. ISBN: 978-1-4799-0752-6. DOI: 10.1109/INDIN.2013.6622885.

CIRP 2011

Wörterbuch der Fertigungstechnik Bd. 4 (2011). 1. Auflage. Berlin u. a.: Springer, 2011. ISBN: 978-3-642-12006-0. DOI: 10.1007/978-3-642-12007-7.

DAENZER & HABERFELLNER 2002

Daenzer, W. F.; Haberfellner, R.: Systems engineering. Methodik und Praxis. 11., durchges. Aufl. Zürich: Verl. Industrielle Organisation, 2002. ISBN: 9783857439988.

DEMANT ET AL. 2011

Demant, C.; Streicher-Abel, B.; Springhoff, A.: Industrielle Bildverarbeitung. Wie optische Qualitätskontrolle wirklich funktioniert. Berlin, Heidelberg: Springer, 2011. ISBN: 978-3642130960.

DIN 1319-1

Deutsches Institut für Normung e.V.: Grundlagen der Meßtechnik. DIN. ICS 17.020; 01.040.17. Berlin: Beuth. 1995.

DIN 8580

Deutsches Institut für Normung e.V.: Fertigungsverfahren. DIN. ICS 01.040.25; 25.020. 2003a.

DIN 8593

Deutsches Institut für Normung e.V.: Fertigungsverfahren Fügen. DIN. ICS 01.040.25; 25.020; 25.120.99. 2003b.

DIN 61131-3

Deutsches Institut für Normung e.V.: Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen. DIN. ICS 25.040.40; 35.060; 35.240.50. Berlin: Beuth. 2003c.

DIN 62390

Deutsches Institut für Normung e.V.: Leitfaden für Geräteprofile in der Automatisierungstechnik. Fachbericht. ICS 35.240.50. Berlin: Beuth. 2006.

DIN 4002-1

Deutsches Institut für Normung e.V.: Merkmale und Geltungsbereiche zum Produktdatenaustausch. DIN. ICS 35.240.60. Berlin: Beuth. 2007.

IEC 62541

Deutsches Institut für Normung e.V.: OPC Unified Architecture. IEC. ICS 35.200; 35.240.50. Berlin: Beuth. 2010.

DIN EN 61499

Deutsches Institut für Normung e.V.: Verteilte Funktionsbausteine für die Automatisierungstechnik. DIN EN. ICS 35.240.50. Berlin: Beuth. 2012.

IEC 60050

Deutsches Institut für Normung e.V.: Internationales elektrotechnisches Wörterbuch. DIN IEC. ICS 01.040.35; 01.040.29; 35.240.50; 29.020. Berlin: Beuth. 2014.

DRATH 2010

Drath, R.: Datenaustausch in der Anlagenplanung mit AutomationML. Integration

von CAEX, PLCopen XML und COLLADA. Berlin, Heidelberg: Springer, 2010. ISBN: 978-3-642-04673-5. DOI: 10.1007/978-3-642-04674-2.

DÜRKOP ET AL. 2012

Dürkop, L.; Imtiaz, J.; Trsek, H.: Service-Oriented Architecture for the Autoconfiguration of Real-Time Ethernet Systems. Lemgo: Inst. für Industrielle Informationstechnik, 2012. ISBN: 978-3-9814062-2-1.

DÜRKOP ET AL. 2014

Dürkop, L.; Trsek, H.; Otto, J.; Jasperneite, J.: A field level architecture for reconfigurable real-time automation systems. In: 2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014) Toulouse, France. IEEE. 2014, S. 1–10. ISBN: 978-1-4799-3235-1. DOI: 10.1109/WFCS.2014.6837601.

eCl@ss 2015

eCl@ss e.V.: eCl@ss-Whitepaper Industrie 4.0. 2015. URL: https://www.eclass.eu/fileadmin/downloads/ecl_Whitepaper-Industrie-de.pdf-de.pdf&usg=AOvVaw1Ory-3U1nr2LRygrIcLi0 (Letzter Zugriff am 16. 02. 2018).

EIGNER & STELZER 2009

Eigner, M.; Stelzer, R.: Product Lifecycle Management. Ein Leitfaden für Product Development und Life Cycle Management. Berlin, Heidelberg: Springer, 2009. ISBN: 978-3-540-68401-5.

EILERS 2015

Eilers, J.: Methodik zur Planung skalierbarer und rekonfigurierbarer Montagesysteme. 1. Aufl. Bd. Band 9/2015. Fertigungsmesstechnik & Qualitätsmanagement. Aachen: Apprimus-Verl., 2015. ISBN: 9783863592950.

ELMARAGHY & ELMARAGHY 2016

ElMaraghy, H.; ElMaraghy, W.: Smart Adaptable Assembly Systems. Procedia CIRP 44 (2016), S. 4–13. DOI: 10.1016/j.procir.2016.04.107.

ELMARAGHY 2006

ElMaraghy, H. A.: Flexible and reconfigurable manufacturing systems paradigms. International Journal of Flexible Manufacturing Systems 17 (2006) (4), S. 261–276. DOI: 10.1007/s10696-006-9028-7.

ELMARAGHY 2009

ElMaraghy, H. A.: Changeable and Reconfigurable Manufacturing Systems. Springer series in advanced manufacturing. London: Springer London, 2009. ISBN: 978-1-84882-066-1.

ENSTE & MAHNKE 2011

Enste, U.; Mahnke, W.: OPC Unified Architecture. at - Automatisierungstechnik 59 (2011) (7), S. 397–404. DOI: 10.1524/auto.2011.0934.

EPSG 2016

Ethernet Powerlink Standardization Group: Industrial Ethernet Facts. Systemvergleich. 2016. URL: <http://www.ethernet-powerlink.org/de/downloads/industrial-ethernet-facts/>.

EVERSHEIM 1983

Eversheim, W.: Ein Baukastensystem für die Montage konzipieren. Industrie-Anzeiger 105 (1983) (92).

EVERSHEIM 2002

Eversheim, W.: Organisation in der Produktionstechnik - Arbeitsvorbereitung. Berlin, Heidelberg: Springer, 2002. ISBN: 978-3-642-62640-1. DOI: 10.1007/978-3-642-56336-2.

EVERTZ & EPPLE 2014

Evertz, L.; Epple, U.: Vergleich von SOA-Ansätzen unter den Gesichtspunkten des OASIS-Referenzmodells. In: Automation 2014. 15. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verl., 2014, ISBN: 978-3-18-092231-7.

FELDMANN & SPUR 2014

Feldmann, K.; Spur, G.: Handbuch Fügen, Handhaben, Montieren. München: Hanser. 2014.

FISCHER 1975

Fischer, L.: Steuersystem und betriebliche Rationalisierung. Wiesbaden: Gabler Verlag, 1975. ISBN: 978-3-322-97925-4. DOI: 10.1007/978-3-322-98469-2.

FRIEDL 2014

Friedl, A.: Planung, Projektierung und Programmierung der Automatisierungstechnik.

In: Feldmann, K.; Spur, G. (Hrsg.): Handbuch Fügen, Handhaben, Montieren. München: Hanser, 2014, S. 622–644.

FRIEDRICH 2010

Friedrich, T.: Technologieorientiertes Programmier- und Steuerungssystem für Industrieroboter. Dissertation. Technische Universität Berlin, 2010.

FRÜH & AHRENS 2009

Früh, K. F.; Ahrens, W.: Handbuch der Prozessautomatisierung. Prozessleittechnik für verfahrenstechnische Anlagen. 4., überarb. Aufl. München: Oldenbourg, 2009. ISBN: 978-3-8356-3142-7.

GERDES ET AL. 2004

Gerdes, I.; Klawonn, F.; Kruse, R.: Evolutionäre Algorithmen. Wiesbaden: Vieweg+Teubner Verlag, 2004. ISBN: 978-3-528-05570-7. DOI: 10.1007/978-3-322-86839-8.

GORECKY ET AL. 2017

Gorecky, D.; Hennecke, A.; Schmitt, M.; Weyer, S.; Zühlke, D.: Wandelbare modulare Automatisierungssysteme. In: Reinhart, G. (Hrsg.): Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. München: Hanser, 2017, S. 555–583, ISBN: 978-3-446-44642-7.

GRAHAM 2015

Graham, D. W.: Heraclitus. In: Zalta, E. N. (Hrsg.): The Stanford Encyclopedia of Philosophy. Fall 2015. Metaphysics Research Lab, Stanford University, 2015. URL: <https://plato.stanford.edu/archives/fall2015/entries/heraclitus/> (Letzter Zugriff am 18. 01. 2017).

GRANZOW-EMDEN & LUBER 2014

Granzow-Emden, M.; Luber, J.: Deutsche Grammatik verstehen und unterrichten. 2., überarb. Aufl. bachelor-wissen. Tübingen: Narr, 2014. ISBN: 978-3-8233-6883-0.

GRIFFEN 2014

Griffen, B.: PackML. B&R automotion 13 (2014) (05).

GROSSMANN 2008

Großmann, D.: Offene Integrationsplattform für das Feldgeräte-Management. 1. Aufl. Informationstechnik im Maschinenwesen. Göttingen: Sierke, 2008. ISBN: 9783868440706.

GRUNWALD ET AL. 2008

Grunwald, G.; Plank, G.; Reintsema, D.; Zimmermann, U. E.; Bischoff, R.: Communication, Configuration, Application. The three layer concept for Plug-and-Produce. In: ICINCO 2008. Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics. 2008. URL: <http://elib.dlr.de/55498/>.

HAMMERSTINGL & REINHART 2015a

Hammerstingl, V.; Reinhart, G.: Umfrage zu Technologien und Standards im Bereich Montagetechnik 2015. 2015a. URL: <http://mediatum.ub.tum.de/?id=1342279> (Letzter Zugriff am 30. 03. 2018).

HAMMERSTINGL & REINHART 2015b

Hammerstingl, V.; Reinhart, G.: Unified Plug&Produce architecture for automatic integration of field devices in industrial environments. In: 2015 IEEE International Conference on Industrial Technology (ICIT) Seville. 2015, S. 1956–1963. DOI: 10.1109/ICIT.2015.7125383.

HAMMERSTINGL & REINHART 2017

Hammerstingl, V.; Reinhart, G.: Task-oriented Programming of Assembly Systems based on self-describing Components. In: 16th International Conference on Computer Aided Systems Theory (EUROCAST 2017) - Extended Abstracts. Las Palmas, Spain. Hrsg. von Alexis Quesada-Arencibia; José Carlos Rodríguez; Roberto Moreno-Díaz; Roberto jr. Moreno-Díaz. 2017, S. 172–173.

HAMMERSTINGL & REINHART 2018

Hammerstingl, V.; Reinhart, G.: Skills in Assembly. 2018. URL: <https://mediatum.ub.tum.de/1428286> (Letzter Zugriff am 19. 01. 2018).

HAMMERSTINGL ET AL. 2015

Hammerstingl, V.; Moule, L.; Reinhart, G.: Bildverarbeitungssysteme als cyber-physische Sensoren. Wenn das Produkt seine Wünsche kommuniziert. atp edition 57 (2015) (11), S. 44–57. URL: <https://www.di-verlag.de/de/Zeitschriften/atp-edition/2015/11/Bildverarbeitungssysteme-als-cyber-physische-Sensoren> (Letzter Zugriff am 20. 02. 2018).

HAMMERSTINGL ET AL. 2016

Hammerstingl, V.; Reinhart, G.; Zimmermann, P.: Automatisierte Konfigurati-

on und Selbstauskunft von Industrierobotern. Intelligente Vernetzung mittels ROS und OPC UA. *Industrie 4.0 Management* 32 (2016) (6), S.17–21. URL: https://www.researchgate.net/publication/312471476_Automatisierte_Konfiguration_und_Selbstauskunft_von_Industrierobotern.

HEINRICH ET AL. 2015

Heinrich, B.; Linke, P.; Glöckler, M.: *Grundlagen Automatisierung*. Wiesbaden: Springer Fachmedien, 2015. ISBN: 978-3-658-05960-6. DOI: 10.1007/978-3-658-05961-3.

HEPP ET AL. 2007

Hepp, M.; Leukel, J.; Schmitz, V.: A quantitative analysis of product categorization standards. Content, coverage, and maintenance of eCl@ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary. *Knowledge and Information Systems* 13 (2007) (1), S. 77–114. DOI: 10.1007/s10115-006-0054-2.

HERING & SCHÖNFELDER 2012

Hering, E.; Schönfelder, G.: *Sensoren in Wissenschaft und Technik. Funktionsweise und Einsatzgebiete*. Wiesbaden: Vieweg+Teubner Verlag, 2012. ISBN: 978-3-8348-8635-4.

HERKOMMER 2017

Herkommer, G.: *Die Marktanteile industrieller Netzwerke in 2017*. Hrsg. von WEKA FACHMEDIEN GmbH. 2017. URL: <http://www.computer-automation.de/feldebene/vernetzung/artikel/139254/> (Letzter Zugriff am 01. 11. 2017).

HESSE & MALISA 2009

Hesse, S.; Malisa, V.: *Taschenbuch Robotik - Montage - Handhabung*. München: Hanser, Carl, 2009. ISBN: 978-3-446-41969-8.

HESSE & SCHNELL 2014

Hesse, S.; Schnell, G.: *Sensoren für die Prozess- und Fabrikautomation*. Wiesbaden: Springer Fachmedien, 2014. ISBN: 978-3-658-05866-1. DOI: 10.1007/978-3-658-05867-8.

HODEK 2013

Hodek, S.: *Methode zur vollautomatischen Integration von Feldgeräten in industrielle Steuerungssysteme. Ein Beitrag zur Plug&Play-Feldgeräteintegration*. Bd. 26 : Automatisierungstechnik. Fortschritt-Berichte pak. Kaiserslautern: Techn. Univ., 2013. ISBN: 978-3-943995-35-0.

HU ET AL. 2011

Hu, S. J.; Ko, J.; Weyand, L.; ElMaraghy, H. A.; Lien, T. K.; Koren, Y.; Bley, H.; Chrysosouris, G.; Nasr, N.; Shpitalni, M.: Assembly system design and operations for product variety. *CIRP Annals - Manufacturing Technology* 60 (2011) (2), S. 715–733. DOI: 10.1016/j.cirp.2011.05.004.

HUBER 2016

Huber, W.: *Industrie 4.0 in der Automobilproduktion*. Wiesbaden: Springer Fachmedien, 2016. ISBN: 978-3-658-12731-2. DOI: 10.1007/978-3-658-12732-9.

HUCKABY & CHRISTENSEN 2012

Huckaby, J.; Christensen, H. I.: A Taxonomic Framework for Task Modeling and Knowledge Transfer in Manufacturing Robotics. In: *Cognitive robotics. Papers from the 2012 AAI Workshop*. Technical report / American Association for Artificial Intelligence WS 2012,6. Palo Alto, Calif.: AAI Press, 2012, S. 94–101, ISBN: 978-157735571-7.

HUCKABY ET AL. 2013

Huckaby, J.; Vassos, S.; Christensen, H. I.: Planning with a task modeling framework in manufacturing robotics. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems Tokyo*. IEEE. 2013, S. 5787–5794. ISBN: 978-1-4673-6358-7. DOI: 10.1109/IROS.2013.6697194.

HUMBURGER 1998

Humburger, R.: *Konzeption eines Systems zur aufgabenorientierten Roboterprogrammierung*. Bd. 98. *Berichte aus der Produktionstechnik*. Aachen: Shaker, 1998. ISBN: 3-8265-3427-1.

IEC 62714

International Electrotechnical Commission: *Engineering data exchange format for use in industrial automation systems engineering - Part 2: Role class libraries*. Version Edition 1.0. ICS 25.040.40; 35.06; 35.240.50. Geneva: International Electrotechnical Commission. 2015.

IEC 62453

International Electrotechnical Commission: *Field device tool (FDT) interface specification*. IEC. Version 2.0. ICS 25.040.40; 35.100.05; 35.110. Geneva. 2016a. URL: <https://webstore.iec.ch/publication/31126>.

IEC 62424

International Electrotechnical Commission: Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools. IEC. ICS 25.040.40; 35.240.50. Geneva: International Electrotechnical Commission. 2016b.

ISO/IEC 7498

International Organization for Standardization: Open Systems Interconnection Model. ISO/IEC. ICS: 35.100.01. Genf. 1994. URL: <https://www.iso.org/standard/20269.html>.

IWG 2015

Interoperability Working Group: Definition of Interoperability. 2015. URL: <http://interoperability-definition.info/en/>.

JAMMES & SMIT 2005

Jammes, F.; Smit, H.: Service-Oriented Paradigms in Industrial Automation. IEEE Transactions on Industrial Informatics 1 (2005) (1), S. 62–70. DOI: 10.1109/TII.2005.844419.

JÄRVENPÄÄ 2012

Järvenpää, E.: Capability-based Adaptation of Production Systems in a Changing Environment. Dissertation. Tampere University of Technology, 2012. Tampere.

JOHN & TIEGELKAMP 2009

John, K.-H.; Tiegelkamp, M.: SPS-Programmierung mit IEC 61131-3. Konzepte und Programmiersprachen, Anforderungen an Programmiersysteme, Entscheidungshilfen. VDI. Dordrecht & New York: Springer, 2009. ISBN: 978-3-642-00268-7.

KEDDIS ET AL. 2013

Keddis, N.; Kainz, G.; Buckl, C.; Knoll, A.: Towards adaptable manufacturing systems. In: 2013 IEEE International Conference on Industrial Technology (ICIT) Cape Town. IEEE. 2013, S. 1410–1415. ISBN: 978-1-4673-4569-9. DOI: 10.1109/ICIT.2013.6505878.

KEDDIS ET AL. 2014

Keddis, N.; Kainz, G.; Zoitl, A.: Capability-based planning and scheduling for adaptable manufacturing systems. In: 2014 IEEE Emerging Technology and Factory Automation (ETFAs) Barcelona, Spain. 2014. DOI: 10.1109/ETFAs.2014.7005213.

KHRONOS GROUP 2008

Khronos Group: COLLADA – Digital Asset Schema Release 1.5.0. 2008. URL: <https://www.khronos.org/collada/> (Letzter Zugriff am 16. 02. 2018).

KLASEN ET AL. 2010

Klasen, F.; Oestreich, V.; Volz, M.: Industrielle Kommunikation mit Feldbus und Ethernet. Berlin [u.a.]: VDE-Verl, 2010. ISBN: 978-3-8007-3297-5.

KLEIN 2016

Klein, R.: Industrie 4.0 requires more flexible connection solutions. industrial ethernet book 96 (2016) (10), S. 36–37.

KLEINEBERG ET AL. 2016

Kleineberg, O.; Schneider, A.; Hummen, R.: TSN: a definitive kit for demanding automation networks. industrial ethernet book 96 (2016) (10), S. 8–12.

KLUGE 2011

Kluge, S.: Methodik zur fähigkeitsbasierten Planung modularer Montagesysteme. Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik. Dissertation. Universität Stuttgart, 2011.

KOCH 2017

Koch, J.: Manufacturing Change Management – a Process-Based Approach for the Management of Manufacturing Changes. Dissertation. Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb), TUM, 2017. München.

KOCH ET AL. 2015

Koch, J.; Brandl, F.; Hofer, A.; Reinhart, G.: Studie: Änderungsmanagement in der Produktion. Hrsg. von Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb), TUM. München. 2015.

FOCUS 2016

Kollege Klaus, der Roboter (2016). Focus (2016) (40).

KOREN ET AL. 1999

Koren, Y.; Heisel, U.; Jovane, F.; Moriwaki, T.; Pritschow, G.; Ulsoy, G.; van Brussel, H.: Reconfigurable Manufacturing Systems. CIRP Annals - Manufacturing Technology 48 (1999) (2), S. 527–540. DOI: 10.1016/S0007-8506(07)63232-6.

KOREN & SHPITALNI 2010

Koren, Y.; Shpitalni, M.: Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems* 29 (2010) (4), S. 130–141. DOI: 10.1016/j.jmsy.2011.01.001.

KRUG 2013

Krug, S.: Automatische Konfiguration von Robotersystemen (Plug&Produce). Bd. 270. *Forschungsberichte iwb*. München: Utz, 2013. ISBN: 978-3-8316-4243-4.

KRÜGER ET AL. 2017

Krüger, J.; Wang, L.; Verl, A.; Bauernhansl, T.; Carpanzano, E.; Makris, S.; Fleischer, J.; Reinhart, G.; Franke, J.; Pellegrinelli, S.: Innovative control of assembly systems and lines. *CIRP Annals - Manufacturing Technology* 66 (2017) (2), S. 707–730. DOI: 10.1016/j.cirp.2017.05.010.

KÜRSCHNER 2016

Kürschner, W.: *Grammatisches Kompendium. Systematisches Verzeichnis grammatischer Grundbegriffe*. 2. überarb. Aufl. Bd. Nr. 1526. Linguistik. UTB. Tübingen & Tübingen: UTB & Francke, A, 2016. ISBN: 978-3-8252-4693-8.

LANGE ET AL. 2010

Lange, J.; Burke, T. J.; Iwanitz, F.: *OPC. Von Data Access bis Unified Architecture*. 4., völlig neu bearb. und erw. Aufl. Berlin [u.a.]: VDE-Verl., 2010. ISBN: 9783800732173.

LAPERRIÈRE & REINHART 2014

Laperrière, L.; Reinhart, G.: *CIRP Encyclopedia of Production Engineering*. 1., st Edition. Berlin, Heidelberg: Springer, 2014. ISBN: 978-3-642-20616-0.

LEI & WISSE 2016

Lei, Q.; Wisse, M.: Object grasping by combining caging and force closure. In: 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV) Phuket, Thailand. IEEE. 2016, S. 1–8. ISBN: 978-1-5090-3549-6. DOI: 10.1109/ICARCV.2016.7838638.

LEIBER ET AL. 2018

Leiber, D.; Hammerstingl, V.; Reinhart, G.: Automated design of multi-station assembly lines. A knowledge-based approach for selecting and positioning production resources in an assembly line. In: 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering Naples, Italy. 2018.

LIEBHART 2007

Liebhart, D.: SOA goes real. Service-orientierte Architekturen erfolgreich planen und einführen. eibis e-Bib Projekt. München: Hanser, 2007. ISBN: 978-3-446-41088-6.

LOSKYLL 2013

Loskyll, M.: Entwicklung einer Methodik zur dynamischen kontextbasierten Orchestrierung semantischer Feldgerätefunktionalitäten. Bd. 25 : Kontextadaptive Automatisierung. Fortschritt-Berichte pak. Kaiserslautern: Techn. Univ, 2013. ISBN: 978-3-943995-29-9.

LOSKYLL ET AL. 2011

Loskyll, M.; Schlick, J.; Hodek, S.; Ollinger, L.; Gerber, T.; Pirvu, B.: Semantic service discovery and orchestration for manufacturing processes. In: ETFA2011. Factory Automation Toulouse, France. IEEE. 2011, S. 1–8. ISBN: 978-1-4577-0017-0. DOI: 10.1109/ETFA.2011.6058988.

LOSKYLL ET AL. 2012

Loskyll, M.; Heck, I.; Schlick, J.; Schwarz, M.: Context-Based Orchestration for Control of Resource-Efficient Manufacturing Processes. Future Internet 4 (2012) (4), S. 737–761. DOI: 10.3390/fi4030737.

LOTTER & WIENDAHL 2012

Lotter, B.; Wiendahl, H.-P.: Montage in der industriellen Produktion. Ein Handbuch für die Praxis. 2., Aufl. VDI-Buch. Berlin: Springer, 2012. ISBN: 978-3-642-29060-2.

LYDON 2013

Lydon, B.: Industrial Ethernet Growing but Fieldbus Remains Dominant. Hrsg. von Automation.com. 2013. URL: <https://www.automation.com/automation-news/article/industrial-ethernet-growing-but-fieldbus-remains-dominant> (Letzter Zugriff am 01. 11. 2017).

MAHLER & GOLDBERG 2017

Mahler, J.; Goldberg, K. Y.: Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences. In: Proceedings of the 1st Annual Conference on Robot Learning (CoRL) Mountain View, California. Hrsg. von S. Levine; V. Vanhoucke; K. Goldberg. 2017, S. 515–524.

MALEC ET AL. 2013

Malec, J.; Nilsson, K.; Bruyninckx, H.: Describing assembly tasks in declarative way. In: 2013 IEEE International Conference on Robotics and Automation (ICRA 2013) Karlsruhe. 2013.

MASHKOOR & FERNANDES 2007

Mashkooor, A.; Fernandes, J. M.: Deriving Software Architectures for CRUD Applications. The FPL Tower Interface Case Study. In: International Conference on Software Engineering Advances (ICSEA 2007). International Conference on Software Engineering Advances (ICSEA 2007) Cap Esterel, France. IEEE. 2007, S. 25. ISBN: 0-7695-2937-2. DOI: 10.1109/ICSEA.2007.25.

MATHAS 2008

Mathas, C.: SOA intern. München Wien: Hanser, 2008. ISBN: 978-3-446-41189-0.

MAYER ET AL. 2013

Mayer, M.; Odenthal, B.; Ewert, D.; Kempf, T.; Behnen, D.; Büscher, C.; Kuz, S.; Müller, S.; Hauck, E.; Kausch, B.; Schilberg, D.; Herfs, W.; Schlick, C. M.; Jeschke, S.; Brecher, C.: Self-optimising Assembly Systems for Handling Large Components. In: Jeschke, S.; Isenhardt, I.; Hees, F.; Henning, K. (Hrsg.): Automation, Communication and Cybernetics in Science and Engineering 2011/2012. Berlin, Heidelberg: Springer, 2013, S. 681–740, ISBN: 978-3-642-33388-0. DOI: 10.1007/978-3-642-33389-7_53.

METCALFE 1999

Metcalfe, B.: Word origin: Ethernet. 1999. URL: <http://listserv.linguist-list.org/pipermail/ads-l/1999-July/001027.html> (Letzter Zugriff am 20. 10. 2017).

MICHNIEWICZ & REINHART 2014

Michniewicz, J.; Reinhart, G.: Cyber-physical Robotics – Automated Analysis, Programming and Configuration of Robot Cells based on Cyber-physical-systems. *Procedia Technology* 15 (2014), S. 567–576. DOI: 10.1016/j.protcy.2014.09.017.

MICHNIEWICZ & REINHART 2016

Michniewicz, J.; Reinhart, G.: Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. *Mechatronics* 34 (2016), S. 170–180. DOI: 10.1016/j.mechatronics.2015.04.012.

MICHNIEWICZ ET AL. 2016

Michniewicz, J.; Reinhart, G.; Boschert, S.: CAD-Based Automated Assembly Planning for Variable Products in Modular Production Systems. *Procedia CIRP* 44 (2016), S. 44–49. DOI: 10.1016/j.procir.2016.02.016.

MÜLLER ET AL. 2009

Müller, R.; Buchner, T.; Fayzullin, K.; Gottschalk, S.; Herfs, W.; Hilchner, R.; Pyschny, N.: Studie Strategien und Trends in der Montagetechnik und -organisation. Cluster of excellence integrative production technology for high wage countries. Hrsg. von Christian Brecher; Lutz Schapp. *Excellence in materials and production*. Brecher, Christian (Hrsg.) Aachen: Apprimus, 2009. ISBN: 978-3940565273.

MÜLLER ET AL. 2016a

Müller, R.; Vette, M.; Mailahn, O.: Process-oriented Task Assignment for Assembly Processes with Human-robot Interaction. *Procedia CIRP* 44 (2016), S. 210–215. DOI: 10.1016/j.procir.2016.02.080.

MÜLLER ET AL. 2016b

Müller, R.; Hörauf, L.; Vette, M.; Speicher, C.: Planning and Developing Cyber-physical Assembly Systems by Connecting Virtual and Real Worlds. *Procedia CIRP* 52 (2016), S. 35–40. DOI: 10.1016/j.procir.2016.07.050.

MÜLLER ET AL. 2016c

Müller, R.; Vette, M.; Hörauf, L.; Speicher, C.: Identification of Assembly System Configuration for Cyber-Physical Assembly System Planning. *Applied Mechanics and Materials* 840 (2016), S. 24–32. DOI: 10.4028/www.scientific.net/AMM.840.24.

MUNZERT 2010

Munzert, U.: *Bahnplanungsalgorithmen für das robotergestützte Remote-Laserstrahlschweißen*. München: Utz, 2010. ISBN: 978-3-8316-0948-2.

MVTEC GMBH 2018

MVTec GmbH: *HDevelop User's Guide*. HALCON 18.05. 2018.

NAUMANN & VERL 2008

Naumann, M.; Verl, A.: Plug'n'Produce-Steuerungsarchitektur für Roboterzellen. 98 (2008) (5), S. 384–390.

NAUMANN ET AL. 2010

Naumann, M.; Bengel, M.; Verl, A.: Automatic Generation of Robot Applications Using a Knowledge Integration Framework. In: Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK) Munich. German Conference on Robotics. Berlin: VDE. 2010. ISBN: 978-3-8007-3273-9.

NYHUIS ET AL. 2008

Nyhuis, P.; Reinhart, G.; Abele, E.: Wandlungsfähige Produktionssysteme. Heute die Industrie von morgen gestalten. 2008.

OASIS 2006

MacKenzie, C. M.; Laskey, K.; McCabe, F.; Brown, P. F.; Metz, R.: Reference model for service oriented architecture 1.0. Hrsg. von OASIS. 2006. URL: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> (Letzter Zugriff am 10. 02. 2017).

OPC 2013

OPC Foundation: OPC Unified Architecture for Devices. Companion Specification. 2013.

OPC 2017

OPC Foundation: OPC Unified Architecture - Interoperability for Industrie 4.0 and the Internet of Things. 2017. URL: <https://opcfoundation.org/resources/brochures/> (Letzter Zugriff am 25. 01. 2018).

OTTO & NIGGEMANN 2015

Otto, J.; Niggemann, O.: Automatic Parameterization of Automation Software for Plug-and-Produce. In: Proceedings Twenty-Ninth AAAI Conference. Conference on Artificial Intelligence Austin/Texas. 2015. URL: <http://publica.fraunhofer.de/dokumente/N-332493.html>.

PAPAGEORGIU ET AL. 2015

Papageorgiou, M.; Leibold, M.; Buss, M.: Optimierung. Berlin, Heidelberg: Springer, 2015. ISBN: 978-3-662-46935-4. DOI: 10.1007/978-3-662-46936-1.

PAWELLEK 2008

Pawellek, G.: Ganzheitliche Fabrikplanung. Grundlagen, Vorgehensweise, EDV-Unterstützung. VDI. Berlin, Heidelberg: Springer, 2008. ISBN: 978-3-540-78402-9.

PFROMMMER ET AL. 2013

Pfrommer, J.; Schleipen, M.; Beyerer, J.: PPRS. Production skills and their relation to product, process, and resource. In: 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA) Cagliari, Italy. IEEE. 2013, S. 1–4. ISBN: 978-1-4799-0864-6. DOI: 10.1109/ETFA.2013.6648114.

PFROMMMER ET AL. 2014

Pfrommer, J.; Stogl, D.; Aleksandrov, K.; Schubert, V.; Hein, B.: Modelling and orchestration of service-based manufacturing systems via skills. In: 2014 IEEE Emerging Technology and Factory Automation (ETFA) Barcelona, Spain. 2014, S. 1–4. DOI: 10.1109/ETFA.2014.7005285.

PFROMMMER ET AL. 2015

Pfrommer, J.; Stogl, D.; Aleksandrov, K.; Escada Navarro, S.; Hein, B.; Beyerer, J.: Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. at - Automatisierungstechnik 63 (2015) (10). DOI: 10.1515/auto-2014-1157.

PLATON & KRAPINGER 2014

Platon; Krapinger, G.: Platon Kratylos. Bd. Nr. 18994. Reclams Universal-Bibliothek. Krapinger, Gernot (Hrsg.) Stuttgart: Reclam. 2014. ISBN: 9783150189948.

PLATTFORM I4.0 2015

Plattform Industrie 4.0: Whitepaper FuE-Themen. 2015.

PLATTFORM I4.0 2016a

Plattform Industrie 4.0: Forschungsagenda Industrie 4.0 – Aktualisierung des Forschungsbedarfs. 2016a.

PLATTFORM I4.0 2016b

Plattform Industrie 4.0: Interaktionsmodell für Industrie 4.0-Komponenten. 2016b. URL: <http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/interaktionsmodell-i40-komponenten-it-gipfel.html> (Letzter Zugriff am 10. 02. 2017).

PLCOPEN 2009

PLCopen: XML Formats for IEC 61131-3. Technical Paper - PLCopen Technical Committee 6. Version 2.01. 2009. URL: http://www.plcopen.org/pages/tc6_xml/xml_intro/-index.htm (Letzter Zugriff am 16. 10. 2017).

PLCOPEN 2011

PLCopen: Technical Specification - Function blocks for motion control. 2011. URL: http://www.plcopen.org/pages/tc2_motion_control/forms/conf-motioncontrol.htm (Letzter Zugriff am 23. 02. 2018).

PROSTEP 2013

PROSTEP AG: 3D-Formate im Engineering-Umfeld - ein Vergleich. 2013. URL: <https://www.prostep.com/downloads/whitepaper/whitepaper-3d-formate-im-vergleich> (Letzter Zugriff am 16. 02. 2018).

REGULIN ET AL. 2015

Regulin, D.; Glaese, A.; Feldmann, S.; Schutz, D.; Vogel-Heuser, B.: Enabling flexible automation system hardware. Dynamic reconfiguration of a real-time capable field-bus. In: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN) Cambridge, United Kingdom. IEEE. 2015, S. 1198–1205. ISBN: 978-1-4799-6649-3. DOI: 10.1109/INDIN.2015.7281906.

REINHART & KRUG 2012

Reinhart, G.; Krug, S.: Automatic Configuration (Plug & Produce) of Robot Systems – Data-Interpretation and Exchange. In: ElMaraghy, H. A. (Hrsg.): Enabling Manufacturing Competitiveness and Economic Sustainability. Berlin, Heidelberg: Springer, 2012, S. 147–152, ISBN: 978-3-642-23859-8. DOI: 10.1007/978-3-642-23860-4_24.

REINHART & KRUG 2010

Reinhart, G.; Krug, S.: Current State Model for Easy Reconfiguration of Robot Systems and Offline-Programming-Environments. In: Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK) Munich. German Conference on Robotics. Berlin: VDE. 2010. ISBN: 978-3-8007-3273-9.

REINHART ET AL. 2017

Backhaus, J.; Hammerstingl, V.; Michniewicz, J.; Stocker, C.; Ulrich, M.; Reinhart, G.: Montage 4.0. In: Reinhart, G. (Hrsg.): Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. München: Hanser, 2017, S. 513–554, ISBN: 978-3-446-44642-7.

REISSENWEBER 2009

Reißenweber, B.: Feldbussysteme zur industriellen Kommunikation. 3., vollst. überarb. Aufl. München: Oldenbourg-Industrieverl., 2009. ISBN: 9783835631434.

RICHARDS 2015

Richards, M.: Software architecture patterns. Understanding common architecture patterns and when to use them. First edition. Sebastopol, CA: O'Reilly Media, 2015. ISBN: 978-1-491-92424-2.

RIDGE ET AL. 2017

Ridge, B.; Gaspar, T.; Ude, A.: Rapid state machine assembly for modular robot control using meta-scripting, templating and code generation. In: 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids) Birmingham. IEEE. 2017, S. 661–668. ISBN: 978-1-5386-4678-6. DOI: 10.1109/HUMANOIDS.2017.8246943.

ROS 2017

Supported Hardware (2017). 2017. URL: http://wiki.ros.org/Industrial/supported_hardware (Letzter Zugriff am 16. 12. 2017).

ROS 2018

ROS.org (2018). 2018. URL: <https://www.ros.org> (Letzter Zugriff am 28. 01. 2018).

ROTHHÖFT 2014

Rothhöft, M.: Marktstudie SPS-Systeme. Ergebnisse einer bundesweiten Befragung von Unternehmen aus den Bereichen Maschinenbau und Ingenieurbüros im Januar/Februar 2014. Hrsg. von Michaela Rothhöft. Hemer. 2014. URL: http://www.marktstudien.org/-marktstudien/marktstudie_sps.html (Letzter Zugriff am 17. 10. 2014).

SACHSE 2017

Sachse, S.: OPC UA TSN solutions: from the field to the cloud. industrial ethernet book (2017) (102), S. 8–10.

SAYLER & DILLMANN 2011

Sayler, S.; Dillmann, R.: Experience-based optimization of universal manipulation strategies for industrial assembly tasks. Robotics and Autonomous Systems (2011). DOI: 10.1016/j.robot.2011.06.017.

SCHENK ET AL. 2014

Schenk, M.; Wirth, S.; Müller, E.: Fabrikplanung und Fabrikbetrieb. Berlin, Heidelberg: Springer, 2014. ISBN: 978-3-642-05458-7. DOI: 10.1007/978-3-642-05459-4.

SCHMIDT 1992

Schmidt, M.: Konzeption und Einsatzplanung flexibel automatisierter Montagesysteme. Bd. 41. iwb Forschungsberichte. Berlin, Heidelberg: Springer, 1992. ISBN: 978-3540550259.

SCHMIDTKE & JASTRZEBSKA-FRACZEK 1993

Schmidtke, H.; Jastrzebska-Fraczek, I.: Ergonomie. Daten zur Systemgestaltung und Begriffsbestimmungen. München: Hanser, 1993. ISBN: 978-3-446-43480-6.

SCHNELL & WIEDEMANN 2006

Schnell, G.; Wiedemann, B.: Bussysteme in der Automatisierungs- und Prozesstechnik. Grundlagen, Systeme und Trends der industriellen Kommunikation. Bd. 6. Vieweg Praxiswissen. Schnell, Gerhard, (Hrsg.) Wiesbaden: Vieweg. 2006. ISBN: 3-8348-0045-7.

SCHOU ET AL. 2013

Schou, C.; Damgaard, J. S.; Bogh, S.; Madsen, O.: Human-robot interface for instructing industrial tasks using kinesthetic teaching. In: IEEE ISR 2013. 2013 44th International Symposium on Robotics (ISR) Seoul, Korea. IEEE. 2013, S. 1–6. ISBN: 978-1-4799-1173-8. DOI: 10.1109/ISR.2013.6695599.

SECKNER 2008

Seckner, M.: Unterstützung automatisierter Mikroproduktion durch wandlungsfähige Montageanlagen. Konzeption und Realisierung einer flexiblen und reaktiven Mikromontage. Bd. 19: Mikromontage. Fortschritt-Berichte pak. Kaiserslautern: Techn. Univ., 2008. ISBN: 978-3-939432-91-3.

SELIG 2010

Selig, A.: Informationsmodell zur funktionalen Typisierung von Automatisierungsgeräten. Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik. Dissertation. Universität Stuttgart, 2010. Stuttgart.

SENDLER ET AL. 2013

Sendler, U.; Baum, G.; Borchering, H.; Broy, M.; Eigner, M.; Huber, A. S.; Kohler, H.; Russwurm, S.; Stümpfle, M.: Industrie 4.0. Beherrschung der industriellen Komplexität mit SysLM. Xpert.press. Berlin: Springer-Vieweg, 2013. ISBN: 978-3-642-36916-2.

SICILIANO & KHATIB 2008

Siciliano, B.; Khatib, O.: Springer Handbook of Robotics. Berlin, Heidelberg: Springer, 2008. DOI: 10.1007/978-3-540-30301-5.

SIEGERT & BOCIONEK 1996

Siegert, H.-J.; Bocionek, S.: Robotik: Programmierung intelligenter Roboter. Springer-Lehrbuch. Berlin, Heidelberg: Springer, 1996. ISBN: 978-3-540-60665-9. DOI: 10.1007/978-3-642-80067-2.

SMALE & RATCHEV 2009

Smale, D.; Ratchev, S.: A Capability Model and Taxonomy for Multiple Assembly System Reconfigurations. IFAC Proceedings Volumes 42 (2009) (4), S. 1923–1928. DOI: 10.3182/20090603-3-RU-2001.0556.

SPINNARKE 2016

Spinnarke, S.: Etabliert sich IO-Link Technologie auf dem Markt? Hrsg. von Produktion.de. 2016. URL: <https://www.produktion.de/trends-innovationen/etabliert-sich-io-link-technologie-auf-dem-markt-118.html> (Letzter Zugriff am 15. 09. 2016).

STATISTA 2015

Statista: Prognose zur Preisentwicklung eines Industrieroboters in den USA nach Posten bis zum Jahr 2025. 2015. URL: <https://de.statista.com/statistik/daten/studie/416525/umfrage/preisentwicklung-eines-industrieroboters> (Letzter Zugriff am 10. 03. 2018).

STENMARK & MALEC 2015

Stenmark, M.; Malec, J.: Knowledge-based instruction of manipulation tasks for industrial robotics. *Robotics and Computer-Integrated Manufacturing* 33 (2015), S. 56–67. DOI: 10.1016/j.rcim.2014.07.004.

STENMARK ET AL. 2015

Stenmark, M.; Malec, J.; Stolt, A.: From High-Level Task Descriptions to Executable Robot Code. In: Filev, D.; Jablkowski, J.; Kacprzyk, J.; Krawczak, M.; Popchev, I.; Rutkowski, L.; Sgurev, V.; Sotirova, E.; Szynekarczyk, P.; Zadrozny, S. (Hrsg.): *Intelligent Systems 2014*. Bd. 323. *Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, 2015, S. 189–202, ISBN: 978-3-319-11309-8. DOI: 10.1007/978-3-319-11310-4_17.

STÖRIG 1997

Störig, H. J.: Kleine Weltgeschichte der Philosophie. Bd. 11142. Frankfurt am Main: Fischer-Taschenbuch-Verl., 1997. ISBN: 3596111420.

TAKAI 2017

Takai, D.: Architektur für Websysteme. München: Carl Hanser, 2017. ISBN: 978-3-446-45056-1.

TAVARES & PAULA CAURIN 2010

Tavares, D. M.; Paula Caurin, G. A. de: Proposal for a device proxy using XIRP. In: 2010 9th IEEE/IAS International Conference on Industry Applications - INDUSCON 2010 Sao Paulo, Brazil. IEEE. 2010, S. 1–6. ISBN: 978-1-4244-8008-1. DOI: 10.1109/INDUSCON.2010.5740073.

THOMAS & WAHL 2001

Thomas, U.; Wahl, F. M.: A system for automatic planning, evaluation and execution of assembly sequences for industrial robots. In: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems Maui, HI, USA. IEEE. 2001, S. 1458–1464. ISBN: 0-7803-6612-3. DOI: 10.1109/IROS.2001.977186.

THOMAS 2008

Thomas, U.: Automatisierte Programmierung von Robotern für Montageaufgaben. In: Ausgezeichnete Informatikdissertationen 2008. Bonn: Köllen Druck+Verlag, 2008, S. 291–300, ISBN: 978-3-88579-413-4.

THOMAS ET AL. 2015

Thomas, U.; Stouraitis, T.; Roa, M. A.: Flexible assembly through integrated assembly sequence planning and grasp planning. In: 2015 IEEE International Conference on Automation Science and Engineering (CASE) Gothenburg, Sweden. IEEE. 2015, S. 586–592. ISBN: 978-1-4673-8183-3. DOI: 10.1109/CoASE.2015.7294142.

TOMPKINS ET AL. 2006

Tompkins, J. A.; White, J. A.; Bozer, Y. A.; Tanchoco, J. M. A.: Facilities Planning. John Wiley & Sons Inc., 2006.

TÖPFER 2012

Töpfer, A.: Erfolgreich Forschen: Ein Leitfaden für Bachelor-, Master-Studierende

und Doktoranden. Berlin, Heidelberg: Springer, 2012. ISBN: 978-3-642-34168-4. DOI: 10.1007/978-3-642-34169-4.

UFFELMANN & WIENZEK 2010

Uffelmann, J. R.; Wienzek, P.: IO-Link. München: Oldenbourg Industrieverlag, 2010. ISBN: 978-3-8356-3115-1.

ULRICH 2018

Ulrich, M.: 3D-Image-Stitching für roboterbasierte Messsysteme. *iwb*. Dissertation. Technische Universität, 2018. München.

ULRICH & HILL 1976

Ulrich, P.; Hill, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil I). *Wirtschaftswissenschaftliches Studium* 7 (1976), S. 304–309.

UNIVERSAL ROBOTS 2017

Universal Robots: User Manual. 2017. URL: <https://www.universal-robots.com/download> (Letzter Zugriff am 22. 01. 2018).

VAJNA ET AL. 2009

Vajna, S.; Hehenberger, P.; Zeman, K.; Bley, H.; Weber, C.: CAx für Ingenieure. Eine praxisbezogene Einführung. 2. Aufl. Berlin, Heidelberg: Springer, 2009. ISBN: 978-3-540-36039-1.

VDI 2016

VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik: Industrie 4.0 Service Architecture. 2016.

VDI 2860

Verein Deutscher Ingenieure: Montage- und Handhabungstechnik - Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole. VDI-Richtlinie. Düsseldorf: VDI. 1990.

VDMA 66430-1

VDMA: XML-basiertes Kommunikationsprotokoll für Industrieroboter und prozessorgestützte Peripheriegeräte (XIRP). VDMA. ICS 25.040.30; 35.240.50. Berlin: Beuth. 2006.

VDMA 2013

VDMA und IKB: Maschinenbau und Investitionsgüterindustrie – weiter auf Erfolgskurs. 2013.

VDI 2861

Verein Deutscher Ingenieure: Kenngrößen für Industrieroboter. VDI. Düsseldorf: VDI-Verl. 1988.

VOGEL-HEUSER ET AL. 2012

Vogel-Heuser, B.; Bayrak, G.; Frank, U.: Forschungsfragen in 'Produktionsautomatisierung der Zukunft'. Acatech-Materialien. München: Dt. Akad. der Technikwiss, 2012. ISBN: 3942044048.

VOGEL-HEUSER ET AL. 2017

Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M. ten: Handbuch Industrie 4.0 Bd.2. Berlin, Heidelberg: Springer, 2017. ISBN: 978-3-662-53247-8. DOI: 10.1007/978-3-662-53248-5.

VOIGT 2015

Voigt, K.-I.: Automatisierung. Hrsg. von Springer Gabler Verlag. Gabler Wirtschaftslexikon. 2015. URL: <http://wirtschaftslexikon.gabler.de/Archiv/72569/automatisierung-v7.html> (Letzter Zugriff am 05. 10. 2017).

VOLZ 2016

Volz, M.: Trends in der industriellen Kommunikation. Hrsg. von Polyscope. Blinkert Medien AG. 2016. URL: http://www.polyscope.ch/archiv/2016/8/trends-in-der-industriellen-kommunikation_56277 (Letzter Zugriff am 01. 11. 2017).

WEBER 2009

Weber, W.: Industrieroboter. Methoden der Steuerung und Regelung. 2., neu bearb. Aufl. München: Hanser, 2009. ISBN: 978-3-446-41031-2.

WEBER 2017

Weber, M.: Architektur zur Eignungsüberprüfung bei der dienstbasierten Programmierung von Automatisierungsanlagen. Nicht veröffentlichte Studienarbeit. Technische Universität München, 2017. München.

WECK 2006

Weck, M.: Werkzeugmaschinen 4. Berlin, Heidelberg: Springer, 2006. ISBN: 978-3-642-38747-0. DOI: 10.1007/978-3-540-45366-6.

WEINLÄNDER 2016

Weinländer, M.: OPC UA industrial networks for digital factory infrastructure. industrial ethernet book 97 (2016) (10), S. 19–21.

WELLENREUTHER & ZASTROW 2008

Wellenreuther, G.; Zastrow, D.: Automatisieren mit SPS – Theorie und Praxis. 4. Aufl. Wiesbaden: GWV Fachverlage GmbH, 2008. ISBN: 978-3-8348-0231-6.

WGP 2016

Bauernhansl, T.; Krüger, J.; Reinhart, G.; Schuh, G.: WGP-Standpunkt Industrie 4.0. Hrsg. von Wissenschaftliche Gesellschaft für Produktionstechnik WGP e. V. 2016.

WIENDAHL 2002

Wiendahl, H.-P.: Wandlungsfähigkeit. wt Werkstattstechnik online 92 (2002).

WIENDAHL 2005

Wiendahl, H.-P.: Planung modularer Fabriken. Vorgehen und Beispiele aus der Praxis. München & Wien: Hanser, 2005. ISBN: 9783446400450.

WIENDAHL 2014

Wiendahl, H.-P.: Betriebsorganisation für Ingenieure. 8., überarb. Aufl. München: Hanser, 2014. ISBN: 9783446440531.

WIENDAHL ET AL. 2002

Wiendahl, H.-P.; Hegenscheidt, M.; Winkler, H.: Anlaufrobuste Produktionssysteme. wt Werkstattstechnik online (2002).

WIENDAHL ET AL. 2004

Wiendahl, H.-P.; Gerst, D.; Keunecke, L.: Variantenbeherrschung in der Montage. Berlin, Heidelberg: Springer, 2004. ISBN: 978-3-642-62372-1. DOI: 10.1007/978-3-642-18947-0.

WIENDAHL ET AL. 2007

Wiendahl, H.-P.; ElMaraghy, H. A.; Nyhuis, P.; Zäh, M. F.; Wiendahl, H.-H.; Duf-

fie, N.; Brieke, M.: Changeable Manufacturing - Classification, Design and Operation. CIRP Annals - Manufacturing Technology 56 (2007) (2), S. 783–809. DOI: 10.1016/j.cirp.2007.10.003.

DIN EN 8373

Wörterbuch Industrieroboter (1996). DIN EN ISO. ICS 01.040.25; 25.040.30. Berlin: Beuth. 1996.

XU ET AL. 2012

Xu, L. D.; Wang, C.; Bi, Z.; Yu, J.: AutoAssem: An Automated Assembly Planning System for Complex Products. IEEE Transactions on Industrial Informatics 8 (2012) (3), S. 669–678. DOI: 10.1109/TII.2012.2188901.

ZACHARIAS 2012

Zacharias, F.: Knowledge Representations for Planning Manipulation Tasks. Bd. 16. Berlin, Heidelberg: Springer, 2012. ISBN: 978-3-642-25181-8. DOI: 10.1007/978-3-642-25182-5.

ZHANG ET AL. 2017

Zhang, N.; Zhang, Y.; Ma, C.; Wang, B.: Path planning of six-DOF serial robots based on improved artificial potential field method. In: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO) Macau. IEEE. 2017, S. 617–621. ISBN: 978-1-5386-3742-5. DOI: 10.1109/ROBIO.2017.8324485.

Anhang

A.1 Übersicht aller gestellten Konzeptanforderungen

	ID	Anforderung
Vernetzung	AV1	Echtzeitfähigkeit der Komponentensteuerung
	AV2	Automatisierung der Konfigurationsschritte zur Feldgeräteeinbindung
	AV3	Informationstechnische Schnittstelle für den Zugriff auf den Kommunikationsmaster
	AV4	Unterstützung der Heterogenität von Zugriffstechnologien
	AV5	Unterstützung von verschiedenen, hierarchischen Netzwerken
	AV6	Unterstützung primitiver Geräte- und Kommunikationstechnologien
	AV7	Nutzung bestehender Beschreibungsstandards zur Konfiguration
	AV8	Stellvertretende Publikation des Digital Twins
	AV9	Synchronisation des Digital Twins mit physischen Geräten
Fähigkeiten	AF1	Erstellung eines Fähigkeitenkatalogs
	AF2	Trennung zwischen einfachen und kombinierten Fähigkeiten
	AF3	Definition der Struktur von Fähigkeitenparametern
	AF4	Methode zur Definition neuer Parameterklassen
	AF5	Identifikation von Standardparameterklassen
	AF6	Stärkere Fokussierung auf Kontrollaufgaben und Transportprozesse
	AF7	Definition indirekter Fähigkeiten (MMI, Datenaustausch)

Ressourcenauswahl	AR1	Framework zum fähigkeitsbasierten Matchmaking
	AR2	Unterstützung des Matchmakings auf Parameterenebene
	AR3	Unterstützung der Lösung von Teilproblemen
	AR4	Klassifizierung der Teilprobleme in der Montage
	AR5	Unterstützung der verknüpften Lösung von Teilproblemen
	AR6	Erweiterbarkeit um neue Problemlösungsalgorithmen
	AR7	Einbeziehung des Menschen als Problemlösungsinstanz
	AR8	Unterstützung des Matchmakings virtueller Ressourcen
	AR9	Unterstützung des Matchmakings kombinierbarer Ressourcen
Programmierung	AP1	Effiziente Programmerstellung aus vorhandenen Planungsdaten
	AP2	Automatisierte Generierung und Integration des Materialflusses
	AP3	Unterstützung des Matchmakings von Materialflussressourcen
	AP4	Präsentation von Alternativen in der Materialflussplanung
	AP5	Möglichkeit zur getrennten Modellierung von Prozess- und Materialfluss
	AP6	Unterstützung der Verknüpfung von Prozesswerten (Datenfluss)
	AP7	Codegenerator zur Überführung des Prozessmodells in ausführbare Steuerungsprogramme
	AP8	Konzept zur geräteseitigen Akquise fehlender Ausführungsparameter

A.2 Ablauf der Vernetzung (Host-unabhängige KKTs)

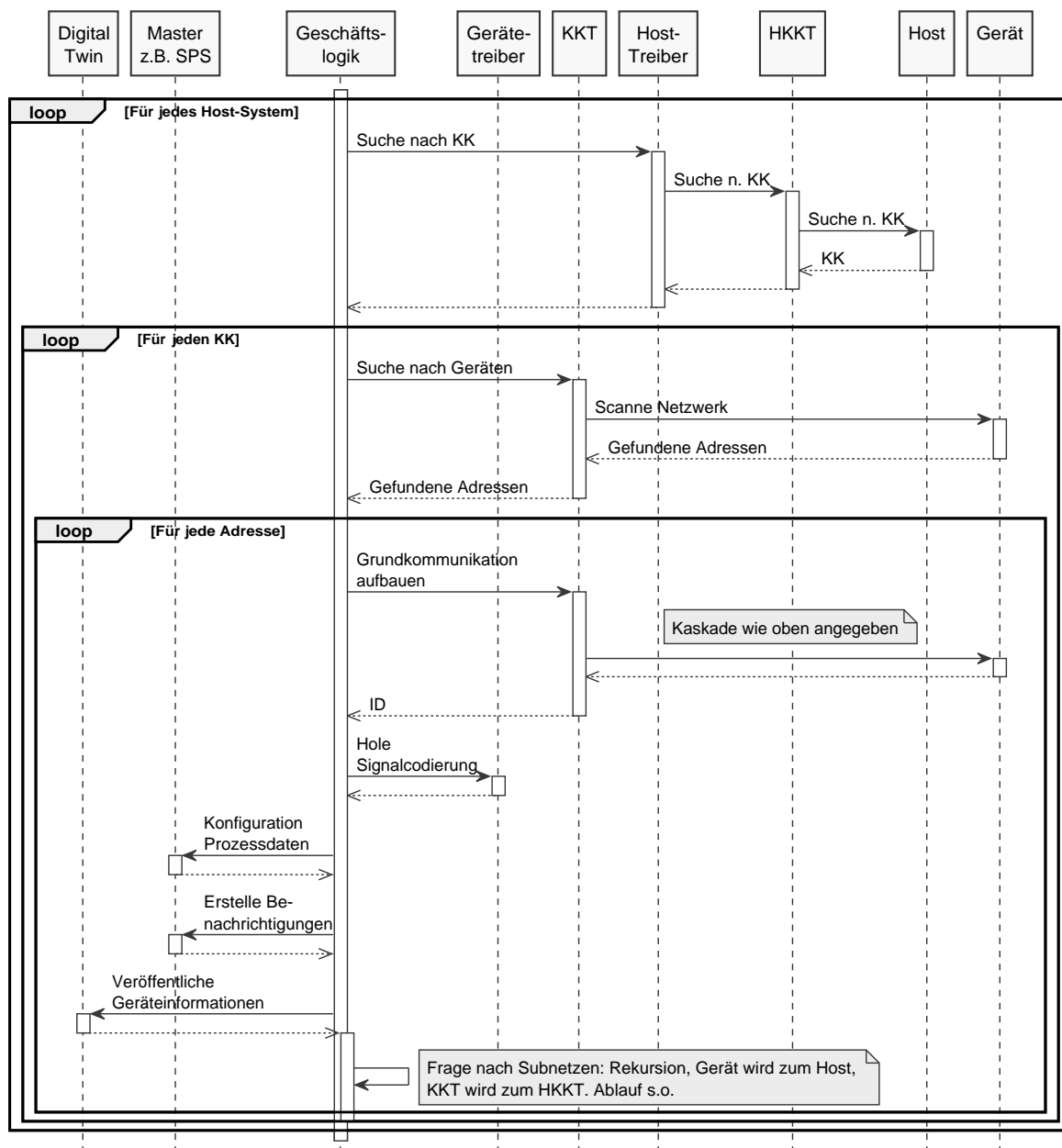


Abbildung A.1: Ablauf der Vernetzung (Host-unabhängige KKTs)

A.3 Elementare Fähigkeitenparameter

Bewegen

Bewegen		<i>Notwendigkeit</i>	<i>Datentyp</i>	<i>Einheit</i>
	Bezugssystem	v	frame	[Koordinaten]
	Endpose	v	frame	[Koordinaten]
	Bewegungsart	o	enum	[lin], [circ]
	Posegenauigkeit	o	vector(6)	mm bzw. rad
<i>Input</i>	Geschwindigkeit	o	real oder function	m/s
	Beschleunigung	o	real oder function	m/s ²
	Trajektorie	o	frame array	[Koordinaten]
	Trajektoriengeauigkeit	o	vector(6) array	[Koordinaten]
	Max. Ausführzeit	v	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

Alle numerischen Input-Parameter werden als Wertebereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen:

Über die maximale Ausführzeit kann das System die Geschwindigkeit und Beschleunigung selbst bestimmen. Werden diese Werte angegeben, können sie als Maximalwert definiert sein oder eine Funktion über die Zeit darstellen. Über die optionale Trajektorie lässt sich eine Bahn zur Zielpose spezifizieren – ansonsten wird diese durch das Gerät berechnet.

Wird die Fähigkeit überbestimmt definiert, etwa durch die Angaben von Geschwindigkeit, Beschleunigung, Genauigkeit und Ausführzeit, obliegt es dem Softwaresystem eine Optimierung innerhalb der gegebenen Wertebereiche durchzuführen, um so die ressourcenseitigen Werte zu erlangen.

Sichern

Sichern		Notwendigkeit	Datentyp	Einheit
<i>Input</i>	Sicherungsart	v	enum	[Norm]
	Sicherungskraft	o	real	N
	Posegenauigkeit	o	vector(6)	mm bzw. rad
	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	o	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

Alle numerischen Input-Parameter werden als Wertbereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen: Über den Aktivparameter kann ein Sicherungsvorgang gestartet oder beendet werden. Über die Sicherungsart wird ein kraft- oder formschlüssiges Wirkprinzip ausgewählt. Wird keine Kraft angegeben, dann sichert das System mit Maximalkraft.

Messen

Kraft messen		Notwendigkeit	Datentyp	Einheit
<i>Input</i>	Einheit	v	string	N
	Messunsicherheit	v	real	[SI-Einheit] oder Prozent
	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	v	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Ergebnis	v	real	[SI-Einheit]

Alle numerischen Input-Parameter werden als Wertbereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen: Weitere Messskills werden ähnlich modelliert.

Prüfen

Anwesenheit prüfen		<i>Notwendigkeit</i>	<i>Datentyp</i>	<i>Einheit</i>
<i>Input</i>	Prüfwert	v	real	[Einheitenlos]
	Prüfbedingung	v	string	[Operator]
	Prüftoleranz	v	string/real	[Einheitenlos]
	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	v	real	ms
	...	o	...	
<i>Output</i>	Status	v	enum	[Statemachine]
	Ergebnis	v	bool	true/false

Alle numerischen Input-Parameter werden als Wertbereiche angegeben. v: verpflichtend, o: optional

Anmerkungen: Weitere Prüfskills werden ähnlich modelliert.

Datenaustausch

Daten speichern		<i>Notwendigkeit</i>	<i>Datentyp</i>	<i>Einheit</i>
<i>Input</i>	Tabelle	o	string	[...]
	Identifizierung	v	string	[...]
	Wertname (Key)	v	string	[...]
	Wert	v	string	[...]
	Max. Ausführzeit	v	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

Alle numerischen Input-Parameter werden als Wertbereiche angegeben. v: verpflichtend, o: optional

Daten lesen		<i>Notwendigkeit</i>	<i>Datentyp</i>	<i>Einheit</i>
<i>Input</i>	Tabelle	o	string	-
	Identifizierung	v	string	-
	Wertname (Key)	v	string	-
	Max. Ausführzeit	v	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%
	Wert	v	string	-

Alle numerischen Input-Parameter werden als Wertbereiche angegeben. v: verpflichtend, o: optional

Daten löschen und *Daten aktualisieren* werden analog zu *Daten lesen* modelliert.

Anmerkungen:

Für das Verarbeiten von Informationen ist die Semantik der Information stark von der Aufgabe bzw. Hardware abhängig. Eine generische Fähigkeitenschnittstelle lässt sich demnach nur aufbauen, wenn die Informationsverarbeitung allgemeingültig gehalten wird. Hiermit wird die semantische Standardisierung eine Ebene „tiefer“ innerhalb der abzuspeichernden Parameterwerte verlegt.

Die *Identifizierung* kennzeichnet eine Entität eindeutig (z. B. ein spezifisches Produkt), an dieses können über die Wertname–Wert–Beziehung Informationen angehängt werden. Die Identifizierung kann als Index gesehen werden, welche mit Key-Value-Paaren befüllt werden. Informationsfähigkeiten besitzen keine ROI, da sie Outputparameter von Gerätefähigkeiten abspeichern und damit nicht direkt produktbezogen sind.

Mensch-Maschine-Interaktion

	Informationen visuell anzeigen	<i>Notwendigkeit</i>	<i>Datentyp</i>	<i>Einheit</i>
<i>Input</i>	Anzeigewert	v	[...]	[...]
	Betrachtungszeit	v	enum	[1]
	Datenumfang	v	enum	[2]
	Extremwerte	o	[...]	[3]
	Max. Ausführzeit	v	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

Alle numerischen Input-Parameter werden als Wertbereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen:

- Bei Fähigkeiten zur Anzeige wird davon ausgegangen, dass der Anwender nur die Rahmenparameter der Anzeige festlegt und das Gerät die notwendige Anzeigeform und -layout selbst generiert.
- Fähigkeiten der Mensch-Maschine-Interaktion besitzen keine ROI, da sie Outputparameter von Gerätefähigkeiten verarbeiten und damit nicht direkt produktbezogen sind.
- Über den Datentyp des *Anzeigewerts* (z. B. enumeration oder numerisch) sowie der weiteren Eingabeparameter kann das Anzeigegerät eine geeignete Darstellungsform selbst bestimmen. Im Falle der Fähigkeit *visuelle Anzeige* sind das bspw. Analoganzeigen, Digitalanzeigen oder bildhafte Anzeigen mit ihren jeweiligen Ausprägungen als

Istwertanzeige, Differenzanzeige, Voranzeige, Imperative Anzeige etc.¹. Der Aufbau der Darstellungsform benötigt mitunter Rückfragen beim Anwender.

- [1]: Zeitpunktbetrachtung oder historischer Datenverlauf.
- [2]: Anzahl der gleichzeitig anzuzeigenden Informationen (1 bis ∞). Beispielsweise ergibt die Anzeige von zwei Achsgeschwindigkeiten in einem Diagramm den Datenumfang von *Zwei*.
- [3]: Min.-Max.-Werte zur Festlegung der Darstellungsgrenzen.

Sonderoperationen

Erwärmen		Notwendigkeit	Datentyp	Einheit
	Wärmestrom	v	real	W
<i>Input</i>	ROI	v	region	[Koordinaten]
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

Alle numerischen Input-Parameter werden als Wertebereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen:

Das Erreichen einer geforderten Zieltemperatur kann über die *Temperatur messen-*Fähigkeit ermittelt und in Kombination als zusammengesetzte Fähigkeit modelliert werden.

¹vgl. SCHMIDTKE & JASTRZEBSKA-FRACZEK (1993)

A.4 Zusammengesetzte Fähigkeitenparameter

Verbinden

Lichtbogenschweißen (Gasentladung)		Notwendigkeit	Datentyp	Einheit
<i>Input</i>	Aktiv	v	bool	true/false
	Schweißspannung	v	real	V
	Schweißstrom	v	real	A
	Gastyp	v	enum	[Norm]
	Gasvolumenstrom	v	real	m ³ /s
	Elektrodenform	v	enum	[Norm]
	Elektrodenwerkstoff	v	enum	[Norm]
	Drahtvorschubgeschwindigkeit	v	real	m/s
	Drahtwerkstoff	v	enum	[Norm]
	ROI	v	region	[Koordinaten]
	Max. Ausführzeit	o	real	ms
...	o	
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%

Alle numerischen Input-Parameter werden als Wertbereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen: Der Grundwerkstoff ergibt sich aus den Produktinformationen.

Speichern

Speichern		Notwendigkeit	Datentyp	Einheit
<i>Input</i>	Ordnungszahl (OZ)	v	enum	[Norm]
	Lagerungspose	o	frame	[Koordinaten]
	Einlagerung	v	bool	true/false
	Speichergröße	o	int	[Koordinaten]
	Max. Durchlaufzeit	o	real	ms
	Max. Ausführzeit	o	real	ms
	...	o
<i>Output</i>	Status	v	enum	[Statemachine]
	Fortschritt	v	real	%
	Füllstand des Speichers	v	real	%

Alle numerischen Input-Parameter werden als Wertbereiche angegeben.

v: verpflichtend, o: optional

Anmerkungen:

- Speichern besteht immer aus Halte- sowie Bewegungsoperationen und wird deshalb als Task gesehen.
- Die Fähigkeit besitzt zwei abhängige Zustände, das *Einlagern* (Speichern aktiv) und das *Auslagern* (Speichern nicht mehr aktiv). Beide Zustände beziehen sich auf eine gemeinsame Menge. Für eine Modellierung erscheint es deshalb sinnvoll, dass ein Gerät (z. B. ein Lager oder Puffer) die Fähigkeit *Speichern* anbietet und über dessen Parameter die Ein- oder Auslagerung definiert wird.

A.5 Architektur des Prozessmodells

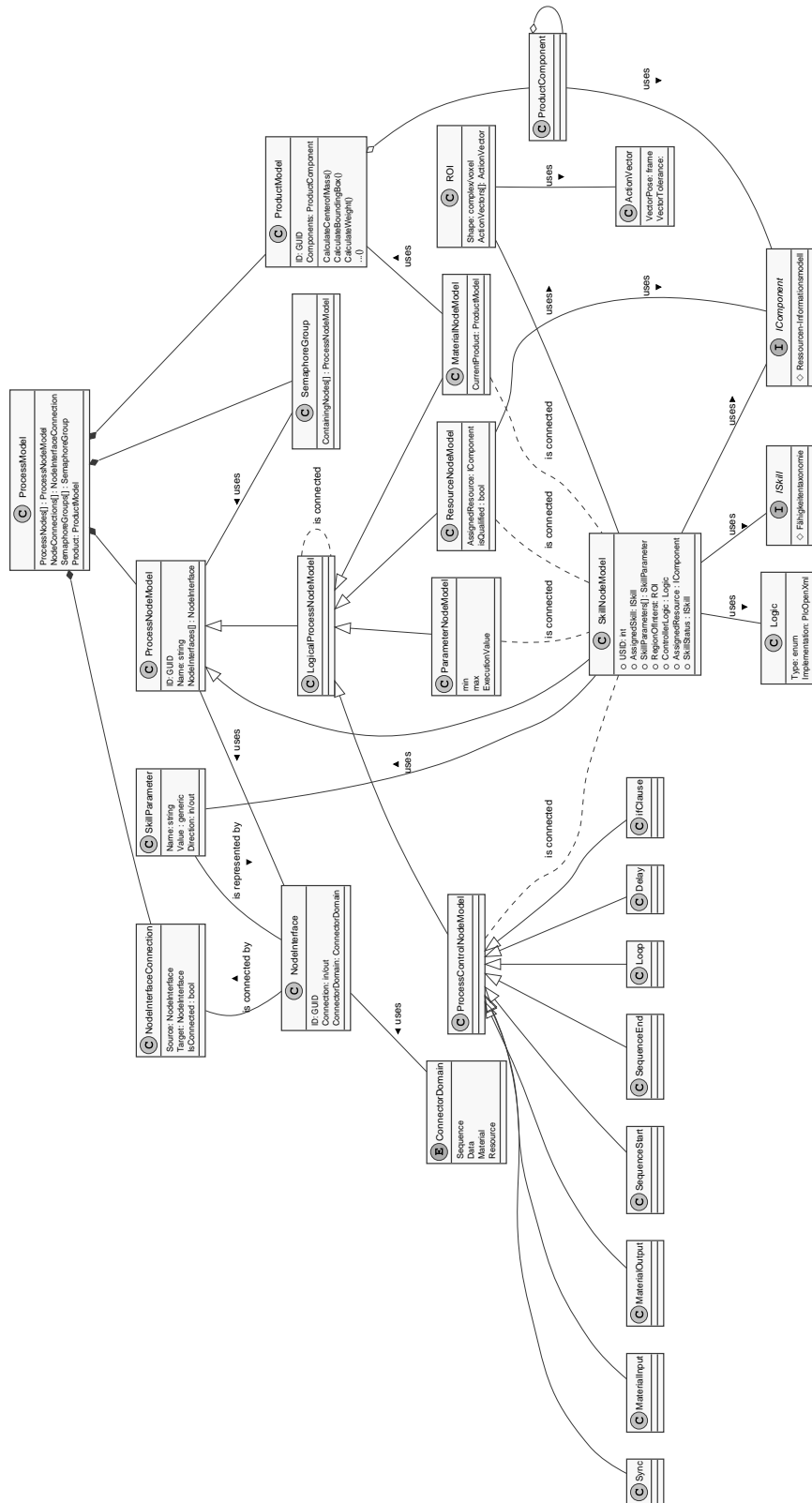


Abbildung A.2: Architektur des Prozessmodells, dargestellt als Klassendiagramm. Attribute zur graphischen Darstellung der Konten (z. B. Layout-Koordinaten) sind nicht aufgeführt.

A.6 Weitere Demonstratorabbildungen

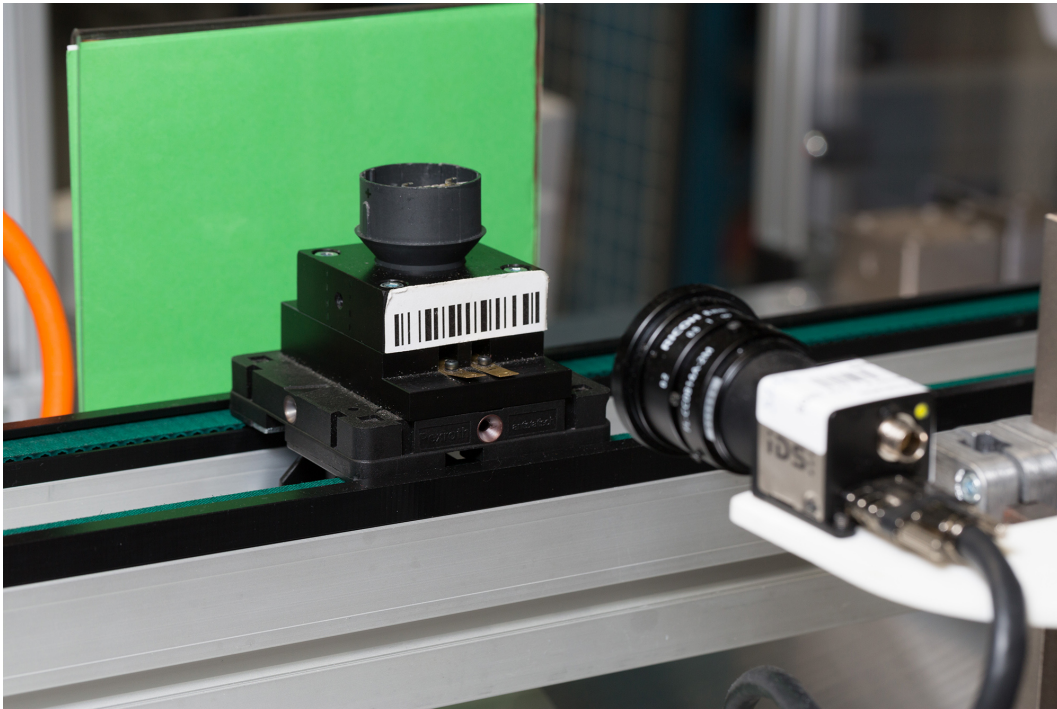


Abbildung A.3: Bestückungsmodul 1 – Barcodescan des Werkstückträgers und optische Identifikation der Lampenunterteilvariante

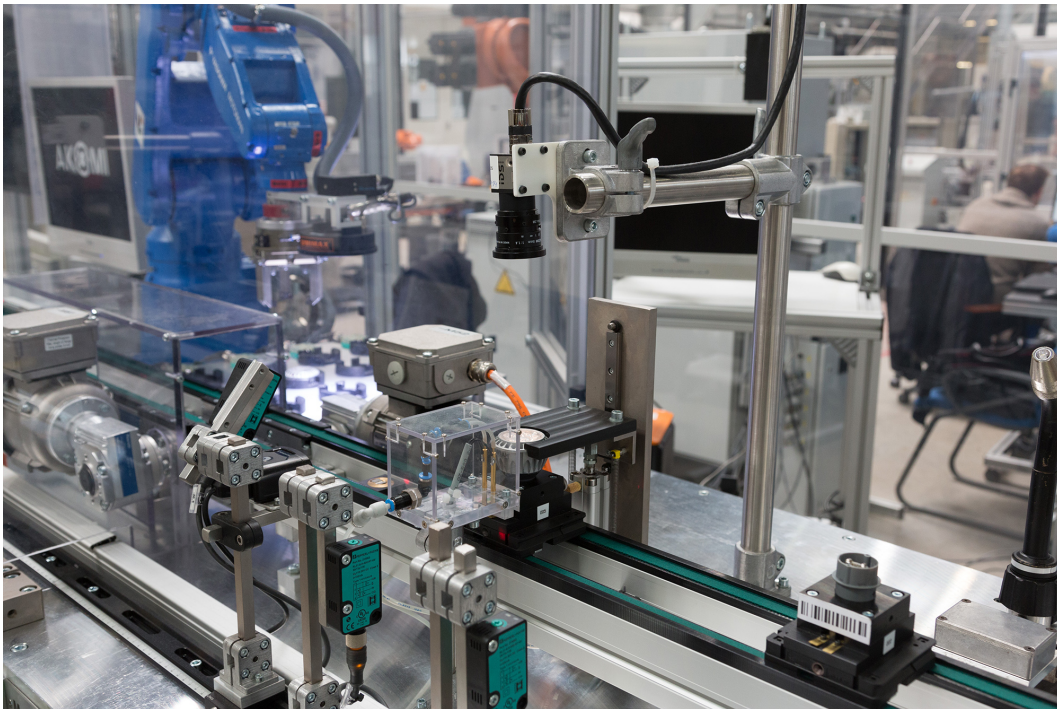


Abbildung A.4: Qualitätssicherungsstation Modul 3

A.7 Ergänzende Informationen zur Umfrage

Tabelle A.3: Benötigter Kenntnisstand zur Programmierung des Demonstrators. Linke Spalte: Herkömmlich. Rechte Spalte: Framework zur Prozessmodellierung TOMAP.

Proband	Welche fachlichen Kenntnisse (Fachdisziplinen, Methoden, Wissen) brauchen Sie zur Programmierung des gezeigten Demonstrators?	Welche fachlichen Kenntnisse (Fachdisziplinen, Methoden, Wissen) brauchen Sie bei der Verwendung des Frameworks zur Prozessspezifikation des gezeigten Demonstrators?
1	Grundlagen Mathematik, Robotik, Aktorik. Kenntnisse im Bereich Pneumatik, Systemtheorie, Programmierung, Sicherheitsnormen, Bildverarbeitung	Grundlagen wie vorher. AT-Kenntnisse entfallen (Systemtheorie, Aktorik, BV, Programmierung)
2	Grundlagen: Kinematik, Programmierung. Kenntnisse: BV-Programierung, SPS, Mechanik und Elektrotechnik, Produkt- und Prozessverständnis	Keine Programmierkenntnisse in den Einzelbereichen mehr. Nach wie vor: Produkt- und Prozessverständnis
3	Programmierung SPS, BV, Robotik. Grundlagen: Elektronik	Automatisierungstechnisches Verständnis
4	Grundlagen Programmierung, SPS, BV, Robotik, Kommunikation und Anlagentechnik	Vorherig genannte Kenntnisse sind hilfreich, werden aber weniger detailliert benötigt
5	Programmierung SPS, BV, Robotik. Grundlagen: AT, Mathematik, Datenbanken, Elektrotechnik	Kenntnisse entfallen: Robotik, SPS, BV, Mathematik. Nach wie vor: Grundlagen AT
6	Programmierung SPS, BV, Robotik. Grundlagen: AT, Maschinenbau, allg. Programmierung	Kenntnisse entfallen: Programmierung SPS, BV, Robotik. Vorher gennante Grundlagen bleiben.
7	Programmierung SPS, BV. Hardwarekenntnisse, Produktionsprozess-Kenntnisse, Sicherheitstechnik	Produktionsprozess-Kenntnisse
8	Programmierung: SPS, Robotik (Hersteller-spezifisch!), BV-Grundlagen, Hardwarekenntnisse, Kommunikation, Sicherheitstechnik, Prozesswissen	Nach wie vor: Produktionsprozess / Aufgabe, Überblick über Fähigkeiten, kein Spezialwissen mehr, Grundlegende AT-Kenntnisse
9	Programmierung SPS, BV, Robotik. AT-Wissen, Kommunikation, Sicherheitstechnik	Nach wie vor: Montageprozess, Kenntnisse der Fähigkeiten, Grundlegende AT-Kenntnisse

AT: Automatisierungstechnik, BV: Bildverarbeitung

Tabella A.4: Einzelzeiten in Stunden für die Einführung einer neuen Produktvariante

	Die Produktanforderungen ändern sich (s. u.). Wie lange benötigen Sie für die Aussage, ob das neue Produkt auf der Anlage montierbar ist?	Die Produktanforderungen ändern sich (s. u.). Wie lange benötigen Sie für die Änderung der Programmierung?	Die Produktanforderungen ändern sich (s. u.). Wie lange benötigen Sie mit dem TOMAP-Framework für die Aussage, ob das neue Produkt auf der Anlage montierbar ist?	Die Produktanforderungen ändern sich (s. u.). Wie lange benötigen Sie mit dem TOMAP-Framework zur Änderung der Programmierung?
	Aktivitäten: Identifikation der Parameter anzupassender Prozesse, Suche korrespondierender Gerätemerkmale. Kein Vergleich zw. geeigneten Gerätealternativen	Identifikation der Anpassungsvariablen, Identifikation der Codebausteine, Durchführung der Anpassung; Test Anzunpassende Systeme: Aktorik; Sensorik; Materialfluss	Aktivitäten: Identifikation der Fähigkeitenparameter, Identifikation der Fähigkeiten, Durchführung der Anpassung. Kein Vergleich zw. geeigneten Gerätealternativen	Aktivitäten: Identifikation der Fähigkeitenparameter, Identifikation der Fähigkeiten, Durchführung der Anpassung, Anzunpassende Systeme: Aktorik; Sensorik; Materialfluss
Pro- bant	Ände- rung Pro- dukt- geome- trie	Ände- rung Werk- stoff	Ände- rung Pro- dukt- geome- trie	Ände- rung Pro- dukt- geome- trie
2. Pro- duktva- riante mit versch. LED- Aufbau	Ände- rung Taktzeit	Ände- rung Pro- dukt- geome- trie	Ände- rung Pro- dukt- geome- trie	2. Pro- duktva- riante mit versch. LED- Aufbau
1	2	5	8	2
2	2	4	2	2
3	2	4	1	1
4	5	5	10	10
5	2	5	5	5
6	4	8	11	8
7	0,5	0,5	2	4
8	1	1	2	8
9	0,5	0,25	1	20

431 936 Zeichen, geschrieben mit ♡ in Lua \TeX .