

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Betriebswissenschaften und Montagetechnik

am Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb)

## **System zur aufgabenorientierten Programmierung für die Mensch-Roboter-Kooperation**

**Julia Katharina Berg**

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. rer. nat. Tim C. Lüth

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Gunther Reinhart
2. Prof. Dr.-Ing. Verena Nitsch

Die Dissertation wurde am 11.02.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 06.07.2020 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.



# Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Fraunhofer-Institut für Gießerei-, Composite- und Verarbeitungstechnik (IGCV) in Augsburg und am Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb) der Technischen Universität München.

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. Gunther Reinhart für die stets wohlwollende Förderung, die Möglichkeit zur Promotion und Weiterentwicklung am Institut. Bei Frau Professorin Dr.-Ing. Verena Nitsch bedanke ich mich recht herzlich für die Übernahme des Koreferats. Durch die Zusammenarbeit während des Projekts FORobotics konnte ich wertvolle Erkenntnisse zur Betrachtung der Robotik aus arbeitswissenschaftlicher Sicht erlangen.

Des Weiteren danke ich auch Herrn Prof. Dr.-Ing. Lüth für die Übernahme des Prüfungsvorsitzes.

Ich bedanke mich bei allen Kolleginnen und Kollegen sowie Projektpartnern für die wertvolle Zusammenarbeit, in der ich viel lernen und zahlreiche Erfahrungen sammeln konnte. Hervorheben möchte ich meinen Dank an Albrecht Lottermoser für seine unermüdliche Motivation, die Robotik am Institut zum Laufen zu bringen. Für die kritische Durchsicht der Arbeit bedanke ich mich bei Dr. Cedric Schultz und Dr. Marcel Wagner. Außerdem bedanke ich mich bei allen Studierenden, die mich mit ihren Arbeiten bei der Dissertation unterstützt haben. Hervorheben möchte ich die Unterstützung durch Danilo Hoss bei Implementierungsarbeiten während seiner Zeit als HiWi und Bachelorand.

Ganz besonders danke ich meinen Eltern für die Unterstützung und die Freiheiten, die sie mir bei der Wahl meines Bildungswegs gelassen haben. Darüber hinaus bedanke ich mich in besonderem Maße bei Christoph Berger, der mich bei der während meiner Dissertation stets unterstützt hat.

München, im Oktober 2020

Julia Berg



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>I</b>
<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>Formelzeichenverzeichnis</b>	<b>VI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Entwicklungen der Mensch-Roboter-Kooperation in der Montage	1
1.2 Zielsetzung der Arbeit und Betrachtungsbereich	5
1.3 Aufbau der Arbeit	7
<b>2 Begriffsbestimmung und Grundlagen</b>	<b>9</b>
2.1 Übersicht	9
2.2 Begriffsdefinitionen	9
2.3 Montage	12
2.3.1 Arten von Montagesystemen	13
2.3.2 Planung von Montagesystemen	14
2.3.3 Methods-Time-Measurement	15
2.4 Formen der Mensch-Roboter-Kooperation	16
2.4.1 Begrifflichkeiten und Definition im Zusammenhang mit der Mensch-Roboter-Kooperation	16
2.4.2 Gründe für die Mensch-Roboter-Kooperation	17
2.4.3 Leichtbauroboter	18
2.4.4 Sicherheit in der Mensch-Roboter-Kooperation	18
2.5 Grundlagen der Roboterprogrammierung	19
2.6 Fazit	21
<b>3 Stand der Wissenschaft und Technik</b>	<b>23</b>
3.1 Ansätze zur Programmierung von Industrierobotern	23
3.1.1 Ansätze zur vereinfachten Programmierung in der Forschung	23
3.1.2 Kommerzielle Programmiersysteme	27
3.2 Planung von Applikationen der Mensch-Roboter-Kooperation	29
3.2.1 Allgemeine Ansätze und Projekte zur Planung	29

3.2.2	Beschreibung von Aufgaben	30
3.2.3	Beschreibung der Fähigkeiten der Ressourcen	32
3.2.4	Abgleich der Anforderungen und Fähigkeiten zur Aufgabenzuteilung	34
3.2.5	Simulation für die Mensch-Roboter-Kooperation	36
3.3	Betrieb in der Mensch-Roboter-Kooperation	38
3.3.1	Sicherheitstechnologien zur Kollisionsvermeidung	39
3.3.2	Ansätze zur kollaborierenden Arbeit	40
3.3.3	Tätigkeitserkennung	42
3.3.4	Arbeitswissenschaftliche Studien zur gemeinsamen Arbeit	43
3.4	Zusammenfassung und Bewertung des Standes der Technik	45
<b>4</b>	<b>Anforderungsanalyse an ein Programmiersystem für die Mensch-Roboter-Kooperation</b>	<b>49</b>
4.1	Vorgehen zur Anforderungsanalyse	49
4.2	Planung	50
4.3	Programmierung	50
4.3.1	Programmgenerierung	50
4.3.2	Analyse der Nutzergruppen	51
4.3.3	Benutzerschnittstelle	52
4.4	Betrieb	53
4.5	Sicherheit	54
4.6	Zusammenfassung der Anforderungen an das aufgabenorientierte Programmiersystem	54
<b>5</b>	<b>Grundstruktur und Nebenmodule des Programmiersystems</b>	<b>57</b>
5.1	Ableitung der Struktur des Programmiersystems	57
5.2	Randbedingungen des Systems	60
5.3	Beschreibung der Aufgaben	61
5.4	Analyse von Montageaufgaben für die MRK	63
5.5	Informationsschicht	64



5.5.1	Aufgabenmodell	65
5.5.2	Umweltmodell	65
5.5.3	Wissensmodell	66
5.6	Nutzungsschicht	70
<b>6</b>	<b>Hauptmodule des Programmiersystems</b>	<b>73</b>
6.1	Planungsmodul	73
6.1.1	Ableitung des Montageplans und Beschreibung der Aufgaben	73
6.1.2	Abgleich der Fähigkeiten	76
6.1.3	Szenarioplanung	78
6.2	Programmiermodul	81
6.2.1	Programmplanung	83
6.2.2	Konfiguration des Tätigkeiterkennungsmodells	85
6.2.3	Integration der Tätigkeitserkennung durch Schnittstellendefinition	99
6.2.4	Simulation	100
6.3	Betriebsmodul	102
6.3.1	Tätigkeitserkennung	103
6.3.2	Kollisionsvermeidung	104
6.4	Sicherheitsbetrachtung	106
6.4.1	Integration in die Module	109
6.4.2	Unterstützte Risikoanalyse	111
<b>7</b>	<b>Vorgehen zur Integration des Programmiersystems</b>	<b>113</b>
7.1	Beschreibung des Vorgehensmodells	113
7.2	Anpassung des Vorgehensmodells	114
7.2.1	Orientierung	116
7.2.2	Analyse	117
7.2.3	Design	117
7.2.4	Realisierung	121
7.2.5	Betrieb	122

<b>8</b>	<b>Umsetzung und Erprobung</b>	<b>123</b>
8.1	Versuchsaufbau	123
8.2	Umsetzung des Systems	124
8.2.1	Planungsmodul	125
8.2.2	Benutzerschnittstelle	128
8.2.3	Programmiermodul	130
8.2.4	Betriebsmodul	133
8.3	Erprobung und Evaluation	137
8.3.1	Evaluation der Benutzerschnittstelle	138
8.3.2	Evaluation der Tätigkeitserkennung	139
<b>9</b>	<b>Wirtschaftliche und Technische Bewertung</b>	<b>141</b>
9.1	Technische Bewertung	141
9.2	Wirtschaftliche Bewertung	143
<b>10</b>	<b>Zusammenfassung und Ausblick</b>	<b>149</b>
10.1	Zusammenfassung	149
10.2	Ausblick	151
<b>11</b>	<b>Literaturverzeichnis</b>	<b>153</b>
<b>Anhang</b>		<b>175</b>
A1	Wissensmodell: Fügeoperationen	175
A2	Wissensmodell: Beschreibung des Roboters	177
A3	Eignungsgrade von Mensch und Roboter	178
A4	Berechnung der Matrizen des Hidden Markov Modells	179
A5	Unterstützte Risikoanalyse	181
A6	Ergebnisse des genetischen Algorithmus	183
A7	Vorgehensmodell	186
A8	Evaluationsbogen	187
<b>Verzeichnis betreuter Studienarbeiten</b>		<b>193</b>

## Abkürzungsverzeichnis

Abkürzung	Bedeutung
3D	3-dimensional
AAPP	Augmented Assembly Priority Plan
BBT	Basisbauteil
CAD	Computer Aided Design
DMM	Digitales Menschmodell
FBT	Fügebauteil
FE	Finite Elemente
HMM	Hidden Markov Modelle
IP	Internet Protocol
IR	Industrieroboter
Json	Java Script Object
KMU	Kleine und mittelständische Unternehmen
LBR	Leichtbauroboter
LHMM	Layered Hidden Markov Modell
MMS	Mensch-Maschine-Schnittstelle
MRK	Mensch-Roboter-Kooperation
MTM	Methods-Time-Measurement
PDDL	Planning Domain Definition Language
ROS	Robot Operating System
TCP	Transmission Control Protocol
VR	Virtual Reality
XML	Extensible Markup Language

## Formelzeichenverzeichnis

Variable	Einheit	Bedeutung
$P$		Wahrscheinlichkeit eines Übergangs
$t$		Zeit
$a$		Übergangswahrscheinlichkeit
$A$		Transitionsmatrix
$B$		Emissionsmatrix
$X$		Zustandsreihenfolge
$\mu$		Mittelwert
$\Sigma$		Kovarianzmatrix
$D$		Dimension von $\Sigma$
$s$		Anzahl der Unteraufgaben
$H$		Hilfsvektor
$m$		Mittelpunkt
$S$		Punkt
$\alpha$		Winkel
$y$		y-Koordinate
$x$		x-Koordinate
$r$		Distanzvektor
$N$		Dichtefunktion

# 1 Einleitung

## 1.1 Entwicklungen der Mensch-Roboter-Kooperation in der Montage

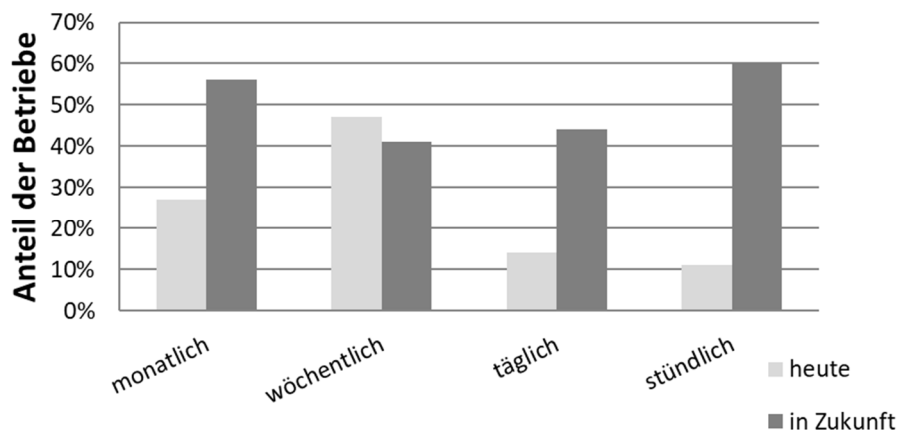
„Kollege Roboter“ – eine Schlagzeile, die seit einiger Zeit oft zu lesen ist (z. B. KREISZEITUNG.DE 2019; PLATTFORM INDUSTRIE 4.0 2019). Es bedeutet, dass Roboter nicht mehr hinter einem Schutzzaun arbeiten und dort täglich dasselbe Programm durchlaufen, sondern dass Mensch und Roboter in einem gemeinsamen Arbeitsraum eine Aufgabe, wie z. B. Montageaufgaben, erfüllen.

Die dafür konzipierten Roboter, die oft auch als Leichtbauroboter bezeichnet werden, ermöglichen diese Zusammenarbeit durch zusätzlich integrierte Sicherheitstechnik. Was bringt jedoch diese Zusammenarbeit zwischen Mensch und Roboter in der Produktion?

Die Montage unterliegt seit Jahren einem Wandel. Die Anzahl der Varianten von Produkten ist laut Studien gestiegen (KINKEL 2005). Dies liegt vor allem an dem Trend zu mehr kundenindividualisierten Produkten (REINHART 2017). Der Markt entwickelt sich zu einem Käufermarkt, bei dem der Kunde bestimmt, was angeboten wird. Dies zwingt die produzierenden Unternehmen dazu, auf die ständigen Veränderungen in der Nachfrage in kurzer Zeit zu reagieren. Gerade in der Montage führt dies zu einem Bedarf an Wandlungsfähigkeit und Flexibilität (ABELE & REINHART 2011). Der Mensch kann nach BEUMELBURG (2005) auf Änderungen, wie neue Produktvarianten oder Veränderungen im Produktionsablauf, reagieren und bietet daher nach der Definition von WIENDAHL ET AL. (2014) Flexibilität und Wandlungsfähigkeit. Der größte Anteil der Montageaufgaben wird auch derzeit noch manuell ausgeführt (FRAUNHOFER IAO 2016; LOTTER & WIENDAHL 2006). Dies führt zu einem hohen Kostendruck, vor allem an Hochlohnstandorten wie Deutschland (FRAUNHOFER IAO 2016). LOTTER & WIENDAHL (2006) beschreiben in ihrem Werk, dass die Montage 70 % der Kosten eines Produkts verursacht, was auf den hohen Anteil manueller Arbeit zurückzuführen ist. Dies führte in der Vergangenheit und führt heute noch zu Verlagerungen der Montage in Länder mit geringeren Lohnkosten (MATTHIAS ET AL. 2013).

Eine Studie zeigt, dass die Steigerung an Varianten und individualisierten Produkten zu einem schwankenden personalseitigen Kapazitätsbedarf führt, wie in Abbildung 1 dargestellt ist. So beschreiben Unternehmen, dass sie in Zukunft vermehrt einen

schwankenden Bedarf an Personalkapazitäten sogar innerhalb eines Tages bzw. stündlich zu verzeichnen haben werden (GANSCHAR & GERLACH 2013).



*Abbildung 1: Ergebnis der Studie zu personalseitigen Kapazitätsschwankungen nach GANSCHAR & GERLACH (2013)*

An vielen Stellen wird nach wie vor die Automatisierung erhöht, um Lohnkosten zu senken. Dabei wird vor allem auf Roboter zurückgegriffen, da diese als Automatisierungskomponenten Flexibilität aufweisen (HAAG 2015). Dies zeigt auch die Statistik der Roboterverkäufe der letzten Jahre, wobei eine Steigerung von 30 % in 2017 gegenüber dem Vorjahr zu verzeichnen ist und die Prognose für die nächsten Jahre eine weitere Steigerung zeigt (INTERNATIONAL FEDERATION OF ROBOTICS 2018).

Um Roboter in dieser volatilen Produktion einsetzen zu können, werden Ansätze verfolgt, um deren Wandlungsfähigkeit zu erhöhen. ROSSANO ET AL. (2013) zeigen in ihrer Arbeit auf, dass für einen flexiblen Einsatz rekonfigurierbare Roboterzellen gefordert sind, die auf Änderungen reagieren können. Durch diese Art der Roboterzellen soll der Aufwand des Umprogrammierens verringert werden, so dass auch bei schwankenden Stückzahlen und einer variantenreichen Montage der Einsatz von Robotern rentabel wird (ROSSANO ET AL. 2013; SCHRAFT & MEYER 2006). Der Aspekt einer einfacheren Programmierung und Konfiguration von Robotersystemen wird auch von den Roboterherstellern aufgegriffen. So entstehen im Rahmen der Entwicklung der Leichtbauroboter auch vereinfachte, meist grafische, Benutzerschnittstellen (FRANKA EMIKA GMBH 2018; RETHINK ROBOTICS 2018; UNIVERSAL ROBOTS A/S 2020), die weniger Expertenkenntnis voraussetzen. Dennoch ist die Benutzerschnittstelle nach wie vor für jeden Roboterhersteller

unterschiedlich, weshalb man sich für die Nutzung verschiedener Roboter in andere Programmierumgebungen einarbeiten muss.

Aber auch bei der Automatisierung mit rekonfigurierbaren und wandlungsfähigen Roboterzellen können weiterhin nicht alle Aufgaben von einem Roboter übernommen werden. Dies sind vor allem Aufgaben, bei denen die kognitiven und sensorischen Fähigkeiten, die Kreativität oder Adaptierbarkeit des Menschen gefordert sind (HAAG 2015; BEUMELBURG 2005). Die Montage weist daher weiterhin einen geringen Anteil an Robotern von 12,8 % auf (INTERNATIONAL FEDERATION OF ROBOTICS 2018). Dies liegt daran, dass Roboter für einige Montageoperationen zurzeit noch nicht geeignet sind. Die International Federation of Robotics sieht hier durch die Mensch-Roboter-Kooperation jedoch Potenzial für eine Steigerung des Einsatzes von Robotik in der Montage (INTERNATIONAL FEDERATION OF ROBOTICS 2018). Bei der Mensch-Roboter-Kooperation arbeiten Mensch und Roboter in einem gemeinsamen Arbeitsraum und der Roboter kann den Menschen z. B. bei nicht-ergonomischen Tätigkeiten unterstützen oder beide zusammen erfüllen jeweils Aufgaben, um durch die Stärken eines Partners die Schwächen des anderen Partners ausgleichen zu können (THIEMERMANN 2005; AKELLA ET AL. 1999). Ein weiterer Aspekt der Mensch-Roboter-Kooperation ist eine höhere Flexibilität bezüglich potenzieller Umbaumaßnahmen der Applikation und damit die Fertigung von Produktvarianten (THIEMERMANN 2005). Aufgrund dieser erhöhten Flexibilität bietet die Mensch-Roboter-Kooperation daher Potenzial, in kleinen und mittelständischen Unternehmen (KMU) eingesetzt zu werden, die den Herausforderungen kleiner Stückzahlen begegnen und für die vor allem der hohe Programmieraufwand durch Experten ein Hindernis für den Einsatz von Robotern ist (SCHRAFT & MEYER 2006).

Neben den steigenden Lohnkosten bewirkt auch der demografische Wandel Bedarf nach Automatisierung. Nach PÖTZSCH & RÖßGER (2015) werden im Jahr 2030 mehr als ca. 35 % der Bevölkerung über 65 Jahre alt sein. Diesen stehen weniger Erwerbstätige gegenüber (PÖTZSCH & RÖßGER 2015). Außerdem wird das Thema eines Fachkräftemangels thematisiert (INSTITUT DER DEUTSCHEN WIRTSCHAFT KÖLN 2017). Die Mensch-Roboter-Kooperation kann auch hier einen Beitrag leisten, diese fehlenden Kapazitäten durch die Übernahme von Aufgaben durch einen Roboter zu kompensieren (HANDLING 2018). Bei Betrachtung des demografischen Wandels, durch den mehr ältere Arbeitskräfte zu verzeichnen sind, ist es wichtig, deren Arbeitskraft bis in ein hohes Alter zu erhalten. Dabei kann der Einsatz eines Roboters unterstützen, die schweren, monotonen Aufgaben, wie das Handhaben von

Bauteilen, für die Arbeitskraft zu übernehmen, so dass dieser entlastet wird (OBERER-TREITZ ET AL. 2013).

Zur Umsetzung von Applikationen der Mensch-Roboter-Kooperation (MRK) wurden sogenannte Leichtbauroboter entwickelt. Diese sind mit integrierter Sensorik ausgestattet, so dass der Roboter bei Kollision mit dem Menschen anhält und damit das Verletzungsrisiko gemindert wird. Basierend auf diesem Konzept haben viele Hersteller das Potenzial erkannt und jeweils ihren eigenen Roboter auf den Markt gebracht. Dennoch gibt es bisher nur eine eingeschränkte Anzahl an Applikationen. Auch bei KMU, die den Bedarf nach flexiblen Automatisierungslösungen haben, werden noch wenige dieser Applikationen tatsächlich eingesetzt.

Denn die Programmierung für die Mensch-Roboter-Kooperation weist eine neue Komplexität auf, da nicht nur das Programm des Roboters betrachtet werden muss, sondern der Mensch und die Schnittstellen zu seinen Aufgaben integriert werden müssen. Des Weiteren sollten die Roboterbewegungen den Anforderungen an eine Zusammenarbeit mit dem Menschen entsprechen. Hier zeigen Untersuchungen, dass die Bewegungen des Roboters vorhersehbar sein sollen (KOPPENBORG ET AL. 2013; THIEMERMANN 2005), aber auch, dass der Roboter antizipatorische Fähigkeiten für eine flüssige Zusammenarbeit aufweisen soll (HOFFMAN & BREAZEAL 2007). Allein diese beiden Aspekte zeigen ein Spannungsfeld auf, da der Roboter nicht zu autonom agieren darf, aber gleichzeitig auf den Menschen in gewisser Weise reagieren können soll. Eine weitere Herausforderung im Rahmen der Mensch-Roboter-Kooperation stellt die Planung dar. Aufgrund der unterschiedlichen Zielgrößen Wirtschaftlichkeit, Zeit und Ergonomie wird dies als komplexe Aufgabe angesehen (LEMMERZ ET AL. 2018).

Des Weiteren haben SIHN ET AL. (2016) in einer Studie die Hindernisse bei der Einführung von Applikationen der Mensch-Roboter-Kooperation untersucht. Aus dieser Studie geht hervor, dass eines der größten Hindernisse der Umgang mit Normen und Richtlinien zur Gewährleistung der Sicherheit und die Rezertifizierung bei Änderung der Applikation sind (SIHN ET AL. 2016). Zur Zertifizierung einer solchen Mensch-Roboter-Kooperation wurden bereits einige Leitfäden zur Verfügung gestellt (VDMA ROBOTICS + AUTOMATION 2016; DGUV 2015). Allerdings zeigt die Studie, dass der Aufwand für diesen Vorgang, der die Risikobeurteilung beinhaltet, zu hoch für viele Unternehmen scheint. Der Aspekt, dass nach einer Änderung an der Applikation der Vorgang erneut durchlaufen werden muss, schränkt das Potenzial der Mensch-Roboter-Kooperation zur flexiblen Produktion ein (SCHIEMANN ET AL. 2018). Daher stellt sich die Frage, wie dieser



Aufwand verringert werden kann und wie Unternehmen mit wenig Expertise auf diesem Gebiet darin unterstützt werden können.

Zusammenfassend lässt sich feststellen, dass die Mensch-Roboter-Kooperation besonders in kleinen und mittelständischen Unternehmen bisher kaum präsent ist, da Herausforderungen, wie die richtige Zuteilung der Aufgaben zu Mensch und Roboter, die Programmierung des Roboters und insbesondere die Zertifizierung der Applikation bzw. die Rezertifizierung bei Änderung, bestehen. Um das Potenzial einer flexiblen Automatisierung durch die Mensch-Roboter-Kooperation ausnutzen zu können, muss diesen Herausforderungen begegnet werden.

### **1.2 Zielsetzung der Arbeit und Betrachtungsbereich**

Daher verfolgt die Arbeit das Ziel, ein System zur Programmierung des Roboters für die Mensch-Roboter-Kooperation zu entwickeln, das den Nutzer zusätzlich bei der Planung der Applikation und der Zertifizierung unterstützt. Ein Planungsmodul generiert einen Vorschlag zur Zuteilung der Aufgaben zu den Ressourcen und ein Konzept für die Sicherheit unterstützt bei der Zertifizierung. Informationen, die für die Programmierung genutzt werden, werden außerdem verwendet, um die Schnittstellen zwischen den Ressourcen abzubilden. Für den Betrieb soll das Programm den Anforderungen an eine Zusammenarbeit zwischen Mensch und Roboter, wie vorhersehbaren Bewegungen, gerecht werden.

Als Zielstellung der Arbeit wird daher ein System zur vereinfachten *Programmierung für die Mensch-Roboter-Kooperation* abgeleitet, die angrenzende Fragestellungen, wie die Zertifizierung und eine Unterstützung bei der Aufgabenplanung, einschließt (vgl. Abbildung 2). Des Weiteren beinhaltet die Zielstellung, dass Informationen, die bei der Programmierung für ein Robotersystem genutzt werden, auch im Betrieb als Schnittstelle zwischen Mensch und Roboter zur Abbildung der Abhängigkeiten untereinander gelten sollen.

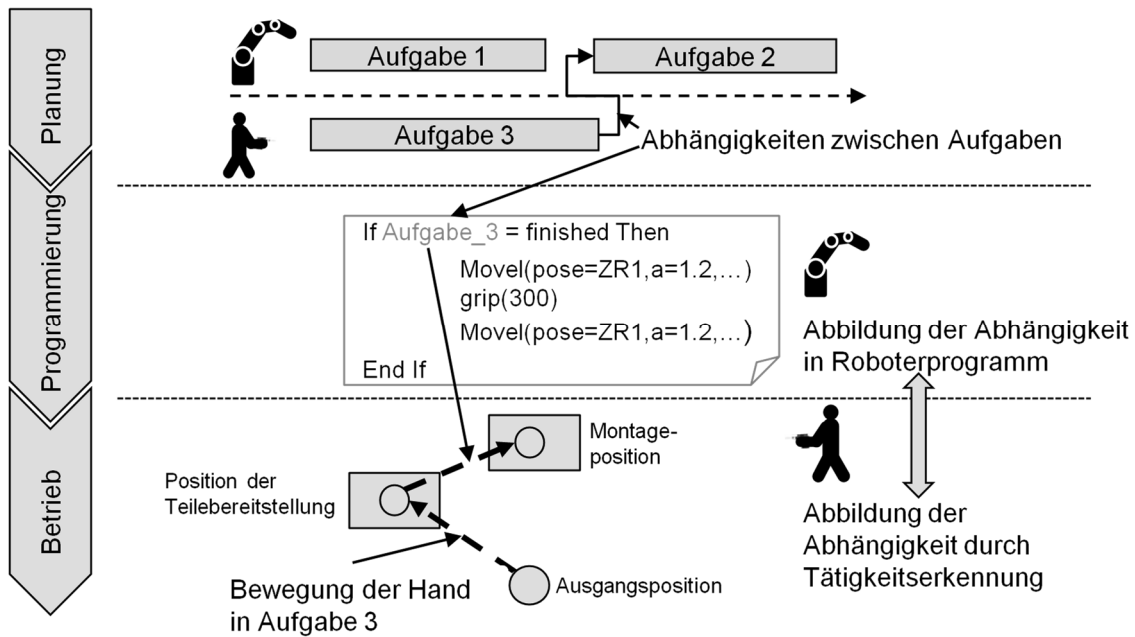


Abbildung 2: Zielbild der Arbeit

Aufgrund der Kopplung zwischen Mensch und Roboter unterscheidet sich das Roboterprogramm in einer Mensch-Roboter-Kooperation von dem in einer konventionellen Anwendung. Daher ergeben sich auch Unterschiede für das Programmiersystem. Der Unterschied bei der Programmierung des Robotersystems für eine MRK besteht darin, dass nicht nur die Aufgaben des Roboters isoliert betrachtet und in ein Programm umgewandelt werden können, sondern dass auch die Aufgaben des Menschen und vor allem sich daraus ergebende Abhängigkeiten bei der Programmierung einbezogen werden müssen. So liegt die Herausforderung darin zu identifizieren, an welchen Stellen des Programmiersystems der Mensch integriert werden muss. Daher ist der Fokus der Arbeit vor allem auf die Integration des Menschen in die unterschiedlichen Bereiche der aufgabenorientierten Programmierung gerichtet.

Bei den betrachteten Anwendungen handelt es sich insbesondere um Montageaufgaben, bei denen der Großteil noch manuell erfolgt. Abhängigkeiten zwischen den Aufgaben ergeben sich aus deren Anordnung auf der Benutzerschnittstelle und werden darauf basierend im weiteren Verlauf verarbeitet. Bereits die Zuteilung der Aufgaben zu den beiden Ressourcen ist eine Herausforderung, da unklar ist, welche der beiden Ressourcen die Aufgaben übernehmen kann und soll. Die Zuteilung bewegt sich dabei in einem Spannungsfeld, das geprägt wird von Aspekten der Sicherheit und der Produktivität, aber auch von psychologischen Komponenten, da versucht wird, die Aufgaben für den Menschen

bestmöglich aufzuteilen. Da dies die anwendende Person des Systems vor Herausforderungen stellt, soll der Vorgang durch ein Planungssystem unterstützt werden, das eine Aufgabenzuteilung vorschlägt, die entweder entsprechend des Vorschlags verwendet oder angepasst werden kann.

Die Zuteilung der Aufgaben zu den beiden Ressourcen auf einer Benutzerschnittstelle beinhaltet bereits einige Informationen, die für die Generierung des Roboterprogramms verwendet werden können. Da die Aufgaben dabei für beide Ressourcen die gleichen Informationen beinhalten müssen, bietet es sich an, diese Informationen als Vorbereitung für den Betrieb zu nutzen, wo die Abhängigkeiten zwischen den Aufgaben in Erscheinung treten. Die Abhängigkeiten werden in der Struktur des Roboterprogramms vorgesehen und mit einem System zur Erkennung der menschlichen Tätigkeiten verknüpft. Eine Abhängigkeit bedeutet in diesem Fall, dass, basierend auf dem Montagevorranggraphen, eine bestimmte Aufgabe nicht vor Beendigung der vorherigen Aufgabe begonnen werden kann. Muss der Mensch auf die Beendigung einer Aufgabe des Roboters warten, so kann er dies sehen. Ein Roboter hingegen kann dies nicht unmittelbar feststellen, weshalb eine Lösung erarbeitet werden muss, wie der Roboter dies detektieren kann. Hierfür wurden in der Literatur bisher kamerabasierte Systeme in Verbindung mit Wahrscheinlichkeitsmodellen verwendet, mit denen Tätigkeiten eines Menschen identifiziert werden können. Mit den Informationen zu den Aufgaben aus der Benutzerschnittstelle wird ein Modell für die Erkennung von Tätigkeiten im Betrieb konfiguriert.

Um eine Applikation der Mensch-Roboter-Kooperation flexibel einsetzen zu können, muss der Vorgang zur Zertifizierung mitbetrachtet werden. Deshalb werden in der Arbeit Strategien entwickelt, wie die Zertifizierung eines solchen Systems mit weniger Aufwand erfolgen kann.

### **1.3 Aufbau der Arbeit**

Die Basis der Arbeit ist die praxisrelevante Fragestellung, wie die Programmierung des Roboters für eine Mensch-Roboter-Kooperation gestaltet werden muss. Bei Betrachtung der Wissenschaftssystematik nach ULRICH & HILI (1976) handelt es sich dabei um die angewandten Handlungswissenschaften, die den Realwissenschaften zuzuordnen sind. Ausgehend von der Fragestellung wurde im Rahmen der Arbeit über eine Kombination aus kreativem und synthetischem Ansatz ein prototypischer Aufbau für ein System zur aufgabenorientierten Programmierung für die Mensch-

Roboter-Kooperation erarbeitet. Dieser wurde auf seine Funktion geprüft und evaluiert.

Abbildung 3 stellt den Aufbau der Arbeit dar. Im Anschluss an die Motivation und Zielsetzung im ersten Kapitel werden im zweiten Kapitel für die Arbeit relevante Begriffe sowie Grundlagen, die zum Verständnis der Arbeit beitragen, erläutert. Im Stand der Wissenschaft und Technik wird auf Ansätze zur Programmierung und zu Arbeiten im Bereich der Mensch-Roboter-Kooperation eingegangen, die die Themen der Planung, der Simulation und des Betriebs beleuchten. Das vierte Kapitel fasst die Anforderungen an das Programmiersystem zusammen, dessen Konzept in Kapitel fünf beschrieben ist. Das Kapitel fünf umfasst den Gesamtüberblick über das System und die zum Betrieb des Programmiersystems notwendigen Nebenmodule. Diese umfassen ein Aufgaben- und Umweltmodell sowie ein Wissensmodell. Die Hauptmodule sind ein Planungs-, ein Programmier- und ein Betriebsmodul und werden in Kapitel 6 beschrieben. Daran anschließend wird im siebten Kapitel ein Vorgehensmodell für die Einführung des Programmiersystems in den Betrieb vorgestellt. Im achten Kapitel erfolgt die Beschreibung der Umsetzung des Programmiersystems. Abschließend wird das System wirtschaftlich und technisch bewertet sowie die Arbeit mit einer Zusammenfassung und einem Ausblick abgeschlossen.

1	Einleitung
2	Begriffsbestimmung und Grundlagen
3	Stand der Wissenschaft und Technik
4	Anforderungsanalyse an ein Programmiersystem für die Mensch-Roboter-Kooperation
5	Grundstruktur und Nebenmodule des Programmiersystems
6	Hauptmodule des Programmiersystems
7	Vorgehen zur Integration des aufgabenorientierten Programmiersystems in den Betrieb
8	Umsetzung und Erprobung
9	Wirtschaftliche und technische Bewertung
10	Zusammenfassung und Ausblick

Abbildung 3: Aufbau der Arbeit

## 2 Begriffsbestimmung und Grundlagen

### 2.1 Übersicht

Dieses Kapitel beschreibt für die Arbeit relevante Grundlagen. Zunächst werden dazu Begriffsdefinitionen aufgeführt bzw. Begriffe definiert. Anschließend daran werden Grundlagen in den Bereichen Montage, Mensch-Roboter-Kooperation und der Programmierung von Robotern erläutert.

### 2.2 Begriffsdefinitionen

Zu den nachfolgend beschriebenen Begriffen gibt es in der Literatur teilweise diverse Definitionen in unterschiedlichen Zusammenhängen, die im Vorfeld diskutiert und im Rahmen der Arbeit verglichen wurden. Bei den aufgeführten Definitionen handelt es sich um die für den Kontext der Arbeit zutreffendste.

#### **Aufgabe**

Nach HAMMERSTINGL & REINHART (2017) entspricht eine Aufgabe einem Prozessablauf, der aus einer Sequenz von elementaren Prozessschritten zusammengesetzt ist. Die Art der Anwendung dieses Begriffs in dieser Arbeit wird in Abschnitt 5.3 beschrieben.

#### **Aufgabenorientierte Programmierung**

Im Gegensatz zur roboterorientierten Programmierung, bei der spezifiziert wird, wie der Roboter eine Aufgabe lösen soll, wird bei der aufgabenorientierten Programmierung nur beschrieben, was der Roboter lösen soll. Die Formulierung der Aufgaben erfolgt dabei auf einer Abstraktionsebene, die über den Programmiersprachen liegt (HAUN 2013).

#### **Basisbauteil und Fügebauteil**

ROSS (2002) verwendet in seiner Arbeit den Begriff Fügebauteil (FBT) für das Bauteil, das an das Basisbauteil gefügt wird. Das Fügebauteil ist daher das Bauteil, das bewegt wird, und das Basisbauteil (BBT) das Bauteil, an das montiert wird.

#### **Fähigkeit**

HAMMERSTINGL & REINHART (2017) beschreiben eine Fähigkeit als eine „durch ein Gerät angebotene lösungsneutrale Funktionalität, welche aus einer

aufgabenspezifischen Sicht (Prozesssicht) beschrieben ist“. Fähigkeiten „besitzen ein spezifisches Set an Ein- und Ausgangsparametern und können automatisiert mit Prozessschritten (und deren Anforderungen) abgeglichen werden“ (HAMMERSTINGL & REINHART 2017, S. 6).

### **Fügehilfsteil**

Bei Fügehilfsteilen handelt es sich beispielsweise um Schrauben oder Nieten (WARNECKE 1995). Verallgemeinert sind es Teile zum Verbinden der Fügebauteile.

### **Leichtbauroboter**

Der Begriff des Leichtbauroboters (LBR) wurde normativ noch nicht definiert. ALBU-SCHÄFFER ET AL. (2007) haben den Begriff im Rahmen der Entwicklung ihres Roboters eingeführt, der mit interner Sensorik ausgestattet ist, um Kollisionen zu vermeiden. Nach FRUTIG (2013) weisen Leichtbauroboter fünf Merkmale auf:

- Masse von bis zu 30 kg, so dass sie von einer Person getragen werden können
- Teach-In der Positionieraufgaben per Hand
- Einfache Programmierung
- Adaptive Sensorik
- Einhaltung der EN ISO 10218-1:2006 für den kollaborativen Betrieb

### **Mensch-Roboter-Kooperation**

Nach THIEMERMANN (2005) ist eine direkte Mensch-Roboter-Kooperation (MRK) über einen überlappenden Arbeitsraum zwischen Mensch und Roboter definiert. Da dieser Ausdruck in der Literatur unterschiedlich definiert ist, jedoch einen Kernaspekt dieser Arbeit darstellt, erfolgt eine weitergehende Diskussion und die Erläuterung der Verwendung des Begriffs in dieser Arbeit im Rahmen der Einführung in die Mensch-Roboter-Kooperation in Kapitel 2.4.1.

### **Montage**

Die Aufgabe der Montage ist es, aus zuvor in verschiedenen Fertigungsverfahren hergestellte Einzelteile zu einem Produkt höherer Komplexität mit vorgegebenen Funktionen in einer bestimmten Zeit zusammenzubauen (LOTTER & WIENDAHL 2006).

### **Planung von Montagesystemen**

„Die Zielsetzung bei der Planung und Gestaltung eines Montagebereiches für ein bestimmtes Erzeugnis oder einer Erzeugnisfamilie ist, die Arbeitsinhalte nach mengenabhängigen – variantenabhängigen – fügetechnischen – organisatorischen und ergonomischen Kriterien so zu gestalten, dass eine möglichst „flexible Montagestruktur“ entsteht“ (KONOLD & REGER 2003, S. 32).

### **Roboterprogrammierung**

Da sich diese Arbeit vorrangig mit der Programmierung von Robotersystemen befasst, wird der Begriff auch für diesen Anwendungsbereich definiert. WECK & BRECHER (2006) definieren ihn wie folgt: „Bei der Roboterprogrammierung wird eine Sequenz von Steuerinformationen festgelegt, die zur Ausführung der gestellten Automatisierungsaufgabe erforderlich sind“ (WECK & BRECHER 2006, S. 378).

### **Simulation**

Nach der VDI-Richtlinie 3633 handelt es sich bei Simulation um folgenden Sachverhalt: "Simulation ist die Nachbildung eines dynamischen Prozesses in einem Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind“ (VDI RICHTLINIE 3633).

### **System**

Nach REFA BUNDESVERBAND E.V. (2002) ist eine allgemein anwendbare Definition des Begriffs System: „Ein System ist eine abgegrenzte Gesamtheit von Elementen, zwischen denen bestimmte Beziehungen bestehen oder hergestellt werden können.“ (REFA BUNDESVERBAND E.V. 2002, S. 64) Dabei kann, je nach Einbindung des Menschen in das System, zwischen sozialen, technischen und soziotechnischen Systemen unterschieden werden.

### **Szenario**

Nachdem der Begriff Szenario in verschiedenen Kontexten unterschiedlich definiert ist, fassen ULBRICH ET AL. (2015) den Begriff in einer Definition zusammen. Ihre Definition lautet: „Ein Szenario beschreibt die zeitliche Entwicklung von Szenenelementen innerhalb einer Folge von Szenen, welche mit einer Startszene beginnt. Aktionen und Ereignisse ebenso wie Ziele und Werte können spezifiziert werden, um diese zeitliche Entwicklung in einem Szenario festzulegen" (ULBRICH ET AL. 2015, S. 10). Übertragen auf diese Arbeit beschreibt ein Szenario eine mögliche Abfolge der Aufgaben zur Montage eines Produkts, wobei die Aufgaben auf die

unterschiedlichen Akteure, Mensch und Roboter, aufgeteilt und von ihnen abgearbeitet werden.

### **Tätigkeitserkennung**

Ein Begriff zur Identifikation durchgeführter Aktionen eines Menschen ist in Deutschland kaum geprägt. Aus dem englischen Begriff *activity recognition* kann jedoch eine Definition abgeleitet werden. Nach KIM ET AL. (2010) ist das Ziel einer Tätigkeitserkennung die Identifikation von menschlichen Aktivitäten unter realen Bedingungen. In dieser Arbeit wird der Begriff, angelehnt an KIM ET AL. (2010), dafür verwendet, durch den Menschen durchgeführte Bewegungen zu detektieren, auszuwerten und damit auf die aktuell ausgeführte Tätigkeit zu schließen.

### **Tool Center Point**

Der Tool Center Point (TCP) beschreibt einen Punkt an der Hand bzw. am Endeffektor des Roboters. Mit Hilfe dieses Punktes beschreibt der Nutzer die Bewegungen in einem benutzerspezifischen Koordinatensystem (CRAIG 2005).

### **Phasen bei der Planung und dem Betrieb von Montagesystemen**

In der Betrachtung des Lebenszyklus von Produktionssystemen definieren KOŠTURIK & GREGOR (1999) vier Phasen: Systemanalyse, Planung, Implementierung und Betrieb. Die Phase der Systemanalyse umfasst Machbarkeitsstudien, Strategien und das Produktionsprogramm. In der Planungsphase werden die Dimensionierung, die Planung der Maschinen, das Steuerungskonzept, die Evaluation von Planungsparametern und die Layoutplanung bearbeitet. Bei der Implementierung stehen die Simulation und die Schulung im Vordergrund und im Betrieb steht unter anderem die Überwachung im Fokus.

Hieran angelehnt werden in dieser Arbeit die Phasen Planung, Programmierung und Betrieb für eine Applikation der Mensch-Roboter-Kooperation in der Montage betrachtet. Die erste genannte Phase mit der Systemanalyse wird für diese Arbeit als abgeschlossen angesehen, weshalb mit der Planungsphase begonnen wird. An dieser sowie den beiden nachfolgenden Phasen Implementierung und Betrieb orientiert ich die Aufbereitung des Standes der Technik und der Hauptteil der Arbeit.

## **2.3 Montage**

Da der Betrachtungsbereich der Arbeit in der Montage liegt, wird in diesem Abschnitt eine Einführung zur Montage gegeben. Die Montage bildet das Bindeglied zwischen



Teilefertigung und Vertrieb (FELDMANN 1992). Nach LOTTER & WIENDAHL (2006) setzen sich die Funktionen in der Montage, wie in nachfolgender Abbildung 4 dargestellt, zusammen.

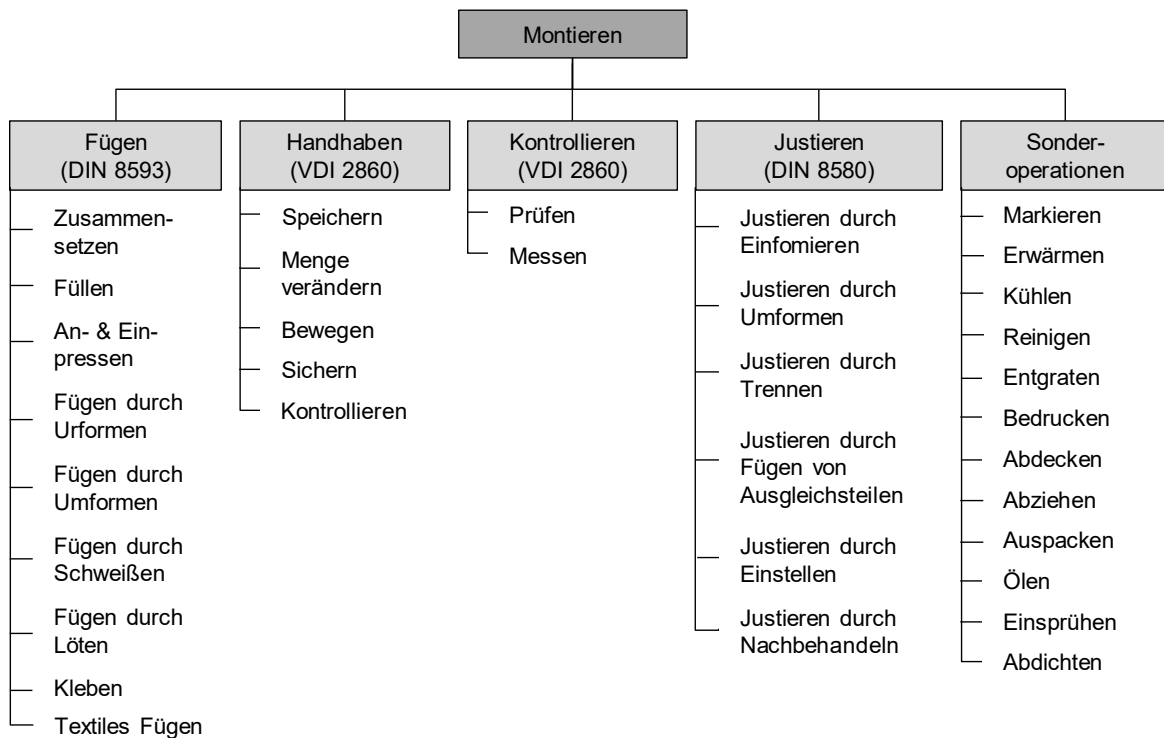


Abbildung 4: Funktionen der Montage nach LOTTER & WIENDAHL (2006)

Die Hauptaufgaben in der Montage bestehen aus den Fügevorgängen, vgl. DIN 8593, und Handhabungsvorgängen. Daneben gehören noch das Kontrollieren und Justieren sowie Sonderoperationen zur Montage (LOTTER & WIENDAHL 2006).

### 2.3.1 Arten von Montagesystemen

Aufgrund der volatilen Anforderungen an die Montage ist die Flexibilität von Montagesystemen notwendig. Nach LOTTER & WIENDAHL (2006) sollten flexible Montageanlagen folgende Eigenschaften aufweisen:

- Die Montagesysteme müssen die Komplettmontage einer Produktfamilie ermöglichen, was durch die Umprogrammierung der Montageanlage erfolgen kann.
- Die Montageanlagen müssen auf ein anderes Produkt umgerüstet werden können.
- Die Komponenten eines solchen Montagesystems sollten wiederverwendbar sein.

Nach LOTTER & WIENDAHL (2006) kann dabei zwischen manuellen und automatisierten Montagesystemen unterschieden werden. Eine Mittelstellung zwischen automatisierten und manuellen Montagesystemen nehmen die hybriden Montagesysteme ein, wobei Automatikstationen mit Handarbeitsplätzen kombiniert werden. Diese ermöglichen die wirtschaftliche Montage im mittleren Stückzahlbereich. Aufgrund der höheren Flexibilität gewinnen diese Systeme hinsichtlich der zunehmenden Produktvarianz an Bedeutung.

Die Mensch-Roboter-Kooperation als gemeinsames Arbeiten von Mensch und Roboter kann folglich zu den hybriden Montagesystemen gezählt werden (SCHRÖTER 2018). Durch die Halbautomatisierung bietet sie damit eine höhere Flexibilität als die automatisierte Montage, liegt in der Produktivität jedoch unterhalb der automatischen Montage (LOTTER & WIENDAHL 2006).

### **2.3.2 Planung von Montagesystemen**

BULLINGER & AMMER (1986) beschreiben in ihrem Vorgehen zur Montageplanung fünf Phasen. Die erste Phase wird dabei als Vorplanung bezeichnet und umfasst die Definition des organisatorischen Rahmens, die Ziele des Vorhabens und die Verantwortlichkeiten. In der anschließenden Grobplanung wird der Montageplan mit einer Reihenfolge der Montageschritte und deren Teilverrichtungen erstellt. In dieser Phase wird überprüft, ob eine Automatisierung von Montageschritten möglich ist. In der Feinplanung werden die Arbeitsplätze detailliert beschrieben und Montageunterlagen dazu erstellt. Darauf folgt die vierte Phase, die die Betriebsmittelbeschaffung, die Personaleinsatzplanung und den Systemaufbau umfasst. Bei der letzten Phase des Vorgehens handelt es sich um den Betrieb.

Ähnlich wie das Vorgehen zur Montageplanung nach BULLINGER & AMMER (1986) ist das Vorgehen zur Montageplanung nach REFA BUNDESVERBAND E.V. (2002). Dieses untergliedert sich in sechs Phasen. Zusätzlich zu den fünf Phasen von BULLINGER & AMMER (1986) beschreibt REFA noch eine vorgelagerte Phase, in der die Ausgangssituation analysiert wird. Daran schließen die Schritte Konkretisierung der Planungsaufgabe, Grobplanung des Produktionssystems, Feinplanung des Produktionssystems, Systemeinführung und Systembetrieb an.

FELDMANN (1992) stellt ein 5-Schichten-Modell zur Planung vor, das parallel zur Konstruktion verläuft. Dabei basieren die Planungsphasen auf dem Zustand des Produkts beginnend mit der Produktstruktur, gefolgt von der Einzelgeometrie und Zusammenbauzeichnung bis hin zu den vollständigen Produktdaten. Bei diesem

Modell werden rechnergestützte Werkzeuge, wie CAD-System, 3D-Simulationssystem und ein Betriebsmittelkonfigurator, eingesetzt.

Die Planung hybrider Montagesysteme, wie die Mensch-Roboter-Kooperation, basiert auf der manuellen Montage, von der ausgehend ein geeigneter Automatisierungsgrad für die jeweilige Montageaufgabe zu ermitteln ist. Für die Aufteilung der Aufgaben zu Mensch und Roboter gilt es, dem Menschen nicht nur Resttätigkeiten zu überlassen und seinen Aufgabenbereich beispielsweise durch Prüfaufgaben zu bereichern. Bezüglich der Arbeitsraumgestaltung eines hybriden Montagesystems, insbesondere mit der Nutzung eines Roboters, gilt es, die Handhabungswege möglichst kurz zu halten (LOTTER & WIENDAHL 2006).

### **2.3.3 Methods-Time-Measurement**

Ziel in der Montageplanung ist es, möglichst wenige Bewegungen durchzuführen, die nicht der Wertschöpfung dienen. Zur Bewertung der Anteile wertschöpfender und nicht-wertschöpfender Bewegungen stellen LOTTER & WIENDAHL (2006) die Primär- und Sekundäranalyse zur Verfügung. Dabei werden die Zeitanteile beider Vorgänge miteinander verglichen. Zur Berechnung der Zeitanteile und auch zur effizienten Anordnung des Montagearbeitsplatzes werden häufig Systeme vorbestimmter Zeiten, zu denen die Methods-Time-Measurement (MTM) zählt, verwendet. Bei der MTM-Methode werden manuelle Montageabläufe in Elementarbewegungen zerlegt, denen jeweils Zeiten zugeordnet sind. Die fünf Elementarbewegungen, mit denen nach LOTTER & WIENDAHL (2006) 85 % der Montagetätigkeiten beschrieben werden können, sind: Hinlangen, Greifen, Bringen, Fügen und Loslassen. Darüber hinaus werden Körper-, Bein- und Fußbewegungen definiert.

Für jede Grundbewegung gibt es Einflussgrößen, die den Zeitanteil der jeweiligen Bewegung beeinflussen und zu einem anderen Wert führen. So sind die Einflussgrößen beim Hinlangen beispielsweise die Bewegungslänge, der Bewegungsfall, der Typ des Bewegungsverlaufs und ob die Bewegung aus der ruhenden oder der sich bewegenden Hand stattfindet. Beim Greifen sind die Einflussgrößen die Art des Greifens, die Lage des Gegenstands und seine Beschaffenheit bzw. Abmessungen. Mit Hilfe dieser Informationen und beim Hinlangen und Bringen basierend auf der Bewegungslänge werden aus den Tabellen die entsprechenden Zeiten entnommen (REFA BUNDESVERBAND E.V. 2002).

Analog zur MTM stellten CHOI & IP (1999) ein System zur Berechnung von Bewegungszeiten von Robotern zur Verfügung. Da dieses allerdings für

konventionelle Industrieroboterumgebungen konzipiert war, ist es für die Mensch-Roboter-Kooperation kaum anwendbar. SCHRÖTER (2018) greift dieses Vorgehen auf und entwickelt eine Methode zur Berechnung der Zeiten von Roboterbewegungen für die MRK basierend auf Prozessbausteinen.

In Bezug auf den Einsatz von Robotern und der Berechnung von Zeiten schlagen LOTTER & WIENDAHL (2006) vor, Greiferwechsel als Sekundäraufwand zu zählen. Daher empfehlen sie, Mehrfachgreifer zu verwenden oder einen satzweisen Montageablauf vorzusehen, um die Umrüstzeiten zu verringern.

### **2.4 Formen der Mensch-Roboter-Kooperation**

Die in dieser Arbeit betrachtete Mensch-Roboter-Kooperation beschreibt die Zusammenarbeit zwischen einem Menschen und einem stationären Leichtbauroboter, der für diese Zusammenarbeit konzipiert und in Abschnitt 2.4.3 beschrieben ist. Zunächst wird ein kurzer Überblick über die Verwendung des Begriffs *Mensch-Roboter-Kooperation* gegeben, bevor auf die Gründe zum Einsatz der MRK eingegangen wird.

#### **2.4.1 Begrifflichkeiten und Definition im Zusammenhang mit der Mensch-Roboter-Kooperation**

Die verschiedenen Arten der Zusammenarbeit zwischen Mensch und Roboter werden in der Literatur unterschiedlich definiert. THIEMERMANN (2005) beschreibt in seiner Arbeit drei Betriebsmodi für die Zusammenarbeit, die abhängig von der örtlichen und zeitlichen Trennung sind. Dabei wird die örtlich getrennte Arbeit als autark, die zeitliche Trennung als synchronisiert und eine Zusammenarbeit ohne örtliche und zeitliche Trennung als kooperierend beschrieben (THIEMERMANN 2005).

ONNASCH ET AL. (2016) definieren in ihrer Taxonomie für die Mensch-Roboter-Interaktion die drei Arten Koexistenz, Kooperation und Kollaboration. Bei der Koexistenz treffen Mensch und Roboter nur sporadisch aufeinander und verfolgen kein gemeinsames Ziel. Im Gegensatz dazu verfolgen die Teilnehmer sowohl bei der Kooperation als auch der Kollaboration ein gemeinsames Ziel. Die intensivste Art der Zusammenarbeit ist die Kollaboration, wobei die beiden Ressourcen auch gleichen Unterzielen nachgehen und die Teilaufgaben gemeinsam bearbeitet werden.

In einer Studie des Fraunhofer IAO wird zu den drei Ausprägungen Koexistenz, Kooperation und Kollaboration noch die Synchronisation ergänzt, wie sie auch bei

THIEMERMANN (2005) beschrieben wird. Bei einem synchronisierten Ablauf arbeiten Mensch und Roboter im gleichen Arbeitsraum, jedoch nicht zum gleichen Zeitpunkt. Die Kooperation wird beschrieben als Arbeit im gleichen Raum zur gleichen Zeit, allerdings nicht am gleichen Bauteil und Produkt. Dies ist demgegenüber bei der Kollaboration der Fall (FRAUNHOFER IAO 2016).

Wie aus diesem Abschnitt hervorgeht, gibt es keine einheitliche Definition der Arten der Zusammenarbeit. Für diese Arbeit wird daher die Unterscheidung in Koexistenz, Kooperation und Kollaboration verfolgt. Der Fokus der Arbeit liegt im Bereich der Mensch-Roboter-Kooperation, die sich auf die Zusammenarbeit eines Menschen mit einem sogenannten Leichtbauroboter, wie er in Absatz 2.4.3 definiert ist, bezieht.

### **2.4.2 Gründe für die Mensch-Roboter-Kooperation**

Ein Einsatz der Mensch-Roboter-Kooperation ermöglicht die Kombination der Stärken und Schwächen der jeweiligen Ressource Mensch und Roboter. Der Roboter überzeugt beispielsweise durch seine gute Wiederholgenauigkeit, die damit verbundene hohe Qualität und seine Ausdauer. Dem stehen beim Menschen vor allem die sensorischen und feinmotorischen Fähigkeiten und die Adaption an wechselnde Arbeitsaufgaben als Stärken gegenüber (BEUMELBURG 2005).

Diese unterschiedlichen Stärken und Schwächen der beiden Ressourcen erlauben die Aufteilung der Aufgaben entsprechend ihren Fähigkeiten (GERKE 2015). DELANG ET AL. (2017) teilen die Gründe für MRK danach auf, ob als Ausgangsbasis eine manuelle oder automatisierte Montagestation vorliegt. Als Gründe für den Einsatz von MRK basierend auf einer manuellen Montage nennen sie, dass die MRK einen Beitrag zur Gesundheit, zur Verringerung der Kosten oder Erhöhung der Geschwindigkeit liefert. Ausgehend von einer Automatisierungslösung sind die Gründe Verfügbarkeit, Flexibilität und das Einsparen von Platz relevant.

Eine Zusammenfassung einiger Gründe für die Mensch-Roboter-Kooperation stellt THIEMERMANN (2005) vor, der Potenziale der MRK aus Expertengesprächen heraus identifizierte. Diese sind in Abbildung 5 dargestellt.

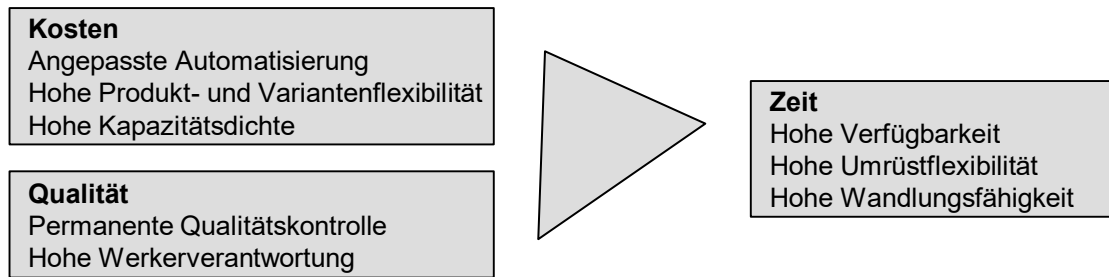


Abbildung 5: Potenziale der Mensch-Roboter-Kooperation  
nach THIEMERMANN (2005)

### 2.4.3 Leichtbauroboter

Wie in Abschnitt 2.4.1 dargelegt, soll sich die MRK in dieser Arbeit auf den Einsatz eines Leichtbauroboters beziehen. Seit der Einführung des ersten Leichtbauroboters ist die Anzahl an Leichtbaurobotern auf dem Markt stark angestiegen (ROBOTIK UND PRODUKTION 2018). Die meisten derzeit vertriebenen Modelle verfügen über sechs oder sieben Achsen. Zur Kraft- und Leistungsbegrenzung werden entweder Kraft-Momenten-Sensoren in allen Gelenken (KUKA AG 2019; FRANKA EMIKA GMBH 2018) oder am Fuß des Roboters (FANUC DEUTSCHLAND GMBH 2019) eingesetzt. Außerdem werden elektrische Ströme zur Kraftmessung verwendet (UNIVERSAL ROBOTS A/S 2018). Neben der leichten Bauweise werden die Roboter mit einem Design versehen, das Scher- und Klemmstellen am Roboter ausschließt (KUKA AG 2019).

### 2.4.4 Sicherheit in der Mensch-Roboter-Kooperation

Die Sicherheit des Menschen in einer Zusammenarbeit mit einem Roboter im gleichen Arbeitsraum ist die Voraussetzung für den Einsatz einer Mensch-Roboter-Kooperation (MAURTUA ET AL. 2017; DELANG ET AL. 2017; HEINZMANN & ZELINSKY 2016). Durch die Risikobeurteilung der gesamten integrierten Applikation wird die Sicherheit in industriellen Applikationen gewährleistet (VDMA ROBOTICS + AUTOMATION 2016). In DIN ISO 10218 Teil 1 und 2 werden sowohl die Vorgehensweise zur Beurteilung als auch die Anforderungen an die Sicherheit von Robotern beschrieben (DIN ISO 10218-1; DIN ISO 10218-2; VDMA ROBOTICS + AUTOMATION 2016;).

In DIN ISO 10218-1 sind die Anforderungen an Industrieroboter dargelegt. Diese Norm wurde im Zuge der Entwicklung der Mensch-Roboter-Kooperation um Abschnitte erweitert, die die hier zum Einsatz kommenden Roboter betreffen. Dies

bezieht sich vor allem auf die Kraft- und Leistungsbegrenzung (DIN ISO 10218-1). Der zweite Teil der Norm beschreibt die Integration der Robotersysteme. Die ISO TS 15066 definiert vier Arten zum Betrieb von Robotersystemen. Diese sind Handführung, sicherheitsgerichteter Stopp, Geschwindigkeits- und Abstandsüberwachung und Kraft- und Leistungsbegrenzung. Der schutzzaunlose Betrieb wird nach FACHBEREICH HOLZ UND METALL DER DGUV (2017) durch die Kraft- und Leistungsbegrenzung ermöglicht. Diese Art der Zusammenarbeit wird im Rahmen dieser Arbeit betrachtet.

Die Gefährdung des Menschen entsteht bei Kontakt zwischen Mensch und Roboter, wobei zwischen transientem und quasistatischem Kontakt zu unterscheiden ist. Bei einem transienten Kontakt handelt es sich um einen freien Kontakt, bei dem der Mensch nicht eingeklemmt wird. Quasistatischer Kontakt besteht bei Einklemmen des Menschen (VDMA ROBOTICS + AUTOMATION 2016). In einer Studie wurde untersucht, wieviel Kraft und Druck auf den Menschen in bestimmten Körperregionen einwirken kann, bis eine Verletzung eintritt (BEHRENS & ELKMANN 2014). Basierend auf diesen Ergebnissen wurde die ISO TS 15066 erstellt, die spezifische Anforderungen an einen kollaborierenden Betrieb beschreibt. Diese Spezifikation beinhaltet Hinweise zur Gefahrenreduktion und -eliminierung, Hinweise zur Reduzierung von Risiken und zur Gestaltung einer MRK-Applikation. Des Weiteren werden die Grenzwerte für Kraft und Druck für die verschiedenen Körperregionen festgelegt und das Vorgehen zur Prüfung dieser Grenzwerte beschrieben (ISO TS 15066). Auf Grundlage der Risikobewertung der gesamten Applikation muss eine CE-Kennzeichnung nach der Maschinenrichtlinie erfolgen; hiernach ist die Applikation einsatzbereit. Eine Änderung an der Applikation, z. B. die Produktion eines anderen Bauteils oder eine Anpassung des Roboterprogramms, erfordert eine erneute Risikobewertung (SIHN ET AL. 2016).

## **2.5 Grundlagen der Roboterprogrammierung**

Die Programmierung von Robotersystemen kann nach WECK & BRECHER (2006) oder HAUN (2013) in Online- und Offline-Programmierung eingeteilt werden. Beide Programmierarten unterscheiden sich in der Verfügbarkeit des Roboters. Bei der Offline-Programmierung steht der Roboter der laufenden Produktion weiterhin zur Verfügung, da die Programmierung abseits des Roboters erfolgt. Die Online-Programmierung erfolgt direkt am Robotersystem, weshalb dieser während der Programmierung nicht im Prozess verwendet werden kann (WEBER 2009).

Abbildung 6 zeigt die Einteilung der Programmierarten nach WECK & BRECHER (2006).

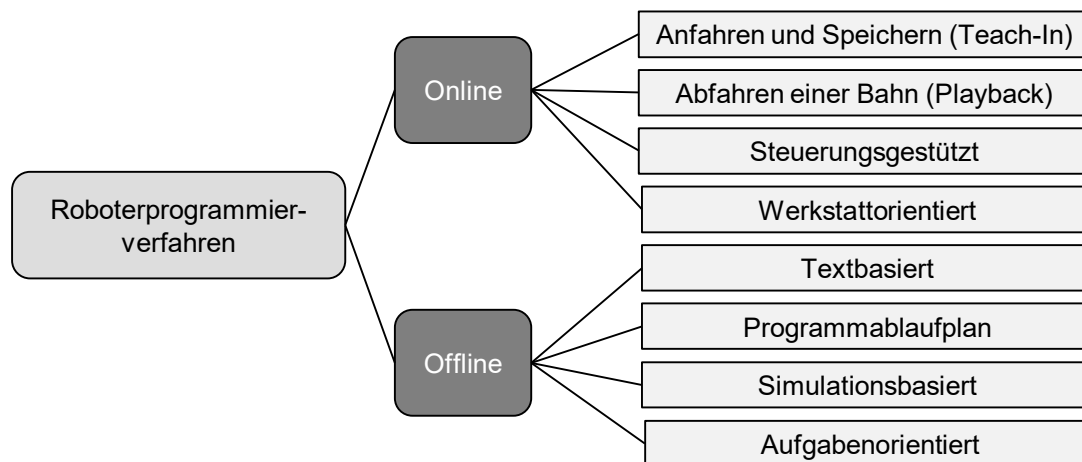


Abbildung 6: Arten der Programmierung von Robotern nach WECK & BRECHER (2006)

Die Verfahren Playback und Teach-In können auch als Lernprogrammierverfahren bezeichnet werden. Dabei führt der Nutzer entweder den Roboter direkt oder über eine Konstruktion, wobei die Daten zur Bewegung abgespeichert werden. Bei der Playback-Programmierung wird der Roboter geführt und die Daten der abgefahrenen Bahn werden abgespeichert und wiedergegeben. Während des Teach-In bewegt der Nutzer den Roboter, meist über ein Bediengerät, und fährt zu bestimmten Stellungen, die durch eine Eingabe am Bediengerät gespeichert werden. Anschließend erfolgt die Parametrierung, wie beispielsweise die Angabe der Geschwindigkeit, und die Programmierung von Funktionen, wie die Ansteuerung des Greifers. Im Betrieb werden die Stellungen abgefahren (WECK & BRECHER 2006). Bei der werkstatorientierten Programmierung wird der Programmierer durch Hilfsmittel, wie die grafische Darstellung von programmierten Bahnen, Automatismen oder die Nutzung von CAD-Daten, unterstützt. Diese Programmierart bietet viel Entwicklungspotenzial, wird jedoch bisher wenig eingesetzt (WECK & BRECHER 2006). Neben den in der Abbildung 6 dargestellten Arten beschreibt HAUN (2013) noch die Master-Slave-Programmierung. Diese ist dann von Vorteil, wenn ein Roboter in einer für den Menschen schwer zugänglichen Umgebung programmiert werden soll. Dazu wird ein Modell des Roboters, das den Master darstellt, mit dem zu programmierenden Roboter, dem Slave, verknüpft. Bewegungen des Master-Arms werden dann durch den Slave-Arm nachgeahmt (HAUN 2013).



Wird das Programm im Zuge der Arbeitsvorbereitung erstellt, handelt es sich um eine textbasierte Programmierung, die zu den Offline-Programmiersystemen zählt. Hierbei wird das Gerüst für das Programm ohne Positionswerte angefertigt und die Eingabe der Positionswerte erfolgt direkt am Roboter. Dies wird auch als hybride Programmierung bezeichnet (WECK & BRECHER 2006). In einem Programmablaufplan werden grafische Ablaufdiagramme dargestellt und das Programm über grafische Bausteine zusammengestellt. Die Parametrierung des Programms erfolgt über Eingabemasken, woraufhin der Programmcode automatisch generiert wird. Bei einer simulationsbasierten Programmierung wird ein erstelltes Roboterprogramm simuliert und dargestellt. Die Voraussetzung für diese Art der Programmierung ist die Modellierung der Roboterzelle (WECK & BRECHER 2006).

Eine weitere Art stellt die aufgabenorientierte Programmierung dar. Dabei werden für eine Aufgabe notwendige Bewegungen des Roboters basierend auf einer Beschreibung des Ist- und Sollzustandes einer Zelle oder eines Bauteils selbstständig generiert (WECK & BRECHER 2006). Nach HAUN (2013) werden die Aufgaben dafür auf abstrakter Ebene beschrieben, so dass vorgegeben wird, welche Aufgabe durchgeführt werden soll und nicht, wie diese Aufgabe auszuführen ist. Eine Aufgabe kann beispielsweise lauten: „Hole rote Schachtel  $S_{rot}$  und lege es in Maschine  $M_{Ziel}$  ein“ (HAUN 2013, S. 178). Da hierbei Bewegungen und Aktionen nicht explizit vorgegeben werden, sondern durch eine abstrakte Beschreibung nur impliziert werden, bezeichnen HAUN (2013) oder WEBER (2009) diese Art auch als *implizite Programmierung*. Der Vorteil dieses Systems liegt vor allem darin, dass ein Roboter damit von Nicht-Experten bedient und programmiert werden kann (WEBER 2009). Weitere Vorteile werden in der Vereinfachung einer Produktionsumstellung und damit der Erhöhung der Flexibilität einer Roboterzelle gesehen (BACKHAUS 2016; GOTTSCHALK 2000). Nach WEBER (2009) gibt es bisher Einzelfälle in der industriellen Praxis, aber die Gestaltung der Schnittstellen und Komponenten sind noch Gegenstand der Forschung.

## 2.6 Fazit

Das Kapitel der Grundlagen zeigt auf, dass aufgrund einer höheren Variantenanzahl flexiblere Montagesysteme gefordert sind. Dies können hybride Montagesysteme sein, wobei die Mensch-Roboter-Kooperation einen Fall dieser Systemart darstellt. Zur Planung von Montagesystemen wurden unterschiedliche Vorgehen aufgezeigt, die für die Anwendung in einer Mensch-Roboter-Kooperation angepasst werden

müssen. Zur Planung von Montagesystemen ist die MTM-Methode relevant, mit der Zeiten für Montagevorgänge berechnet werden können.

Die Begrifflichkeiten zur Mensch-Roboter-Kooperation zeigen unterschiedliche Definitionen für die Zusammenarbeit auf, wobei in dieser Arbeit der Begriff der Mensch-Roboter-Kooperation für die Arbeit von Mensch und Roboter in einem gemeinsamen Arbeitsraum verwendet wird. Die vorgestellten Leichtbauroboter bieten die Funktionalitäten für eine Zusammenarbeit zwischen Mensch und Roboter an, jedoch muss auch bei diesen Systemen die Sicherheit der Applikation durch eine Risikoanalyse gewährleistet werden. Im Bereich der Roboterprogrammierung gibt es unterschiedliche Arten, wobei die aufgabenorientierte Programmierung noch als Forschungsgegenstand, jedoch auch als verwendbar durch Nicht-Experten gilt.

Nach Darlegung der Grundlagen wird im nächsten Kapitel der Stand der Wissenschaft und Technik beleuchtet, um daraus die Handlungsfelder für die Arbeit abzuleiten. Da sich die Arbeit mit den drei Phasen Planung, Programmierung und Betrieb befasst, werden Arbeiten aus diesen Bereichen beschrieben. Des Weiteren wird auf die Sicherheit in der Mensch-Roboter-Kooperation eingegangen. Zunächst wird auf die Programmierung von Robotern eingegangen, bei der es bisher wenige Ansätze mit Bezug zur Mensch-Roboter-Kooperation gibt. Anschließend werden die Phasen der Planung und des Betriebs betrachtet.

## **3 Stand der Wissenschaft und Technik**

### **3.1 Ansätze zur Programmierung von Industrierobotern**

In Abschnitt 2.5 wurden die Grundlagen der Programmierung von Robotern aufgezeigt. Da in dieser Arbeit der Fokus auf der aufgabenorientierten Programmierung liegt, werden im nächsten Abschnitt einerseits Methoden dazu und andererseits weitere Ansätze, die sich mit vereinfachter Programmierung beschäftigen, dargelegt.

#### **3.1.1 Ansätze zur vereinfachten Programmierung in der Forschung**

Die Vereinfachung der Programmierung von Industrierobotern stellt einen Schwerpunkt in vielen Forschungsarbeiten im Bereich der Robotik dar. Ziel dabei ist es, dass der Nutzer zur Programmierung kein Expertenwissen aufweisen muss. Um dieses Ziel zu erreichen, gibt es Ansätze zur aufgabenorientierten Programmierung, Programmierung durch Teach-In oder eine automatisierte Programmierung bzw. Konfiguration von Roboterzellen, die in diesem Abschnitt beschrieben werden.

HUMBURGER (1998) stellt in seiner Arbeit ein aufgabenorientiertes Programmiersystem für Roboter vor. Ein Schwerpunkt seiner Arbeit liegt auf der Gestaltung und Implementierung von Wissensquellen, die für die Lösung von Problemen notwendiges Wissen zur Verfügung stellen. Diese Wissensquellen können Informationen zur Generierung von Bewegungsbahnen, aber auch Funktionen zur Modifikation von Ergebnisparametern enthalten. Zur Definition dieser Wissensquellen beschreibt er eine Methode.

In einer anderen Arbeit wurde die aufgabenorientierte Offline-Programmierung von KUGELMANN (1999) bearbeitet. Hierbei lag das Ziel vor allem in der Verbesserung der Bahnplanung durch ein genaueres 3D-Modell für die Programmgenerierung. Dazu wird die Bahn, die nach der Programmierung durch den Roboter abgefahren wird, in das Simulationsmodell zurückgeführt, wodurch dieses eine größere Realitätsnähe erreichen soll. Ein weiterer Fokus ihrer Arbeit liegt in der Greifplanung, wobei ein Greifer und der Griff abhängig vom Bauteil und dessen Greifflächen ausgewählt werden. Das Vorgehen ist iterativ gestaltet. Dabei wird ein Vorgang, wie beispielsweise die Auswahl des Greiftyps, so lange wiederholt, bis der anschließende Schritt den Anforderungen entspricht.

WECK & BRECHER (2006) zeigen eine Struktur zur aufgabenorientierten Programmierung auf, wobei die Elemente den drei Phasen Planung, Programmierung und Betrieb zugeordnet werden können (vgl. Abbildung 7).

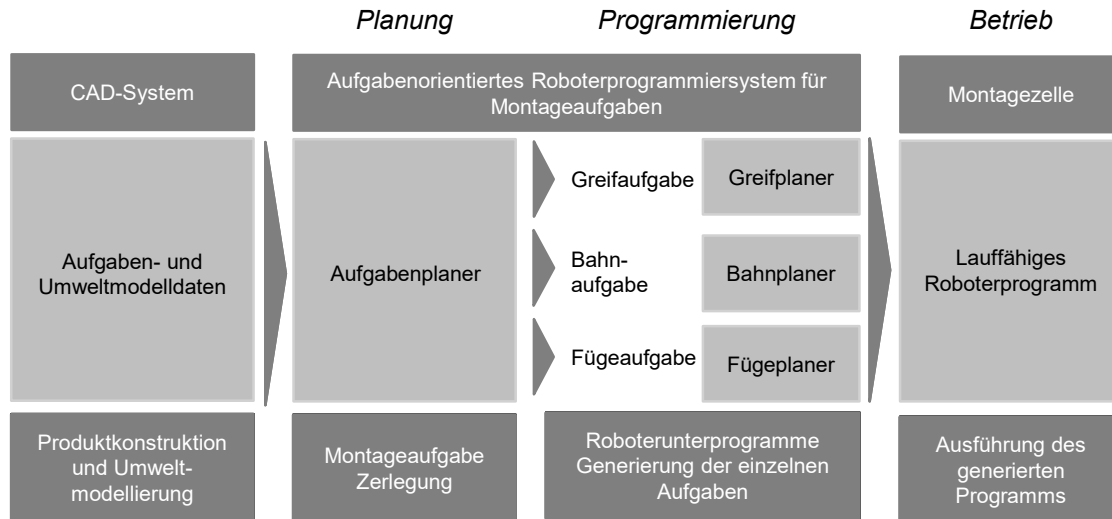


Abbildung 7: Struktur des aufgabenorientierten Programmiersystems nach WECK & BRECHER (2006)

Die Basis für die aufgabenorientierte Programmierung stellt die Repräsentation der Aufgaben- und Umweltdaten im CAD-System dar. Mit diesen Informationen kann im Montageplaner die Montageaufgabe zerlegt werden, so dass die Informationen durch Greif-, Bahn-, und Fügeplaner verarbeitet werden können. Ziel des Greifplaners ist es, die Handhabung eines Objekts zu planen, so dass der Transport gesichert ist und keine Beschädigungen am Werkstück auftreten. Der Greifplaner umfasst die Identifikation der Greifposition, die Greiferauswahl und die Orientierung des Greifers. Der Bahnplaner ist für die Generierung einer kollisionsfreien Roboterbewegung, basierend auf der Roboterumwelt, der Anfangs- und der Endkonfiguration, zuständig (WECK & BRECHER 2006).

EHRMANN (2007) präsentiert ein nutzergerechtes Programmiersystem für Montagezellen. Dabei werden die Komponenten der Montagezellen durch ein UML-Klassendiagramm beschrieben, wozu jeweils eine Basisklasse für Geräte und für die Steuerung mit ihren Klassen definiert wird. Ein Montagesystem wird durch eine Klasse desselben repräsentiert. Die Montagevorgänge werden durch einen Interaktionsbereich und eine Bewegung dargestellt, wobei der Interaktionsbereich durch eine Position und eine zugehörige Aktion definiert ist und Bewegungen durch Aktionen beschrieben und über die Start- und Endposition konfiguriert werden. Das Programmiersystem ermöglicht es dem Nutzer, Bedingungen und Verzweigungen

einzufragen. Über einen Postprozessor wird das herstellerspezifische Roboterprogramm erstellt, wozu die Informationen aus dem Interaktionsbereich und den Bewegungen in die spezifische Sprache überführt werden. Neben dem Programmablauf wird ein Hauptprogramm zur Initialisierung der Geräte, zur Aktualisierung der Positionsvariablen und zum Aufruf der Code-Module erstellt.

BACKHAUS (2016) stellt in seiner Arbeit ein adaptierbares aufgabenorientiertes Programmiersystem für wandlungsfähige Montageanlagen vor. Der Fokus der Arbeit liegt auf einem adaptierbaren Aufgabenmodell, das es ermöglicht, eine Aufgabe auf unterschiedlichen Abstraktionslevels anzugeben. Die Programmierung der technischen Systeme erfolgt über einen skill-basierten Ansatz. Die Übersicht der Systemarchitektur zeigt nachfolgende Abbildung 8.

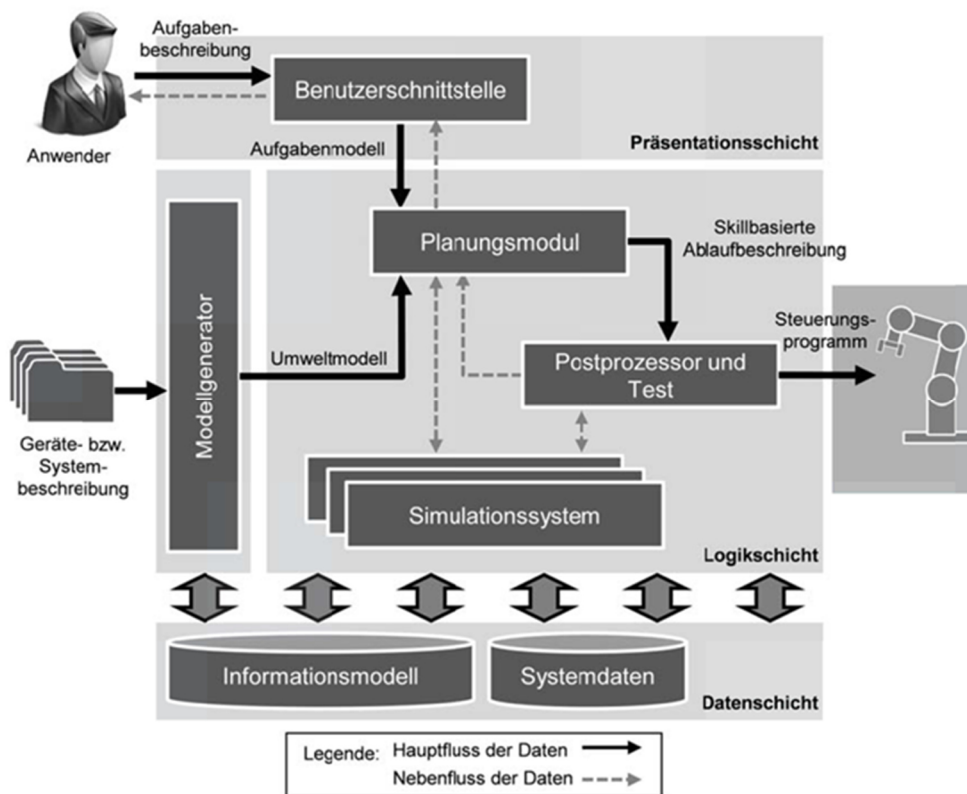


Abbildung 8: Architektur des aufgabenorientierten Programmiersystems von BACKHAUS (2016)

Über eine Benutzerschnittstelle kann der Nutzer über Drag & Drop eine Bearbeitungssequenz zusammenstellen. Es wird eine Blackboard-Architektur verwendet.

Das Ziel des EU-Forschungsprojekts SMERobot, das von 2005 bis 2009 bearbeitet wurde, war die Entwicklung von Robotern, die den Anforderungen von KMUs

entsprechen, um dort die Automatisierung mit Robotern zu erhöhen. Im Bereich der Programmierung wurden dabei verschiedene Ansätze, wie die Programmierung über Demonstration (PIRES ET AL. 2009), auf Basis von Prozess- und Gerätebeschreibungen (NAUMANN ET AL. 2010), über Gesten, Sprache und Kraftsteuerung (NETO ET AL. 2010b) oder mit Hilfe von 3D-CAD-Dateien (NETO ET AL. 2010), betrachtet.

Das Projekt SMERobotics bildete das Anschlussprojekt von SMERobot mit einer Laufzeit von 2012 bis 2016, wobei ähnliche Ziele verfolgt wurden. Dabei wurden ebenfalls Ansätze zur Roboterprogrammierung verfolgt, wie z. B. durch Teach-In (PERZYLO ET AL. 2015b) oder aufgabenbasiert (PERZYLO ET AL. 2015a). Außerdem wird an der Erkennung menschlicher Tätigkeiten gearbeitet, auf die in Abschnitt 3.3 noch näher eingegangen wird.

KRUG (2013) entwickelte ein System, mit dem Roboterzellen automatisch über einen Plug & Produce Ansatz konfiguriert werden. Dazu wurde eine Modularisierung der beteiligten Komponenten, wie Roboter und sonstiger Peripheriegeräte, vorgenommen und der notwendige Informationsfluss untersucht. Um die automatische Konfiguration des Roboters, der Peripheriegeräte sowie des Bussystems zu ermöglichen, wurde ein Zustandsmodell entwickelt. Ein fünfstufiges Vorgehensmodell fasst die Schritte vom Geräteanschluss bis zur abschließenden Konfiguration zusammen.

MICHNIEWICZ & REINHART (2016) stellen einen Ansatz zur Analyse, Programmierung und Konfiguration von Roboterzellen basierend auf Cyber-Physischen Systemen vor. Bei Cyber-Physischen Systemen handelt es sich um „[...] Systeme, die durch die hochgradige Vernetzung der physikalischen, sozialen und virtuellen Welt sowie durch die intelligente Nutzung von Informations- und Kommunikationstechnologien entstehen“ (GEISBERGER & BROY 2012, S. 17). In der Methode von MICHNIEWICZ & REINHART (2016) wird die Roboterzelle als Cyber-Physisches System abgebildet. Basierend auf einer 2D- oder 3D-Modellierung des Produkts werden die Bauteile und die zugehörigen Aufgaben abgeleitet und in einem sogenannten Augmented Assembly Priority Plan zusammengefasst. Für einen Vergleich zwischen den Anforderungen an eine Aufgabe und den angebotenen Fähigkeiten der Komponenten werden die Skills mit Hilfe von Funktionsprimitiva beschrieben. Zunächst überprüft die Cyber-Physische Roboterzelle, ob sie mit einer Konfiguration alle Anforderungen des Produkts erfüllen kann. Ist dies nicht möglich, so wird analysiert, ob die Anforderungen durch eine Rekonfiguration während des Montageprozesses erfüllt werden können. Bei mehreren Möglichkeiten wird die

optimale Konfiguration basierend auf Kriterien, die durch den Nutzer angegeben werden, ermittelt. Anschließend wird die Programmierung des Systems durchgeführt.

Neben den zuvor genannten Ansätzen zur vereinfachten Programmierung von Robotern wird im Bereich der Leichtbauroboter die Programmierung durch Demonstration oder Teach-In betrachtet. ORENDT ET AL. (2016) präsentieren in diesem Feld einen Ansatz für das Einteachen eines Roboters ohne Expertenkenntnis. Darauf aufbauend betrachten RIEDL ET AL. (2016) die Programmierung mehrerer Roboter durch Führen des Roboters. Dabei können die Roboter sequentiell programmiert werden, parallel, wobei mehrere Roboter gleichzeitig geführt werden müssen, oder durch Führen eines Roboters, während das Programm eines weiteren Roboters abläuft. Das aufgenommene Programm wird in Form eines Geschwindigkeitsdiagramms in einem Gantt-Chart auf einer Benutzerschnittstelle angezeigt, wo es editiert werden kann.

Die beschriebenen Ansätze befassen sich jeweils mit einer Art der vereinfachten Programmierung oder der Konfiguration von Robotersystemen. Dabei kommen aufgabenorientierte Verfahren oder die Programmierung durch Handführen zum Tragen. In allen Arbeiten ist der Zielzustand nach der Programmierung ein automatisiertes System. Eine Betrachtung der Mensch-Roboter-Kooperation erfolgt in den Arbeiten nicht, sodass auch keine Schnittstellen zwischen dem automatisierten System und einem Menschen beschrieben werden. Die Integration von Schnittstellen zum Menschen in das Roboterprogramm bleibt in den Arbeiten ungeklärt und muss daher untersucht werden.

#### **3.1.2 Kommerzielle Programmiersysteme**

Neben den Ansätzen in der Wissenschaft werden bereits kommerzielle Programmiersysteme zur vereinfachten und herstellerunabhängigen Programmierung von Robotern vertrieben.

Das Unternehmen Artiminds bietet eine Programmierumgebung, bei der die Offline-Programmierung mit einer Online-Programmierung verknüpft ist. Über eine Benutzeroberfläche können Befehle in ein Programm integriert und parametrisiert werden. Nach Abschluss der Generierung wird das Programm auf die Steuerung des Roboters übertragen, wo es dann abgerufen wird. Das Roboterprogramm wird zwischen Generierung und Übertragung auf die Steuerung simuliert, wobei es auf Kollisionen überprüft wird (ARTIMINDS ROBOTICS GMBH 2019).

Aus Forschungsaktivitäten am Fraunhofer IPA heraus entstanden ist die Software Drag & Bot, die ähnlich wie die Software von Artiminds arbeitet. Ziel dieser Programmierumgebung ist ebenfalls die vereinfachte, herstellerunabhängige Programmierung, die auf unterschiedliche Anwendungen übertragen werden kann. Die Software bietet eine Benutzeroberfläche, auf der Roboterbefehle über Drag & Drop zu einem Programm zusammengeführt und konfiguriert werden können. Das Programm wird anschließend generiert, wird aber nicht auf die Robotersteuerung übertragen, sondern über das Robot Operating System (ROS) abgerufen (DRAG AND BOT GMBH 2019).

Der Leichtbauroboter Panda der Firma Franka Emika wird mit einer grafischen Benutzerschnittstelle mit Apps vertrieben. Dabei werden die Apps, die jeweils Programmbausteine repräsentieren, auf einem Webinterface zu einem Programm zusammengesetzt und jeweils parametrisiert. Die nutzende Person wird bei der Parametrisierung durch die einzelnen Parameter geführt. Zur Zusammenarbeit mit dem Menschen kann ein Baustein *Interaktion* eingefügt werden. Dieser bewirkt, dass der Roboter auf ein Signal des Menschen, beispielsweise eine Berührung des Roboters, wartet, bevor er seine Arbeit fortsetzt (FRANKA EMIKA GMBH 2018). In diesem Programmiersystem werden jedoch nicht die Aufgaben des Menschen dargestellt oder die Zuteilung von Aufgaben unterstützt.

Im Bereich der Mensch-Roboter-Kooperation finden vor allem die Teach-In und Playback-programmierung Anwendung. Dies hängt insbesondere damit zusammen, dass die meisten Leichtbauroboter diese Funktion zur Verfügung stellen. Damit können auch Arbeitskräfte mit geringen Programmierkenntnissen diese Programmierung vornehmen. In den genannten Programmiersystemen sind jeweils nicht die Aufgaben des Menschen angezeigt, aus denen Schnittstellen des Menschen zum Roboterprogramm generiert werden können.

Die aufgezeigten kommerziellen Programmiersysteme können zwar eine Vereinfachung der Programmierung vorweisen, jedoch werden die Zusammenhänge zwischen den beiden Partnern Mensch und Roboter nur unzureichend betrachtet. In keinem der vorgestellten Programmiersysteme werden die Aufgaben beider Ressourcen und der Gesamttablauf dargestellt und verwendet. Außerdem erfolgt kaum eine Betrachtung der Zusammenarbeit zwischen Mensch und Roboter im Betrieb. Die einzige Lösung, die vorgestellt wird, ist die Signalübergabe an den Roboter durch eine Berührung des Roboters.



## **3.2 Planung von Applikationen der Mensch-Roboter-Kooperation**

Ein Schwerpunkt der Arbeiten zum Thema der Mensch-Roboter-Kooperation liegt in der Planung. Zunächst werden allgemeine Ansätze und Projekte zur Planung von Applikationen beschrieben. Zur Programmierung von Robotersystemen und zur Planung einer Mensch-Roboter-Kooperation ist eine definierte Form zur Beschreibung von Aufgaben erforderlich. Diese Aufgaben werden nachfolgend mit den Fähigkeiten der Ressourcen abgeglichen und dann den Ressourcen zugeteilt. Anschließend werden jeweils Ansätze zu den genannten Bereichen dargelegt.

### **3.2.1 Allgemeine Ansätze und Projekte zur Planung**

In den frühen Phasen der Betrachtung der MRK untersuchte THIEMERMANN (2005) die Aufgabenaufteilung in der Zusammenarbeit zwischen Mensch und einem SCARA-Roboter. Die betrachteten Kriterien für die Aufteilung umfassten dabei Sicherheit, Produktivität und Usability-Variablen (THIEMERMANN 2005). Ebenfalls bereits zu einem frühen Zeitpunkt der Entwicklung der MRK wurde das Thema einer geeigneten Aufgabenaufteilung in der Mensch-Roboter-Kooperation durch BEUMELBURG (2005) behandelt. Dieses Vorgehen wird in Abschnitt 3.2.4 näher beschrieben.

Im Projekt *Manuserv* wurde die Analyse von Montageaufgaben zur Identifikation von Potenzialen für eine kooperative Durchführung mit einem Leichtbauroboter behandelt. Zunächst werden die manuellen Montageaufgaben aufgenommen, analysiert und daraufhin simuliert. Basierend auf den durch den potenziellen Nutzer eingegebenen Daten zum Prozess, wird ein Roboter empfohlen und ein Vorschlag für die Zuteilung der Aufgaben erarbeitet. Anschließend wird dem Nutzer das Szenario in einer 3D-Umgebung dargestellt (DEUSE ET AL. 2014). Dieser Ansatz wurde von HEINZE ET AL. (2016) und WANTIA ET AL. (2016) weiterentwickelt. Dabei wurden die Prozesse auf ihre technische Umsetzung hin überprüft, die Prozesse den Ressourcen zugeordnet und eine Simulation mit Hilfe der Simulationsumgebung VEROSIM durchgeführt.

Ziel des internationalen Projekts *ROBO-Partner* ist die Planung von Mensch-Roboter-Kooperations-Szenarien. Dabei wird eine Architektur zur flexibleren Integration und Kommunikation bereitgestellt. Der Anwendungsfall dieses Projekts liegt in der Automobilindustrie (MICHALOS ET AL. 2014). Im Rahmen des Projekts stellen TSAROUCHI ET AL. (2017) eine Methode zur Zuteilung der Aufgaben zu den

Ressourcen vor. Dabei gibt der Nutzer die Liste der Aufgaben und die zur Verfügung stehenden Ressourcen an. Die Ressourcen werden dann basierend auf Kriterien wie der Reichweite, der Traglast, der Fähigkeit zum Handhaben von Bauteilen und anderen bewertet. Die zahlenmäßige Bewertung erfolgt anhand von durch den Nutzer vorgegebenen Zielen, wie dem geringsten Arbeitsraum, der Verbesserung der Ergonomie oder der Fähigkeit des Roboters zum Ausführen einer Aufgabe.

Das Projekt *KomPi* beschäftigt sich mit einer verrichtungsbasierten, digitalen Planung kollaborativer Montagesysteme und der Integration in variable Produktionsszenarien (KOMPI 2019). Ein Fokus in diesem Projekt liegt auf der Simulation von MRK-Applikationen (LEMMERZ ET AL. 2018). Außerdem werden über eine Potenzialanalyse, basierend auf den Aufgaben, MRK-Potenziale identifiziert (LINSINGER ET AL. 2018).

Im Projekt *KoKoMo*, geht es um die Konsolidierung von Produkt- und Prozessdaten aus kollaborativen Montagesystemen und deren Rückführung und Integration in ein Planungssystem zur Vereinfachung des Einsatzes einer solchen Applikation (SEITER 2017). Um einen Roboter flexibel für die jeweilige Aufgabe einsetzen zu können, entwickeln RÜCKERT ET AL. (2018) einen Ansatz zur Kalibrierung eines Leichtbauroboters mit Hilfe einer Kamera.

Im Rahmen des Projekts *SafeMate*, dessen Fokus ebenfalls auf dem Einsatz der Mensch-Roboter-Kollaboration liegt, stellen BLANKEMEYER ET AL. (2018) ein Vorgehen zur Identifikation potenzieller Anwendungsfälle für die MRK und darauf aufbauend das Vorgehen zur Implementierung einer Applikation vor. Der zweistufige Potenzialcheck soll mit wenig Aufwand durchführbar sein. Die erste Stufe besteht dabei aus einem Fragebogen, in dem Fragen zu Befähigern und Hindernissen zu einem bestimmten Anwendungsfall beantwortet werden müssen. Nach einer positiven Bewertung der ersten Stufe wird die mögliche Anwendung genauer untersucht. Die Durchführung des Potenzialchecks und der detaillierten Untersuchung erfolgen manuell (BLANKEMEYER ET AL. 2018).

#### **3.2.2 Beschreibung von Aufgaben**

ARAI ET AL. (2000) beschreiben ein computergestütztes Robotersystem. Dabei unterscheiden sie zwischen einem Management- und einem Ausführungsbereich. Der Managementbereich besteht aus drei Ebenen: Aufgabenebene, Prozessebene und Operationsebene. Die Aufgabenebene beschreibt den gesamten Montageprozess, der in der Prozessebene weiter untergliedert wird. Montageprozesse aus der

Prozessebene werden in der Operationsebene in Einzelbewegungen der Geräte unterteilt. Der Ausführungsbereich ist zuständig für die Steuerung der technischen Geräte.

BEUMELBURG (2005), die in ihrer Arbeit eine fähigkeitsbasierte Montageablaufplanung für die Mensch-Roboter-Kooperation entwickelt, verwendet als Basis einen Montagevorranggraphen, in dem einzelne Montageschritte weiter unterteilt sind in einen Handhabungs- und einen Fügevorgang. Das Handhaben und das Verwenden von Fügehilfsmitteln werden dabei jeweils als separater Schritt angesehen.

LOTTER & WIENDAHL (2006) nutzen die Beschreibung der Aufgaben für eine Primär- und Sekundäranalyse. Im Bereich der Kleingerätemontage erfolgt die Beschreibung auf der Ebene eines Montageschrittes und wird durch die fünf Elementarbewegungen, basierend auf MTM, beschrieben. Bei der Montage von Großgeräten wird aufgrund der meist größeren Bewegungen des Menschen eine Unterteilung angelehnt an MTM-UAS betrachtet. Die Autoren beschreiben dabei folgende Grundvorgänge: Aufnehmen und Platzieren bzw. nur Platzieren, visuelle Kontrolle, Körperbewegungen, Hilfsmittel handhaben, Bewegungszyklen und gegebenenfalls Stellteile betätigen. Diese Grundvorgänge können teilweise weiter in die Elementarvorgänge untergliedert werden.

DEUSE ET AL. (2014) wählen eine standardisierte Prozessbeschreibung basierend auf MTM. Ein Prozess besteht dabei aus einer Sequenz von Elementarbewegungen. Die Beschreibung der Elementarbewegungen wird um Kriterien zur Bewertung der Automatisierbarkeit angereichert. Über die Bewertung der Automatisierbarkeit der einzelnen Elementarbewegungen kann identifiziert werden, ob ein Prozess automatisierbar ist.

In einem Ansatz zur ROS-basierten Koordination einer Montageaufgabe in einer Mensch-Roboter-Kooperation unterteilen TSAROUCHI ET AL. (2015) und TSAROUCHI ET AL. (2016b) die Montageaufgabe in Auftrag, Aufgaben und Vorgänge. Aufgaben können dabei von Mensch, von Roboter oder von beiden Ressourcen ausgeführt werden. Auf der Ebene der Vorgänge werden für den Menschen keine weiteren Parameter beschrieben, wohingegen diese Beschreibung bei einem Roboter notwendig ist.

MÜLLER ET AL. (2016) erarbeiten in ihrem Ansatz ebenfalls eine Methode, um die Aufgaben prozessorientiert zu Mensch und Roboter zuzuordnen. Die Aufgaben werden hierbei in Basis- und spezifische Aufgaben unterteilt. Zu den Basisaufgaben

werden Handhaben, Fügen, Kommissionieren, Hilfsprozesse und Spezialprozesse gezählt. Spezifische Aufgaben sind beispielsweise Zuführen, Transportieren, Säubern oder Nacharbeit. Durch Parameter, wie z. B. minimaler Druck oder Position für den Fall des Einpressens, werden die Aufgaben weiter beschrieben. Auf Basis der Aufgaben werden diese den Ressourcen zugeordnet.

RANZ ET AL. (2017) definieren für ihre Arbeit zur fähigkeitsbasierten Zuteilung von Aufgaben in einer Mensch-Roboter-Kooperation eine Aufgabenebene, die einen Bewegungsablauf abbildet. Ein Beispiel für die Ebene der Beschreibung der Aufgabe ist *Wasserpumpe aufnehmen und platzieren*. In einer tieferen Ebene werden diese Aufgaben weiter heruntergebrochen.

MICHNIEWICZ ET AL. (2016) beschreiben in ihrer Arbeit zur automatischen Programmierung, Konfiguration und Optimierung von Roboterzellen die Aufgaben mit Hilfe von Funktionsprimitiva. Die Aufgaben werden automatisiert von einer virtuellen Repräsentation des Produkts in Form von CAD-Modellen abgeleitet. Die am häufigsten beschriebene Aufgabe ist dabei *Teile montieren*, die weiter in Funktionsprimitiva, wie Pose detektieren, bewegen oder loslassen, heruntergebrochen wird.

### 3.2.3 Beschreibung der Fähigkeiten der Ressourcen

HAMMERSTINGL & REINHART (2017) beschreiben in ihrer Arbeit ein Modell zur Definition von Fähigkeiten, die die produkt- und prozessseitigen Anforderungen mit den Betriebsmitteln verbinden. Diese Fähigkeiten werden unterteilt in elementare und zusammengesetzte Fähigkeiten, die eine Kombination mehrerer elementarer Fähigkeiten darstellen (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**). Des Weiteren werden Prozessschritte als elementare Operation definiert, die nicht weiter zerlegt werden kann. Eine Aufgabe wird beschrieben als eine Sequenz von Prozessschritten. Für den Abgleich der Aufgabe mit den Gerätefähigkeiten korrelieren die Prozessschritte semantisch mit den Gerätefähigkeiten. Standardisierte elementare Fähigkeiten in der Montage werden unterteilt in Handhaben, Verbinden, Kontrollieren, Informationsaustausch, Mensch-Maschine-Interaktion und Sonderfunktionen, die jeweils weiter untergliedert werden. Die Fähigkeit *Verbinden* beispielsweise kann weiter verfeinert werden in Form-, Kraft- und Stoffschluss. Unterhalb dieser Ebene finden sich für den Stoffschluss Vorgänge wie das Lötens, was weiter zergliedert wird in verschiedene Arten des Lötens. Für die Fähigkeiten werden Parameter näher definiert und anschließend miteinander verglichen.

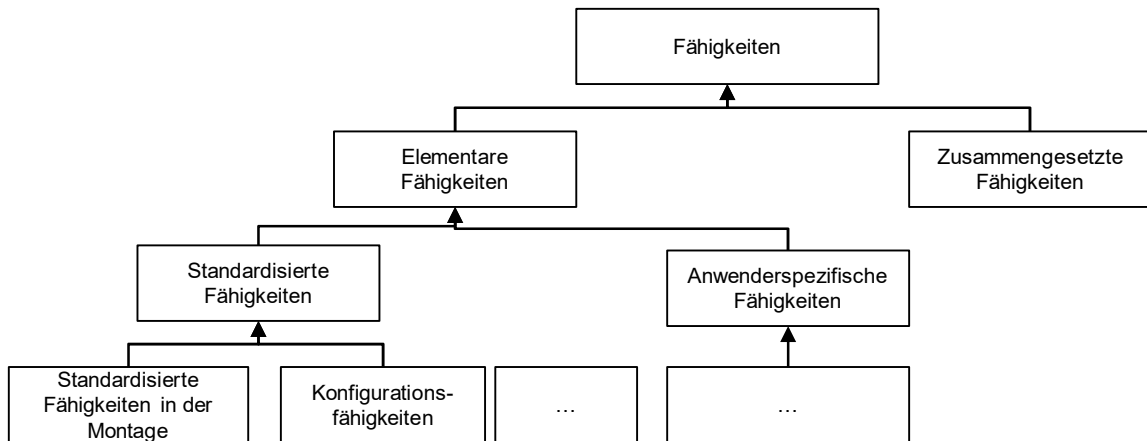


Abbildung 9: Hierarchie der elementaren Fähigkeiten nach HAMMERSTINGL & REINHART (2017)

Die Parameter der elementaren Fähigkeiten werden in produkt- oder prozessbezogene unterschieden. Die prozessbezogenen Parameter werden weiter in Prozessgrößen und eine Geometriebeschreibung untergliedert. Bei den Prozessgrößen gelten als Input die Anforderungen des Anwenders bzw. des Prozesses und als Output das Ergebnis der Fähigkeit. Die prozessbezogenen Parameter werden über die in Abbildung 10 dargestellten Eigenschaften beschrieben (HAMMERSTINGL & REINHART 2017):

Bezeichner	Datentyp	Wert oder Wertebereich	Einheit	Notwendigkeit (option./ verpfl.)	Input / Output
------------	----------	------------------------	---------	----------------------------------	----------------

Abbildung 10: Eigenschaften der prozessbezogenen Parameter nach HAMMERSTINGL & REINHART (2017)

Auf Seiten der Anforderungen wird ein Wertebereich für den Parameter angegeben, wohingegen auf Seite der Gerätefähigkeit ein Einzelwert spezifiziert wird, der als Basis für den Abgleich gilt. Bei den produktbezogenen Parametern handelt es sich beispielsweise um Kennwerte wie Masse, Abmessungen sowie Materialkennwerte, die zusätzliche Informationen für den Abgleich mit Gerätefähigkeiten liefern (HAMMERSTINGL & REINHART 2017).

In der Arbeit von BACKHAUS (2016) beschreiben Skills herstellerunabhängig die von einer Ressource über Befehle, Funktionen, Funktionsbausteine oder Schnittstellen zur Verfügung gestellten Funktionalitäten. Die Skills enthalten Eigenschaften zu den Ressourcen, mit denen die Einsetzbarkeit für einen Montageprozess überprüft werden kann. Sie werden darüber hinaus für die Programmierung eingesetzt und inkludieren daher auch Informationen über Befehle und Kommunikationsschnittstellen. Die Beschreibung der Skills basiert auf der Arbeit von SELIG (2011). BACKHAUS (2016)

beschreibt in seiner Arbeit die Herleitung direkter Skills, die den Produktionsprozess unmittelbar unterstützen sowie deren Struktur.

MÜLLER ET AL. (2016) beschreiben die Fähigkeiten von Ressourcen. Dafür werden die in MÜLLER ET AL. (2015) betrachteten Fähigkeiten für den Fall der Mensch-Roboter-Kooperation um zusätzliche Kriterien nach SPILLNER (2015) erweitert. Eine Fähigkeit wird über unterschiedliche Kriterien, wie beispielsweise der minimale Druck oder die Position im Falle einer Fügeaufgabe, beschrieben. Basierend auf den Aufgaben und den Fähigkeiten, kann im nachfolgenden Schritt der Abgleich der Fähigkeiten durchgeführt werden.

#### **3.2.4 Abgleich der Anforderungen und Fähigkeiten zur Aufgabenzuteilung**

Zur Zuteilung der Aufgaben auf die verschiedenen Geräte führen MICHNIEWICZ ET AL. (2016) einen Abgleich der Anforderungen an die Aufgabe mit den Fähigkeiten der Geräte, die über die Funktionsprimitive beschrieben sind, durch. Dabei wird überprüft, welches Cyber-Physische-Gerät das geforderte Funktionsprimitiv zur Verfügung stellen kann.

Die Basis für die Fähigkeitsbewertung nach BEUMELBURG (2005) sind die drei Unternehmensziele Zeit, Kosten und Qualität, die, abgebildet auf den Fall der Montage, in die Technologiemerkmale Taktzeit, Zusatzinvest, Prozesssicherheit und Arbeitsqualität übertragen werden. Zur Beurteilung der Fähigkeiten der Ressourcen wird evaluiert, wie sich die Ausprägungen im Hinblick auf diese Technologiemerkmale verhalten. Dazu wird eine Ordinalskala verwendet. Im weiteren Prozess der Bewertung wird die Berechnung eines Eignungsgrads für Mensch und Roboter durchgeführt. Dieser Schritt ist in Abbildung 11 dargestellt.

Die Kriterien werden in die vier Bereiche Montageprozess, Ergonomie, Bauteil und Teilebereitstellung unterteilt. Die Tabelle der Kriterien kann BEUMELBURG (2005) entnommen werden. Die Zuteilung der Aufgaben zu den Ressourcen basierend auf der Bewertung der Fähigkeiten erfolgt mit Hilfe eines genetischen Algorithmus.

HAMMERSTINGL & REINHART (2017) führen zum Abgleich der Fähigkeiten mit den Anforderungen einen automatisierten Abgleich durch. Dabei werden für jeden Prozessschritt Eignungsprüfungen durchgeführt und passende Betriebsmittel ausgewählt.

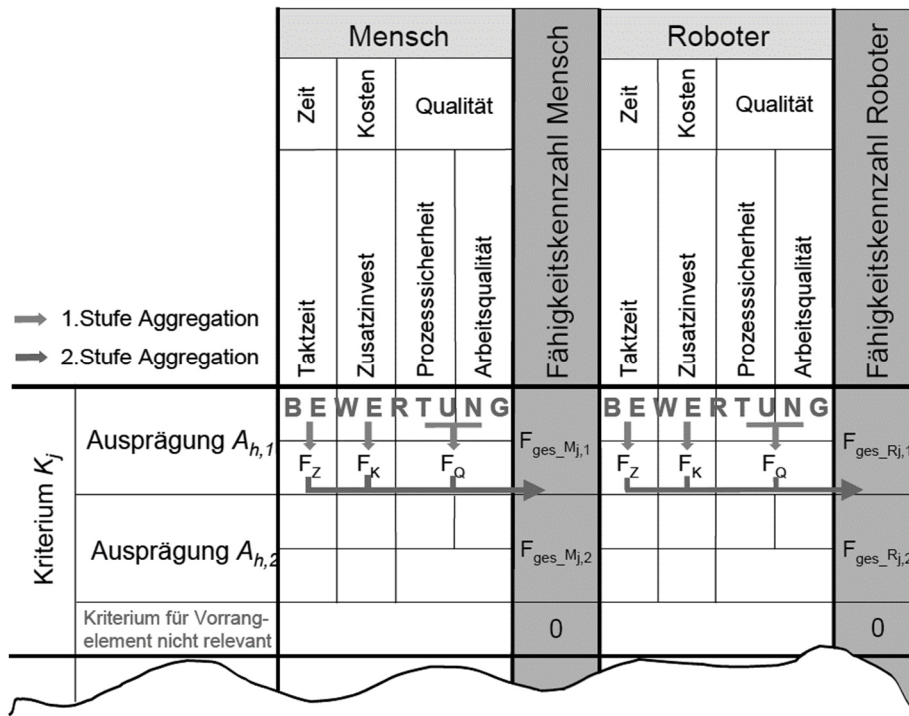


Abbildung 11: Berechnung des Eignungsgrads (BEUMELBURG 2005)

Bei TSAROUCHI ET AL. (2016b) erfolgt zunächst eine Bewertung der Aufgaben bzgl. der Durchführbarkeit durch eine Ressource und anschließend die Generierung mehrerer Möglichkeiten zum Ablauf. Zur Bewertung der Aufgaben werden vor allem die Kriterien Traglast, Fähigkeit zum Handhaben flexibler Gegenstände und Anzahl der benötigten Arme verwendet. Mit diesem Ergebnis werden alternative Abläufe generiert, die bzgl. mehrerer Kriterien bewertet werden, wobei ein Gesamtwert berechnet wird. Die Alternative mit dem höchsten Wert wird ausgewählt.

Zum Abgleich der Anforderungen an die Aufgabe mit den Fähigkeiten der Ressourcen Roboter und Mensch führen MÜLLER ET AL. (2016) einen paarweisen Vergleich zur Bewertung der Fähigkeiten durch. Während die meisten Fähigkeiten, die nicht den Menschen betreffen, alphanumerisch und damit mit Software bewertet werden können, muss der paarweise Vergleich für den Menschen manuell erfolgen. Neben dem paarweisen Vergleich wird eine Analyse des Layouts durchgeführt, wobei die Anordnung der Maschinen und sicherheitsrelevante Aspekte betrachtet werden. Bei Auftritt potenzieller Gefährdungen kann eine Aufgabenübergabe zwischen den Ressourcen als erforderlich angesehen werden.

In den Ansätzen von DEUSE ET AL. (2014) und WANTIA ET AL. (2016) wird überprüft, ob die über MTM beschriebenen Aufgaben von einem Robotersystem übernommen

werden können. Bei einem Greifvorgang werden beispielsweise die Greifkraft und das Gewicht des Bauteils mit den Fähigkeiten des Greifers verglichen. Die Planungsaufgabe wird mit Hilfe der Planning Domain Definition Language (PDDL) beschrieben und an einen Planer gegeben. Die Kosten werden für die jeweiligen Aufgaben beschrieben, wobei diese für die Durchführung durch einen Menschen höher liegen als bei Durchführung durch einen Roboter. Daher werden die Aufgaben priorisiert dem Roboter zugeordnet.

Die Arbeit von RANZ ET AL. (2017) baut auf der Arbeit von BEUMELBURG (2005) auf. Ihr Ansatz zur Zuteilung der Aufgaben unterteilt sich in drei Schritte: Ableitung des Montageplans, Zuteilung der eindeutigen Aufgaben und Zuteilung der varianten Aufgaben. Für den Schritt der Definition von Aufgaben, die entweder nur durch den Menschen oder nur durch den Roboter erledigt werden können, wurden, bezogen auf den Menschen, zwölf montagerelevante menschliche Fähigkeiten und ergonomische Grenzwerte identifiziert. Der Schritt zur Zuteilung der varianten Aufgaben wurde an das Vorgehen von BEUMELBURG (2005) angeknüpft und die Fähigkeiten bezogen auf die Unternehmensziele definiert. Dabei nutzen RANZ ET AL. (2017) die Ziele Prozesszeit, zusätzlicher Invest und Prozessqualität. Hierzu wurden 25 Kriterien identifiziert, wobei für jedes Kriterium ein Fähigkeitsindikator für Mensch oder Roboter berechnet wird, für den eine besser-gleich- oder schlechter-Bewertung getroffen wird. Die Fähigkeitsindikatoren werden anschließend in einer Matrix festgehalten. Darauf basierend werden die Aufgaben zugeteilt (RANZ ET AL. 2017).

#### **3.2.5 Simulation für die Mensch-Roboter-Kooperation**

Im Bereich der Robotik ist die Simulation insbesondere für die Layoutplanung und Generierung von fehlerfreien Roboterprogrammen relevant (MÜLLER ET AL. 2019). So nutzen auch BACKHAUS (2016) und EHRMANN (2007) in ihren Arbeiten zur aufgabenorientierten Programmierung eine Simulation, um die generierten Programme vor der Inbetriebnahme zu prüfen. Auch beim kommerziellen Programm der Firma Artiminds (ARTIMINDS ROBOTICS GMBH 2019) wird das Roboterprogramm nach der Generierung durch eine Simulation überprüft.

Über Robotersimulationssysteme können die Bewegungen von Robotersystemen dargestellt werden (HAUN 2013). Die Simulation von Robotern zielt nach HAUN (2013) auf die vier Ziele Virtualisierung, Totalisierung, Formalisierung und Antizipation ab, wobei die Virtualisierung die wichtigste Form darstellt. Dabei kann eine Situation durchgespielt werden und Alternativen können nebeneinander betrachtet werden (HAUN 2013). Viele Roboterhersteller bieten dazu eigene



Simulationssoftware an, wie z. B. KUKA.Sim (KUKA AG 2019) oder das RobotStudio von ABB (ABB 2018).

Des Weiteren gibt es auch Simulationsprogramme, die nicht herstelleregebunden sind und mit denen daher unterschiedliche Roboter simuliert werden können. Dazu gehören v-rep (COPPELIA ROBOTICS 2019) oder Visual Components (VISUAL COMPONENTS GMBH 2019). In vielen Fällen hängt die Simulation des Roboters auch zugleich mit der Programmgenerierung zusammen. So führt die Firma Artiminds eine Simulation des erstellten Roboterprogramms durch, bevor es auf die Robotersteuerung geladen wird (ARTIMINDS ROBOTICS GMBH 2019). Ähnlich dazu bietet die Firma RoboDk eine Toolbox zur Simulation und Offline-Programmgenerierung von Industrierobotern an (ROBODK 2018).

Eine weitere Möglichkeit zur Simulation von Roboterbewegungen bietet das *Robot Operating System (ROS)*. ROS beschreibt eine Kommunikationsstruktur, die über dem Betriebssystem des Roboters liegt. Durch das Plugin rviz können in ROS dargestellte Szenen, aber auch Bilder oder Punktwolken visualisiert werden (QUIGLEY ET AL. 2009).

Zur Simulation eines Ablaufs der Mensch-Roboter-Kooperation muss neben dem Roboter der Mensch simuliert werden. Dazu stehen digitale Menschmodelle zur Verfügung, wobei es sich nach MÜHLSTEDT (2016) um Softwaresysteme zur Abbildung, Simulation oder Nutzung von geometrischen und weiteren Eigenschaften und Fähigkeiten von Menschen handelt. BUBB ET AL. (2006) beschreiben diese Modelle als Nachbildung physischer und psychischer Eigenschaften des Menschen zur Auslegung und Konstruktion jeglicher Mensch-Maschine-Systeme. Nach MÜHLSTEDT (2016) können die Menschmodelle in drei Arten eingeteilt werden:

- Arbeitswissenschaftliche digitale Menschmodelle
- Dynamische / biomechanische / grafische Modelle, FE- /VR-Modelle
- Avatare / Agenten, 3D-Animationen, Computerspiele, Animationsfilme

Dabei sind mit dem Begriff digitale Menschmodelle meist die arbeitswissenschaftlichen Menschmodelle gemeint, die für diese Arbeit relevant sind und daher nachfolgend näher beschrieben werden. Mit Hilfe dieser Modelle, die meist Teil eines CAx-Systems sind, können simulative Analysen durchgeführt werden und damit Szenarien überprüft werden (MÜHLSTEDT 2016). In der Praxis verbreitet sind vor allem drei digitale Menschmodelle: RAMSIS von Human Solutions (SEIDL 1994), Human Builder von Dassault Systemes und Jack der Firma Siemens PLM

(SIEMENS INDUSTRY SOFTWARE 2020; MÜHLSTEDT 2016; MÜHLSTEDT & SPANNER-ULMER 2010).

Zur gemeinsamen Simulation von Mensch und Roboter existieren bisher kaum Ansätze. Daher bauen LEMMERZ ET AL. (2018) ein System auf, um eine Robotersimulation in ein Simulationssystem für menschliche Abläufe zu integrieren. Dazu nutzen sie als Robotersimulationssoftware ROS und für die Simulation der menschlichen Abläufe das Tool *Editor for Manual Work Activities* (EMA). Zwischen ROS und EMA bauen sie eine Schnittstelle auf, wobei die Daten des Menschmodells an ROS übergeben werden, so dass die Roboterbahn um das Menschmodell herum geplant wird. Die Ergebnisse werden dann zurück an EMA gegeben, um sie zu visualisieren (LEMMERZ ET AL. 2018).

Den Stand der Technik im Bereich der Planung zusammengefasst, kann festgestellt werden, dass einige Arbeiten sich mit diesem Thema beschäftigen. Zur Beschreibung der Aufgaben existieren verschiedene Ansätze, die unterschiedliche Beschreibungsebenen definieren. Zur Verwendung im Programmiersystem ist eine einheitliche Beschreibung notwendig, die an verschiedenen Stellen im System eingesetzt werden kann. Daher ist zu prüfen, welche Ansätze dies bieten. Zum Abgleich der Fähigkeiten und der Zuteilung der Aufgaben werden ebenfalls differente Ansätze vorgestellt. Die Ansätze beinhalten den Abgleich in manueller oder automatisierter Form, z. B. durch einen genetischen Algorithmus. Hierbei ist zu beurteilen, welcher dieser Ansätze erweiterbar für den betrachteten Einsatz ist. Zur Simulation einer Applikation der Mensch-Roboter-Kooperation gibt es bisher nur wenige Ansätze, obwohl sowohl zur Simulation von Robotersystemen als auch des Menschen über digitale Menschmodelle entsprechende Simulationsumgebungen vorhanden sind. Im Rahmen der vorliegenden Arbeit ist daher eine Simulationsumgebung zu identifizieren, die für die Simulation beider Ressourcen und die Integration in das Programmiersystem geeignet ist.

### **3.3 Betrieb in der Mensch-Roboter-Kooperation**

Nach der Planung und Programmierung erfolgt die Betriebsphase, in der die Montageaufgabe durch Mensch und Roboter in einem gemeinsamen Arbeitsraum gelöst wird. Hierbei ist die Sicherheit einer der Schlüsselfaktoren für eine MRK (MAURTUA ET AL. 2017; LASOTA ET AL. 2014). Eine weitere Fragestellung ist die Adaption bzw. der Autonomiegrad des Roboters in der Zusammenarbeit (KOPPENBORG ET AL. 2013; HOFFMAN & BREAZEAL 2007). Nachfolgend werden

zunächst die Strategien zur Vermeidung von Kollisionen betrachtet. Anschließend werden Erkenntnisse aus arbeitswissenschaftlichen Studien zur Zusammenarbeit zwischen Mensch und Roboter vorgestellt.

#### 3.3.1 Sicherheitstechnologien zur Kollisionsvermeidung

Wie in Abschnitt 2.4.4 erläutert, wird die Sicherheit von Applikationen durch die Einhaltung der Normen und Standards gewährleistet. ROBLA-GOMEZ ET AL. (2017) fassen in ihrem Paper Sicherheitsmaßnahmen im Rahmen der Mensch-Roboter-Kollaboration zusammen. Sie ordnen dabei unterschiedliche Maßnahmen in die Ziele Minimierung von Verletzungen und die Vermeidung von Kollisionen ein. Nachfolgend werden ausgewählte Ansätze zur Erhöhung der Sicherheit in der Mensch-Roboter-Kooperation beschrieben.

Durch die Entwicklung mit Sensorik ausgestatteter Roboter, die teilweise über Kraft-Momenten-Sensoren verfügen, sollen Verletzungsgefahren für den Menschen vermindert werden. Über seine Sensorik detektiert der Roboter Kollisionen, die unmittelbar zum Anhalten des Roboters führen (LUCA ET AL. 2006). Um nicht erst bei Eintritt einer Kollision reagieren zu können, wurde zusammen mit der KUKA AG das Konzept einer Sensorhaut entwickelt (ROBERT BOSCH MANUFACTURING SOLUTIONS GMBH 2018). In dieser ist neben kapazitiver Sensorik weiches Material enthalten, das zusätzlich für den Abbau von Energie bei einer Kollision sorgt. Bei den kapazitiven Sensoren wird eine Annäherung durch die Änderung der Kapazität detektiert (HOFFMANN ET AL. 2016). Neben dieser Technologie bietet die Firma Blue Danube Robotics eine Haut mit Luft an. Dabei wird kontinuierlich der Druck gemessen, wodurch Druckveränderungen im Sensorsystem festgestellt werden können. Da diese auf eine Kollision hinweisen, meldet das System dies entsprechend an die Robotersteuerung (JUSCHKAT 2018). Ebenfalls zur Vermeidung von Kollisionen werden kamerabasierte Ansätze betrachtet. Durch den Abgleich einer 3D-Simulation des Roboters mit den Echtzeit-Bildern des Werkers im Arbeitsraum kann eine kritische Situation detektiert und der Roboter entsprechend angehalten werden (SCHMIDT & WANG 2013).

MEZIANE ET AL. (2017) stellen in ihrer Arbeit das System *SPADER* vor, das über eine Online-Trajektorienplanung die Vermeidung von Kollisionen zum Ziel hat. Dabei wird der Werker über ein Kamerasystem detektiert und die Wegpunkte der dynamischen Objekte werden mit Hilfe eines neuronalen Netzes berechnet. Darauf basierend werden die Trajektorien so geplant, dass Kollisionen vermieden werden. Der Algorithmus dazu wird in einer Simulation überprüft.

DRÖDER ET AL. (2018) stellen einen Ansatz zur Online-Bahnplanung von Robotern für die Mensch-Roboter-Kooperation dar, um Kollisionen mit Menschen oder Hindernissen zu vermeiden. Dazu verwenden sie für die Bahnplanung einen Ansatz mit engsten Nachbarn und künstliche neuronale Netze zur Objektdetektion. Zur kamerabasierten Detektion wird eine Kinect verwendet.

Im Bereich der Planung und Gestaltung von sicheren Arbeitsplätzen stellen MICHALOS ET AL. (2015) ein Vorgehen vor, um einen Arbeitsplatz sicher für die Mensch-Roboter-Kollaboration zu gestalten. Sie betrachten dabei Technologien wie Sensoren zur Überwachung des Arbeitsraums, Sicherheitsfunktionen in der Robotersteuerung, aber auch Technologien wie Augmented Reality, um den Menschen besser in den gemeinsamen Betrieb zu integrieren.

Bei Betrachtung der vorgestellten Arbeiten ist festzuhalten, dass die gezeigten Ansätze zur kamerabasierten Überwachung von Kollisionen keine nach Normen und Richtlinien sichere Überwachung liefern.

#### **3.3.2 Ansätze zur kollaborierenden Arbeit**

Für den Betrieb einer Mensch-Roboter-Kooperation gibt es unterschiedliche Ansätze, wobei die Arbeiten in die Bereiche der Antizipation des Roboters (FABER ET AL. 2015; LENZ ET AL. 2008), der Intentionserkennung (AWAIS & HENRICH 2010) und der Bahnplanung (PELLEGRINELLI ET AL. 2017; MEZIANE ET AL. 2017) eingeteilt werden können.

LENZ ET AL. (2008) haben ein System erarbeitet, bei dem auf Basis einer Wissensdatenbank und von Entscheidungsprozessen eine Antizipation des Roboters ermöglicht werden soll. Dazu wird eine Wissensdatenbasis mit Informationen über den Montageplan die Werkzeuge und die Teile erstellt. Abhängig von der durchgeführten Tätigkeit der Werkkraft und den nächsten möglichen Schritten laut Montageplan, wird der nächste Schritt des Roboters generiert. Der Arbeitsbereich wird hierbei mit vier Kameras detektiert. Dabei befindet sich eine Kamera nahe des Tool Center Points, um Objekte im Arbeitsbereich zu detektieren. Die anderen drei Kameras dienen der Detektion der arbeitenden Person, insbesondere ihrer Hände.

Ebenfalls in dem Bereich der Antizipation angesiedelt sind die Arbeiten von SCHLICK ET AL. (2014) und FABER ET AL. (2015). Über ein kognitives Modell wird der Mensch im System repräsentiert. Ziel ist es, die nächsten Tätigkeiten des Roboters zu definieren. Dazu wird Prozesswissen in Form von Wenn-Dann-Regeln hinterlegt. Mit Hilfe des CAD-Modells des finalen Produkts werden die

Montagesequenzen über den assembly-by-disassembly-Ansatz ermittelt. In jedem Schritt werden alle gültigen Montagesequenzen für die restlichen Aufgaben berechnet, wobei ein Zustandsgraph erstellt wird. Die Kanten des Zustandsgraphs werden abhängig vom Aufwand für den Montageschritt mit entsprechenden Kosten versehen. Ausgewählt wird der Pfad mit den geringsten Kosten.

Im Bereich der Bahnplanung stellen THOMAS ET AL. (2016) einen Ansatz zur Adaption des Roboters an den Menschen vor. Dabei wird der Mensch in einem virtuellen Raum über ein Modell repräsentiert. Die Bewegungen des Menschen werden mit Hilfe einer Kinect® über einen Motion Capturing Ansatz aufgenommen. Diese Informationen werden in das virtuelle Modell gespeist, um das digitale Menschmodell zu erstellen. Basierend auf diesem Menschmodell wird im Betrieb die Roboterbahn berechnet und dem Roboter zur Verfügung gestellt.

PELLEGRINELLI ET AL. (2017) beschäftigen sich mit einer integrierten Bahnplanung für den kollaborativen Betrieb. Nach Definition der Aufgaben werden pro Aufgabe mehrere Trajektorien zur Durchführung dieser Aufgabe hinterlegt. Basierend auf der Sicherheitslage und den Zykluszeiten der unterschiedlichen Trajektorien, wird zur Laufzeit die Trajektorie für eine Aufgabe ausgewählt. Die Ausführungszeit des Menschen wird in dieser Arbeit über eine Kinect® identifiziert. Dauert eine Aufgabe länger als geplant, findet eine Umplanung der Aufgaben statt. Die Beendigung einer Aufgabe des Menschen signalisiert dieser über ein Myo-Armband (PELLEGRINELLI ET AL. 2017). Ein Myo-Armband beinhaltet einen elektromyografischen Sensor, mit dem die Muskelaktivitäten unter der Haut gemessen werden. Daher können damit Handgesten und Bewegungen detektiert werden (RAWAT ET AL. 2016).

Einhergehend mit der Betrachtung der Kollisionsvermeidung bzgl. der Sicherheit im kollaborativen Betrieb, verbessert eine kollisionsfreie Bahnplanung die Zusammenarbeit. Wie bereits im vorherigen Abschnitt beschrieben, stellen MEZIANE ET AL. (2017) einen Ansatz zur kollisionsfreien Bahnplanung vor.

BORTOT ET AL. (2010) versuchen in ihrem Ansatz, neben der Sicherheit in einer Mensch-Roboter-Kooperation auch die Effizienz zu erhöhen. Dazu integrieren sie Modelle über das menschliche Bewegungsverhalten in die Robotersteuerung. Die Bewegung des Nutzers soll damit vorhergesehen und diese Information in die Planung der Trajektorien integriert werden. Das Ergebnis ihrer Arbeit zeigt auf, dass die Modelle des Menschen nicht genügend genau waren, um die Bewegungen ausreichend gut vorherzusagen zu können.

AWAIS & HENRICH (2010) beschreiben einen Ansatz zur Intentionserkennung für eine Mensch-Roboter-Kooperation, die auf wahrscheinlichkeitsbasierten Zustandsmaschinen beruht. Zur Aufnahme der Daten wird eine Firewire Digitalkamera verwendet. Die Autoren unterscheiden dabei zwischen impliziten und expliziten Intentionen, die beide mit Hilfe des Systems erkannt werden.

### 3.3.3 Tätigkeitserkennung

Wie im Abschnitt zuvor beschrieben, wird über unterschiedliche Ansätze der Mensch detektiert, um seine aktuell ausgeführte Tätigkeit oder seine Intention zu erkennen, damit sich der Roboter daran anpassen kann. Neben den bereits angesprochenen Ansätzen können dazu Hidden Markov Modelle verwendet werden, wie WARD ET AL. (2006), ROITBERG ET AL. (2014) oder SHEN (2015) zeigen.

Zur Erkennung von Tätigkeiten an einem PC-Arbeitsplatz nutzen PERDIKIS ET AL. (2008) in ihrer Arbeit Layered Hidden Markov Modelle. Die Autoren arbeiten darin mit zwei Schichten und einer Agenten-Integration für die zweite Schicht. Der untere Layer repräsentiert kurze, primitive Bewegungen zu einem Ziel, während in der zweiten Schicht abstraktere Aktionen mit einer größeren Zeitdauer beschrieben werden. Die Aufnahme der Positionen des Menschen erfolgt dabei mit Hilfe von Kameras (PERDIKIS ET AL. 2008).

WARD ET AL. (2006) beschreiben einen Ansatz, um mit Hilfe von Hidden Markov Modellen Montagetätigkeiten zu erkennen. Um Eingangsdaten für das Modell zu generieren, nutzen sie Mikrofone zur Erfassung von Tönen und Beschleunigungssensoren zur Erfassung von Bewegungen an zwei Stellen am Arm. Die Mikrofondaten werden mit einer linearen Diskriminierungsanalyse ausgewertet.

ROITBERG ET AL. (2014) verfolgen mit ihrem Ansatz die Identifikation von Montagetätigkeiten im Umfeld der Mensch-Roboter-Interaktion. Zur Detektion der Position des Menschen werden die Daten zweier Kinect® Kameras kombiniert. Mit drei Klassen von Aktivitäten, die Bewegungen, Gesten und Bewegungen von Objekten umfassen, wird mit einer Kombination aus Random Forest und Hidden Markov Modellen die Tätigkeit identifiziert. Darauf aufbauend wurde die Detektion der Bewegungen um weitere Kamerasysteme, wie einer Asus Xtion Kamera und einem Leap Motion Controller, erweitert. Die Erkennung wurde an vier Szenarien erprobt.

SHEN (2015) hat ein System zur sicheren Mensch-Roboter-Koexistenz in der Fließmontage erarbeitet. Dabei war das Ziel, den Roboter nach einer Tätigkeit durch

einen Werker wieder mit seiner geplanten Bahn zu synchronisieren. Zur frühzeitigen Erkennung der durchgeführten Tätigkeit des Werkers setzt SHEN (2015) ein Layerd Hidden Markov Modell ein. Dabei sind die Tätigkeiten vorher definiert und werden über Beobachtungssequenzen beschrieben. Als Eingangsdaten für das Modell verwendet er die Position des Werkers, die Bewegung des Werkers und die Position des Montagehauptkörpers. Der Werker als auch der Montagehauptkörper werden dabei mit Hilfe eines Kamerasystems detektiert. Um sein Modell einzusetzen, führt SHEN (2015) ein Training für sein Modell durch. Dabei identifiziert er je Tätigkeit vier Modelle, für die jeweils die Trainingsalgorithmen durchlaufen werden müssen.

#### **3.3.4 Arbeitswissenschaftliche Studien zur gemeinsamen Arbeit**

HOFFMAN & BREAZEL (2007) untersuchen in ihrer Arbeit den Einfluss der Antizipation eines Roboters auf die Effizienz, die Flüssigkeit des Arbeitsablaufs und die Empfindung des Menschen bei der Zusammenarbeit von Mensch und Roboter. Als Aufbau diente hierzu ein gemeinsamer Arbeitsplatz, wobei ein Auto zusammengebaut werden sollte. Die Aufgabe des Menschen bestand darin, zwei Bauteile des Autos in den Arbeitsraum zu legen, die der Roboter anschließend mit einem Werkzeug zusammenfügen sollte. Zur Untersuchung der Einflüsse wurde die Studie mit zwei Gruppen durchgeführt, wobei der Roboter bei einer Gruppe nicht antizipieren konnte. Um einen Arbeitsablauf auf dessen Flüssigkeit hin beurteilen zu können, definierten HOFFMAN & BREAZEL (2007) zunächst Messgrößen. Diese umfassen die Zeit zwischen einer Tätigkeit des Menschen und der des Roboters, den Zeitanteil gleichzeitiger Bewegungen der beiden Partner und die Wartezeit des Werkers. Als Ergebnis konnte festgestellt werden, dass die Zeit zwischen einer Tätigkeit des Menschen und der eines Roboters geringer war, wenn der Roboter über antizipatorische Fähigkeiten verfügte. Ebenso war der Zeitanteil von gleichzeitigen Bewegungen in diesem Fall höher, wodurch der Ablauf als besser synchronisiert bezeichnet werden kann. Die Wartezeit des Werkers hingegen war in beiden untersuchten Fällen ähnlich. Generell konnte in der Studie festgestellt werden, dass eine Antizipation des Roboters durch den Menschen erwartet wird.

KOPPENBORG ET AL. (2013) führten eine Studie durch, um die Auswirkungen der Autonomie und Geschwindigkeit des Roboters in einer virtuellen Mensch-Roboter-Kollaboration zu untersuchen. Dazu wurde ein Arbeitsplatz in einer virtuellen Umwelt aufgebaut. Der Mensch musste zunächst eine Produktionsaufgabe und anschließend gemeinsam mit dem Roboter eine Qualitätsprüfung durchführen. Zur Evaluation wurde ein faktorielles Design verwendet, wobei der Einfluss unterschiedlicher

Geschwindigkeiten und des Grads der Autonomie des Roboters hinsichtlich der Performanz, der erlebten Anstrengung, der Gefahrenkognition und des Angstempfindens ermittelt wurde. Eine erhöhte Autonomie des Roboters wurde durch häufige Richtungswechsel und Abweichungen in der Bewegungsbahn dargestellt, so dass die Bewegung des Roboters als unvorhersehbar für den Menschen bezeichnet werden konnte. Die Studie ergab, dass kürzere Antwortzeiten und damit eine schnellere Bearbeitung der Aufgabe bei vorhersehbaren Bewegungen, also geringerer Autonomie des Roboters, erreicht wurden. Ebenso führten autonome und schnelle Bewegungen zu einer erhöhten Beanspruchung des Menschen. Dieser Sachverhalt deckt sich mit den Ergebnissen von DEHAIS ET AL. (2011).

HUBER ET AL. (2013) untersuchen in ihrer Arbeit verschiedene Strategien zur Zusammenarbeit zwischen Mensch und Roboter bei einer Übergabeaufgabe. Sie unterscheiden dabei zwischen vier Strategien, die evaluiert werden. Ihre erste Strategie umfasst die Verwendung von Durchschnittszeiten für die Montage, die über mehrere Durchläufe mit unterschiedlichen Personen ermittelt wurden. Bei der zweiten Strategie ist ein Signal der Person Startzeitpunkt für die Bewegung des Roboters, der auf das Signal reagiert. Im dritten Fall wird die Durchschnittszeit für eine Komponente verwendet. Dabei präsentiert der Roboter der Person das Teil, nachdem die Zeit abgelaufen ist. Die vierte Strategie basiert auf einer Abschätzung der Restzeit nach HUBER ET AL. (2010). Dabei kommt ein wahrscheinlichkeitsbasiertes Bayesisches Netzwerk zum Tragen. Das Ergebnis der Studie zeigt, dass die Strategie, bei der ein Sensor verwendet wird bzw. das Signal durch die Person erfolgt, die längsten Wartezeiten zur Folge hat. Das beste Ergebnis konnte mit der Abschätzung der Restzeit erreicht werden, wobei die Summe der Wartezeiten am geringsten war. Eine weitere Feststellung der Studie war, dass die arbeitende Person ihre Arbeitsgeschwindigkeit stets beibehalten hat.

DRAGAN ET AL. (2015) untersuchen in ihrer Arbeit den Einfluss der Roboterbewegung auf die Zusammenarbeit zwischen Mensch und Roboter. Dabei unterscheiden sie zwischen drei Bewegungsarten. Bei der funktionalen Bewegung sollen das Ziel erreicht und Kollisionen vermieden werden. Die vorhersehbare Bewegung ist die Bewegung, die ein Mensch bei bekanntem Ziel erwarten würde. Bei lesbaren Bewegungen ist das Ziel unbekannt, doch kann der Mensch über die Bewegung Schlüsse über das Ziel ziehen. Dabei wurde festgestellt, dass die vorhersehbaren Bewegungen die Erwartung am ehesten treffen und die Flüssigkeit des Ablaufs positiv beeinflussen. Überraschende Bewegungen werden als unkomfortabel beschrieben.



Für den Betrieb in einer Mensch-Roboter-Kooperation zeigen Studien auf, dass eine flüssigere und effizientere Zusammenarbeit möglich ist, wenn der Roboter Tätigkeiten des Menschen antizipieren kann. Jedoch zeigen weitere Studien, dass der Autonomiegrad des Roboters nicht zu hoch sein sollte. Zur Zusammenarbeit zwischen Mensch und Roboter werden daher Ansätze untersucht, um die Tätigkeiten des Menschen zu erkennen, so dass das Robotersystem darauf reagieren kann. Dazu werden meist Wahrscheinlichkeitsmodelle, wie zum Beispiel Hidden Markov Modelle, verwendet. Auf diesen Ansätzen kann aufgebaut werden. Zur Integration in das Programmiersystem muss allerdings überprüft werden, inwiefern bei diesen Ansätzen das Training reduziert werden kann, damit das im Rahmen dieser Arbeit verfolgte Ziel eines flexiblen Einsatzes des Systems erreicht werden kann. Zur Detektion von möglichen Kollisionen vor deren Eintritt werden Ansätze vorgestellt, die meist kamerabasiert sind. Dabei wird der Arbeitsraum überwacht und das System entweder vor einer möglichen Kollision zum Stillstand gebracht oder es wird eine alternative Bahn generiert.

### **3.4 Zusammenfassung und Bewertung des Standes der Technik**

Dieser Abschnitt beinhaltet eine Zusammenfassung und Bewertung der vorgestellten Arbeiten. Die Ansätze, die im Bezug zur Zielstellung betrachtet wurden, können in die drei Bereiche Planung einer Mensch-Roboter-Kooperation, die vereinfachte Programmierung von Robotern und den Betrieb der Applikation eingeteilt werden. Hinsichtlich der Zielsetzung wurden Ansätze dieser drei Bereiche auf folgende Aspekte hin untersucht und bewertet (vgl. Abbildung 12):

#### **Vereinfachte Programmierung**

- *Aufgabenorientierte Programmierung*: Mit diesem Kriterium wird überprüft, ob der Ansatz der aufgabenorientierten Programmierung verwendet wird.
- *Alternative Programmierung*: Hiermit werden Arbeiten gekennzeichnet, die sich mit Programmiersystemen beschäftigen, die nicht der aufgabenorientierten Programmierung zugeordnet werden können.
- *Betrachtung des Menschen*: Damit wird überprüft, ob der Mensch bei der Programmerstellung als Schnittstelle betrachtet wird.

#### **Planung**

- *Beschreibung von Aufgaben*: Hierbei wird evaluiert, ob die Arbeit eine Beschreibung der Aufgaben, insbesondere die Definition der Aufgabenebene

enthält. Bei der Beschreibung wird vor allem bewertet, welche Informationen zu jeder Aufgabe vorhanden sind.

- *Beschreibung von Fähigkeiten:* Dabei wird betrachtet, inwiefern die Fähigkeiten von Ressourcen beschrieben werden.
- *Abgleich der Fähigkeiten:* Mit diesem Kriterium wird beurteilt, nach welchen Kriterien und mit welchen Algorithmen die Aufgaben den Ressourcen zugeteilt werden. Dies schließt einen Anforderungs-Fähigkeitsabgleich ein.
- *Simulation:* Dies beschreibt die Anwendung einer Simulation in den unterschiedlichen Ansätzen. Der Einsatz einer Simulation kann im Planungsprozess oder im Programmierprozess erfolgen.

#### **Betrieb in der Mensch-Roboter-Zusammenarbeit**

- *Sicherheit:* Im Rahmen der Mensch-Roboter-Kooperation bildet der Bereich Sicherheit ein großes Aufgabengebiet. Hierbei werden sowohl die Sicherheit in Bezug auf Normen und Richtlinien, wie z. B. Vorgehensweisen zur CE-Kennzeichnung einer Applikation, als auch Forschungsthemen im Umfeld der Sicherheit, wie z. B. Strategien und Technologien zur Vermeidung von Kollisionen, beschrieben.
- *Kollaborierende Arbeit:* In diesem Kriterium wird betrachtet, inwiefern der Ansatz Modelle untersucht, die zu einer Verbesserung der Zusammenarbeit führen. Dies können Methoden sein, um die aktuell durchgeführte Tätigkeit des Menschen zu erkennen, oder die Untersuchung, wie die Zusammenarbeit flüssiger erfolgen kann.
- *Arbeitswissenschaftliche Studien:* Die Zusammenarbeit mit einem technischen System erfordert die Betrachtung von psychologischen Aspekten. So wird bspw. untersucht, welche Aufgaben Roboter und Mensch übernehmen sollen, wie die Bewegungen des Roboters erfolgen sollen und wie autonom der Roboter sein darf.

Die im Stand der Wissenschaft und Technik vorgestellten Arbeiten wurden den beschriebenen Aspekten gegenübergestellt und es wurde geprüft, inwiefern das Thema in dem jeweiligen Ansatz betrachtet wird. Das Ergebnis dieser Bewertung ist in nachfolgender Abbildung 12 dargestellt.

Die Bewertung zeigt, dass es für jede der drei Phasen Planung, Programmierung und Betrieb Ansätze gibt. Es ist zu erkennen, dass sich viele Arbeiten im Bereich der Planung einer Mensch-Roboter-Kooperation oder der Mensch-Roboter-Zusammenarbeit nicht mit der Programmierung des Roboters beschäftigen. Bei Betrachtung der Ansätze zur Programmierung von Robotern wird deutlich, dass sich





## 4 Anforderungsanalyse an ein Programmiersystem für die Mensch-Roboter-Kooperation

### 4.1 Vorgehen zur Anforderungsanalyse

Aus den Herausforderungen eines flexibleren Einsatzes von Robotern im Rahmen einer Mensch-Roboter-Kooperation ergeben sich Anforderungen an die Programmierung, so dass diese von Nicht-Experten ausgeführt werden kann. Bei Nicht-Experten handelt es sich um alle Personen, die mit dem Robotersystem arbeiten, aber keine besondere Qualifizierung zur Programmierung von Robotern haben. Damit verbunden ergibt sich der Bedarf, die das System nutzende Person bei der Aufgabenverteilung zu den Ressourcen zu unterstützen, da dies einen komplexen Vorgang darstellt. Als weiterer wichtiger Punkt für den Einsatz der Mensch-Roboter-Kooperation wurde die Adaption des Roboters in der Betriebsphase identifiziert. Die Aufteilung der Aufgaben durch den Nutzer und die damit verbundene Generierung des Roboterprogramms bieten Potenzial, eine gewisse Adaptionfähigkeit des Roboters bereits bei der Programmgenerierung zu integrieren, jedoch den Autonomiegrad des Roboters geeignet zu gestalten. Die Tätigkeitsfelder in den drei Phasen fasst Abbildung 13 zusammen. Basierend auf diesen Tätigkeitsfeldern und dem Stand der Technik werden nachfolgend für jede Phase und der Sicherheit Anforderungen an das Programmiersystem beschrieben. Diese werden anschließend zusammengeführt, um Anforderungen an das Gesamtsystem zu definieren.

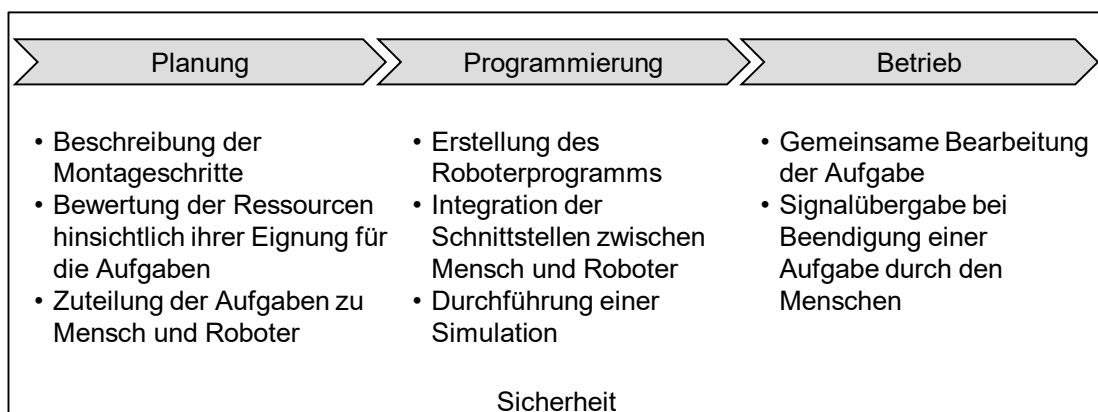


Abbildung 13: Tätigkeitsfelder der Phasen einer Mensch-Roboter-Kooperation

### 4.2 Planung

Ansätze zur Planung von MRK-Applikationen wurden in Abschnitt 3.2 dargelegt. Es ist festzustellen, dass eine Umplanung einer MRK-Applikation Expertenkenntnis und damit einen hohen Aufwand erfordert (LEMMERZ ET AL. 2018).

Bei der Planung einer Mensch-Roboter-Kooperation müssen die Fähigkeiten der Ressourcen mit den Anforderungen an die zu erfüllende Aufgabe abgeglichen werden, wozu die Aufgaben zunächst beschrieben werden müssen. Anschließend muss die zeitliche Planung der Aufgaben erfolgen. Um diesen Prozess zu unterstützen, ergeben sich folgende Anforderungen an das Gesamtsystem:

- Die durchzuführenden Aufgaben und deren Eigenschaften müssen beschrieben werden.
- Es soll ein Abgleich der Fähigkeiten der Ressourcen mit den Anforderungen an die Aufgabe erfolgen.
- Die Aufgaben sollen zeitlich den Ressourcen unter Beachtung der Fähigkeiten und Ziele zugeordnet werden, so dass ein zeitlicher Ablauf entsteht.

### 4.3 Programmierung

Die Programmierung von Robotern speziell für den Einsatz in der Mensch-Roboter-Kooperation wurde in der Literatur bisher kaum betrachtet. In bestehenden Ansätzen zur Programmierung wird nur der Roboter betrachtet, jedoch nicht die Abhängigkeit zwischen den beiden Ressourcen Mensch und Roboter, die besondere Anforderungen an das Programm in einer Mensch-Roboter-Kooperation stellt. Ebenso stellt die Abhängigkeit zwischen den Aufgaben Anforderungen an die Benutzerschnittstelle, die in Abschnitt 4.3.3 beschrieben sind. Zuvor müssen jedoch noch die Nutzerrollen des Programmiersystems analysiert und definiert werden.

#### 4.3.1 Programmgenerierung

Die gleichzeitige Betrachtung von Mensch und Roboter im Programmierprozess und die damit verbundene Integration der Kommunikation zwischen den beiden Partnern ist kaum in der Literatur zu finden. Zur Programmgenerierung für die Mensch-Roboter-Kooperation ergeben sich daher folgende Anforderungen:

- Bei der Programmerstellung müssen die Abhängigkeiten zwischen den Ressourcen integriert werden.

- Die Interaktion zwischen Mensch und Roboter für den Betrieb muss im Programm vorgesehen werden.
- Sicherheitselemente sollten bei der Programmgenerierung integriert werden.
- Die Roboterbewegungen sollten möglichst vordefiniert und dem Menschen bekannt sein.
- Das Roboterprogramm soll auf der Robotersteuerung ablaufen.

### 4.3.2 Analyse der Nutzergruppen

Das Programmiersystem wird von unterschiedlichen Nutzergruppen bedient. Um für die verschiedenen Nutzergruppen Funktionen zur Verfügung zu stellen, werden diese nachstehend analysiert. BACKHAUS (2016) beschreibt in seiner Arbeit die Nutzergruppen des aufgabenorientierten Programmiersystems in Anlehnung an EHRMANN (2007). Er unterscheidet dabei zwischen Prozessexperte, Anwender und Bediener. Bei dem *Bediener* handelt es sich um denjenigen, der im Betrieb mit dem Montagesystem zusammenarbeitet und hauptsächlich den Prozess überwacht, da es sich um automatisierte Montageanlagen handelt. Der *Anwender* hat nur geringes Wissen bzgl. Programmierung, kennt jedoch die Ressourcen des Montagesystems. Ein *Prozessexperte* hat Erfahrung mit den Montageprozessen und der Simulation, jedoch fehlen ihm die Programmierkenntnisse. Da in den vorhergehenden Arbeiten vorrangig automatisierte Montageanlagen betrachtet wurden, hier jedoch der Fokus auf der Mensch-Roboter-Kooperation liegt, wird die Analyse überprüft und angepasst. Der Bediener des Roboters soll mit diesem zusammenarbeiten. Der Bediener entspricht dabei der Person, die an dieser Anlage arbeitet und wird nachfolgend als Werkskraft bezeichnet. Die Werkkraft soll bei der Zuteilung der Aufgaben mitentscheiden können. Daher wird als weiteres Kriterium die Zuteilung von Aufgaben hinzugefügt und bewertet. Je nach Entscheidung der Geschäftsprozesse kann die Werkkraft in diesem Fall die Aufgabe des Programmierers vornehmen. Auf diesen Fall wird in Kapitel 7 noch näher eingegangen. In der Anwendung ist der Bediener auch gleichzeitig der Anwender, da er sowohl im Betrieb mit dem System zusammenarbeitet als auch die Ressourcen kennt. Ist es nicht möglich, dass der Bediener die Programmierung vornimmt, gibt es noch die Rolle des Programmierers. Der Prozessexperte bleibt weiterhin die Person mit Kenntnissen der Montageprozesse. Das Ergebnis der Bewertung mit den veränderten Betrachtungsbereichen zeigt die nachfolgende Abbildung 14.

## 4 Anforderungsanalyse an ein Programmiersystem für die Mensch-Roboter-Kooperation

Kenntnisbereich \ Nutzergruppe	Funktionswissen	Prozesswissen	Erfahrung Simulations-systeme	Programmierenkenntnisse	Zuteilung Aufgaben in einer MRK
Programmierer	●	●	○	●	○
Werkskraft	●	●	○	●	●
Prozessexperte	●	●	●	●	●

○ kaum vorhanden   ● gering   ● durchschnittlich   ● umfangreich   ● vollständig vorhanden

Abbildung 14: Analyse der Nutzergruppen und deren Kenntnisbereiche basierend auf BACKHAUS (2016)

Die Benutzerschnittstelle soll vorrangig von den Nutzergruppen Programmierer und Werkkraft genutzt werden können. Ein Prozessexperte, der beispielsweise die Bibliotheken erweitert, greift auf diese an einem PC zu und arbeitet nicht mit der Benutzerschnittstelle. Für Programmierer als auch Bediener soll die Oberfläche so gestaltet sein, dass sie für beide Nutzergruppen anwendbar ist. Insbesondere für die Nutzergruppe des Bedieners müssen die Informationen daher einfach gestaltet sein. Als Anforderung aus der Analyse der Benutzergruppen ergibt sich daher:

- Die Informationen auf der Benutzerschnittstelle müssen so gestaltet sein, dass eine Werkkraft ohne Programmierkenntnisse damit umgehen kann.
- Die Erweiterung des Systems kann durch einen Prozessexperten vorgenommen werden.

### 4.3.3 Benutzerschnittstelle

Eine Benutzerschnittstelle stellt die Kommunikationsschnittstelle zwischen Mensch und Rechner dar (VOSS & NENTWIG 1998). Im Falle der aufgabenorientierten Programmierung von Robotern stellt sie Funktionen zur Verfügung, um Aufgaben für den Roboter eingeben zu können, die nachfolgend in ein spezifisches Roboterprogramm umgewandelt werden. Allgemein werden die Anforderungen an Benutzerschnittstellen in der DIN EN ISO 9241-210 beschrieben. Zur Entwicklung von Benutzerschnittstellen schlagen BUTZ & KRÜGER (2014) eine iterative Vorgehensweise vor, bei der Zwischenergebnisse durch Prototypen evaluiert werden können. Die Norm beschreibt dazu einen benutzerorientierten Entwicklungsprozess, der aus vier Phasen besteht.

Im Stand der Technik werden Ansätze zu Benutzerschnittstellen für die Mensch-Roboter-Kooperation im Forschungskontext und kommerzielle Programmiersysteme dargelegt. Bei diesen, in Abschnitt 3.1.2 eingeführten Systemen, wie Drag & Bot, Artiminds oder der Benutzerschnittstelle des Roboters Panda, können zwar die



Abläufe des Roboters dargestellt werden, jedoch werden Aufgaben des Menschen nicht repräsentiert. Demgegenüber stellen TSAROUCHI ET AL. (2016A) eine Schnittstelle zur Verfügung, auf der die Aufgaben von Roboter und Mensch in Form eines Gantt-Diagramms dargestellt werden, jedoch nicht über Drag & Drop-Eigenschaften verfügen. An diese Art der Darstellung kann angeknüpft werden.

Die Anforderungen an die Benutzerschnittstelle für die aufgabenorientierte Programmierung in der Mensch-Roboter-Kooperation können daher wie folgt zusammengefasst werden:

- Die Schnittstelle dient zur Eingabe und Veränderung der Aufgaben für Mensch und Roboter.
- Bei der die Schnittstelle nutzenden Person handelt es sich um die Werkkraft, die gemeinsam mit dem Roboter eine Montageaufgabe bearbeitet, oder um einen Programmierer.
- Es sollen die Aufgaben von Mensch und Roboter und deren Abhängigkeiten auf der Oberfläche dargestellt werden.
- Die Zuteilung der Aufgaben zu den beiden Ressourcen Mensch und Roboter sollte auf der Benutzerschnittstelle dargestellt werden, um den zeitlichen Ablauf zu präsentieren und die Abhängigkeiten abzuleiten.
- Die Zusammenstellung der Aufgaben soll über ein Drag & Drop-Verfahren ermöglicht werden.

## 4.4 Betrieb

Bezüglich des Betriebs einer Mensch-Roboter-Kooperation werden im Stand der Technik die Kategorien der Kollisionsvermeidung und der Zusammenarbeit zwischen den Ressourcen beleuchtet. Innerhalb der Kollisionsvermeidung werden Ansätze beschrieben, wie mögliche Kollisionen mit dem Roboter detektiert werden können. Hinsichtlich der Zusammenarbeit wurde festgestellt, dass die Bewegungen des Roboters vorhersehbar sein sollen. Jedoch soll der Roboter über antizipatorische Fähigkeiten verfügen, um einen fließenden Ablauf abbilden zu können. Zur Antizipation des Roboters kann an die vorgestellten Methoden angelehnt werden, um die aktuell durchgeführte Tätigkeit des Menschen zu erkennen. Daraus können folgende Anforderungen für den Betrieb abgeleitet werden.

- Durch die Erkennung der aktuellen Tätigkeit des Menschen über ein System kann der Roboter darauf reagieren; dies kann als Antizipation des Roboters beschrieben werden.

- Es soll eine automatische Signalweitergabe an das Robotersystem zur effizienteren Aufgabenabarbeitung in der Zusammenarbeit erfolgen.
- Die Bewegungen des Roboters sollen vorhersehbar sein.
- Mögliche Kollisionen sollen vor deren Eintritt detektiert werden.

### 4.5 Sicherheit

Zur Implementierung einer Applikation einer Mensch-Roboter-Kooperation müssen die in Abschnitt 2.4.4 betrachteten Normen angewandt werden. Zum Erlangen der CE-Konformität muss die komplette Applikation betrachtet werden, was den Roboter, die Roboterwerkzeuge, die Werkstücke, die Fördertechnik und allgemein die Umgebung des Roboters in seinem Arbeitsraum einschließt (FACHBEREICH HOLZ UND METALL DER DGUV 2017). Die Risikoanalyse zur CE-Kennzeichnung muss nach einer Änderung der Applikation erneut durchgeführt werden (VDMA ROBOTICS + AUTOMATION 2016). Dass dieser Aspekt zu Hemmnissen bei der Einführung entsprechender Applikationen führt, zeigt die Studie des Fraunhofer Austria. Um mehr Applikationen der MRK einzuführen, muss daher identifiziert werden, wie der Aspekt der Sicherheit auf veränderte Weise überprüft werden kann, so dass weiterhin die hohe Sicherheit gewährleistet ist, jedoch der Aufwand diesbezüglich verringert werden kann.

Um das Potenzial des Programmiersystems für die Mensch-Roboter-Kooperation mit dem Ziel einer schnellen Applikationsanpassung erreichen zu können, ergeben sich zusammengefasst folgende Anforderungen:

- Sicherheitsaspekte sollen in den Programmierprozess integriert werden.
- Vor der Inbetriebnahme soll die zuständige Person bei der Risikoanalyse unterstützt werden.
- Der Aufwand für die CE-Kennzeichnung soll dadurch verringert werden.

### 4.6 Zusammenfassung der Anforderungen an das aufgabenorientierte Programmiersystem

In den vorherigen Abschnitten wurden Anforderungen für die verschiedenen Phasen des Systems und übergreifend für die Sicherheit abgeleitet. Insgesamt können die Anforderungen an das Programmiersystem folgendermaßen zusammengefasst werden:

- Die Werkskraft soll beim Planungsprozess unterstützt werden.
- Die Werkkraft soll bei der Planung mitentscheiden können.
- Die Schnittstellen für den Betrieb einer Mensch-Roboter-Kooperation sollen bereits während der Programmerstellung generiert und integriert werden.
- Die Voraussetzungen für die Zusammenarbeit im Betrieb nach arbeitswissenschaftlichen Erkenntnissen sollen geschaffen werden, wozu Tätigkeiten mit dem Ziel der Antizipation des Roboters erkannt werden sollen.
- Es soll eine Benutzerschnittstelle zur Interaktion der nutzenden Person mit dem technischen System geben.
- Der Prozess zur Erlangung der CE-Konformität soll unterstützt werden.

Zur Kommunikation der nutzenden Person mit dem System muss eine Benutzerschnittstelle zur Verfügung gestellt werden. Damit kann das System Aufgaben entgegennehmen und es können wiederum Informationen angezeigt werden. Außerdem dient diese Schnittstelle zur Bedienung des Gesamtsystems und muss Funktionen zum Starten und Beenden enthalten. Da die Zuteilung der Aufgaben zu Mensch und Roboter über diese Schnittstelle erfolgt, muss eine Möglichkeit bestehen, diese Aufgaben dort anzuzeigen, zu bearbeiten und so darzustellen, dass eine Zuordnung zu den Ressourcen erfolgen kann. Um der nutzenden Person die Programmierung zu vereinfachen und den Anforderungen einer aufgabenorientierten Programmierung nach Abschnitt 4.7 gerecht zu werden, muss zunächst eine Ebene definiert werden, auf der die Aufgaben beschrieben werden. Insgesamt soll die Benutzerschnittstelle nach Richtlinien und Handlungsempfehlungen entwickelt werden, um die Usability sicherzustellen.

Das Programmiersystem stellt Funktionen für den Einsatz von Ressourcen unterschiedlicher Hersteller zur Verfügung. Damit soll das System erweiterbar um Ressourcen, wie Roboter oder Greifer unterschiedlicher Hersteller, sein.

Ein hoher Grad an Autonomie des Roboters, d.h. selbständiger Entscheidung, kann die Leistung des Werkers aufgrund nicht vorhersehbarer Bewegungen senken. Daher sollte die Roboterbahn vor dem Betrieb definiert und der Mensch darauf vorbereitet werden. Für den Betrieb gilt, dass durch eine Erkennung der Tätigkeiten des Menschen mit Hilfe des Systems eine flüssigere und damit effizientere Arbeit möglich ist. Für das System ergibt sich daraus die Anforderung, eine Methode zur Verfügung zu stellen, mit der die Tätigkeiten des Menschen bei der Ausführung der zugeordneten Aufgaben erkannt werden können.



## 5 Grundstruktur und Nebenmodule des Programmiersystems

### 5.1 Ableitung der Struktur des Programmiersystems

Zur Ableitung der Architektur des Systems zur aufgabenorientierten Programmierung wurden die bestehenden Ansätze von HUMBURGER (1998), KUGELMANN (1999), WECK & BRECHER (2006), EHRMANN (2007) und BACKHAUS (2016) auf eine mögliche Erweiterung hinsichtlich des Einsatzes für eine Mensch-Roboter-Kooperation und die Abbildung der drei Phasen analysiert.

Das System von HUMBURGER (1998) besteht aus einer Blackboard-Architektur. Als Input gilt ein in einer CAD-Umgebung abgebildetes Produktmodell. Dieses wird in den drei Ebenen Aufgaben-, Vorgangs- und Bewegungsebene analysiert. Der Fokus des Systems von KUGELMANN (1999) liegt auf der Handhabungsplanung, die entsprechend den Hauptbestandteil des Systems abbildet. Die Eingangsgröße ist eine aufgabenorientierte Beschreibung des Handhabungsauftrags. Anschließend wird eine Handhabungsplanung durchlaufen, aus der das RC-Programm abgeleitet wird. Nach dem Einfahren und Verbessern des Programms erfolgt im Betrieb der Abgleich des Simulationslayouts mit dem Programm der Roboterzelle. Das Modell von WECK & BRECHER (2006) wurde in Abschnitt 3.1.1 beschrieben. Es besteht aus drei Modulen, die den drei Phasen Planung, Programmierung und Betrieb zugeordnet werden können. Zusätzliche Informationen werden dem System über das Aufgaben- und Umweltmodell zur Verfügung gestellt. Das Programmiersystem von EHRMANN (2007) besteht aus fünf Ebenen: Hardware-, Modell-, Daten-, Mensch-Maschine-Schnittstellen (MMS)- und Codegenerierungsebene. In der Modellebene werden die Funktionalitäten der Geräte und Steuerungen, die als Hardware vorhanden sind, beschrieben. In der Datenebene werden dem System notwendige Informationen zur Verfügung gestellt und die MMS-Ebene stellt die Schnittstelle zwischen Mensch und System dar. In der Codegenerierungsebene wird der Steuerungscode erzeugt. Das System von BACKHAUS (2016) besteht aus einer Daten-, einer Logik- und einer Präsentationsschicht. Über die Präsentationsschicht erfolgt die Kommunikation mit dem Nutzer. Die Logikschicht umfasst ein Planungsmodul, ein Simulationssystem und einen Postprozessor. In dieser Schicht werden die Informationen verarbeitet und das Steuerungsprogramm generiert. Eine Phase des Betriebs wird nicht betrachtet. Die Kommunikation der Bestandteile der Logikschicht erfolgt durch ein Blackboard.

Bei Betrachtung dieser Ansätze fällt auf, dass das Modell von WECK & BRECHER (2006) durch seine Anlehnung an die Phasen der Planung, der Programmierung und des Betriebs als mögliches Modell für die Erweiterung in Betracht gezogen werden kann. In allen anderen Ansätzen ist keine klare Abbildung dieser Phasen erkennbar. Des Weiteren bildet das Modell von WECK & BRECHER (2006) einen sequentiellen Ablauf ab, was bei den anderen Systemen nicht der Fall ist.

Deshalb wird das Modell von WECK & BRECHER (2006) hinsichtlich der Erweiterung um den Menschen überprüft. Dazu wurde der Bedarf zur Erweiterung in den einzelnen Modulen des bestehenden Systems analysiert. Die notwendigen Anpassungen der Struktur, wie sie in Abschnitt 3.1.1 dargestellt ist, stellt nachfolgende Abbildung 15 dar. Von den anderen Ansätzen können teilweise Elemente für das Programmiersystem für die MRK angepasst und integriert werden.

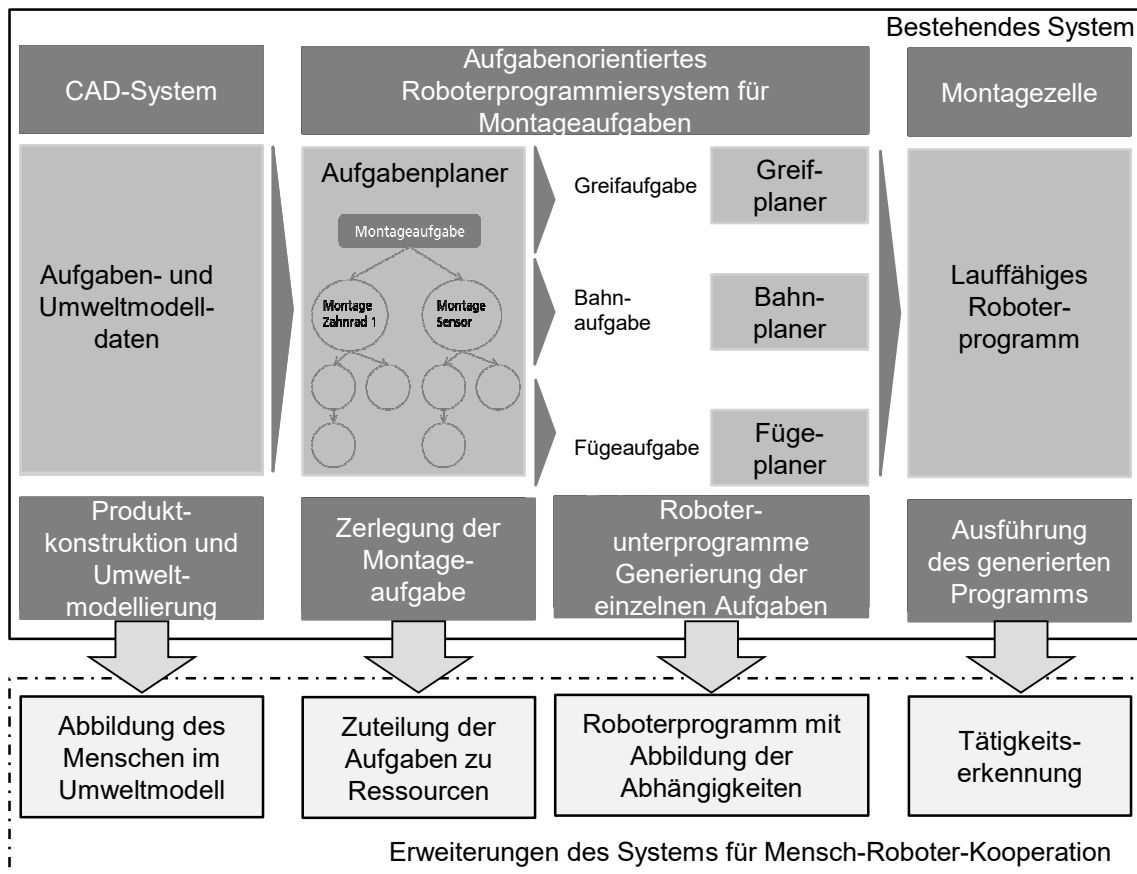


Abbildung 15: Erweiterungsbedarf des aufgabenorientierten Programmiersystems nach WECK & BRECHER (2006) für die Mensch-Roboter-Kooperation

Aus diesem Erweiterungsbedarf wird das System zur Programmierung für die Mensch-Roboter-Kooperation abgeleitet. Abbildung 16 zeigt die Gesamtarchitektur des Systems, die aufbauend auf BERG & REINHART (2017a) erweitert wurde.

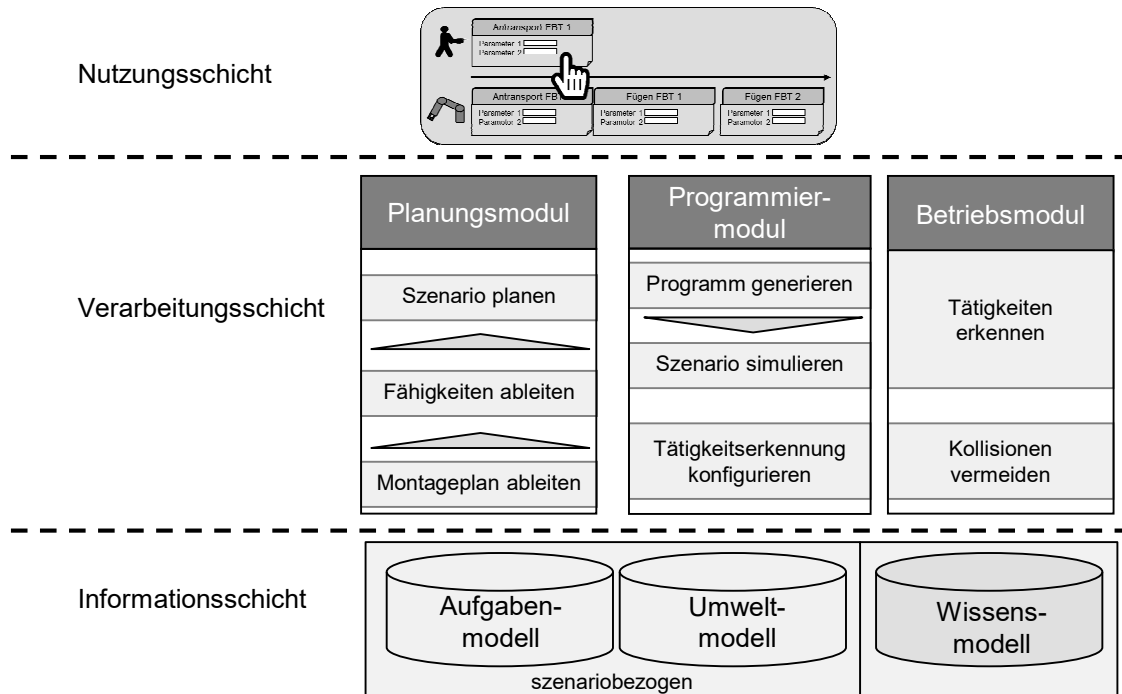


Abbildung 16: Gesamtarchitektur des Programmiersystems

Das System besteht aus drei Schichten: Nutzungsschicht, Verarbeitungsschicht und Informationsschicht. Die Nutzungsschicht ist die Schnittstelle zur nutzenden Person und das, was diese Person sehen kann. In der Verarbeitungsschicht werden die Angaben aus der Nutzungsschicht unter Zuhilfenahme der Informationen aus der Informationsschicht verarbeitet.

Die Informationsschicht umfasst das Aufgaben-, Umwelt- und Wissensmodell. Das Aufgaben- und das Umweltmodell repräsentieren, angelehnt an WECK & BRECHER (2006) und BACKHAUS (2016), das Layout der Applikation und das Produkt in einer 3D-CAD-Umgebung und werden in Abschnitt 5.5 beschrieben. Im Wissensmodell werden für das System notwendige Informationen, wie beispielsweise die vorhandenen Ressourcen und deren Eigenschaften, gesammelt und dem System zur Verfügung gestellt. Des Weiteren werden darin die Aufgabenarten und deren Eigenschaften beschrieben, deren Definition in Abschnitt 5.3 erfolgt.

Die Verarbeitungsschicht ist der Kern des Systems und enthält die Hauptmodule. In dieser Schicht werden die Informationen aus den anderen beiden Schichten verarbeitet. Aus dem Planungsmodul geht ein vorgeschlagenes Szenario für die

Aufteilung der Aufgaben für eine betrachtete Montageaufgabe hervor. Dieser Vorschlag wird dem Nutzer auf der Benutzerschnittstelle aufgezeigt, wo dieser Anpassungen vornehmen kann. Dazu gehören die Parametrierung der Aufgaben, aber auch die Änderung der Zuteilung der Aufgaben, sofern dies aus Prozesssicht möglich ist. Nach Eingabe der Parameter und Bestätigung der Aufgabenaufteilung werden die Informationen im Programmiermodul verarbeitet. Hierbei entsteht das Roboterprogramm, in das die Abhängigkeiten der Aufgaben von Mensch und Roboter integriert werden und das damit für die Tätigkeitserkennung vorbereitet wird. Des Weiteren wird in diesem Modul das Modell der Tätigkeitserkennung für den betrachteten Anwendungsfall konfiguriert. Im Betriebsmodul erfolgt die Tätigkeitserkennung des Menschen, wobei über die Positionsänderungen der Hände und Arme die Bewegungen detektiert werden. Diese Informationen werden in ein Wahrscheinlichkeitsmodell gespeist, mit dem die aktuell durchgeführte Tätigkeit erkannt wird. Die erkannte Tätigkeit wird dann an die Robotersteuerung zurückgegeben, sodass dieser seine nächste geplante Aufgabe starten kann.

Nachfolgend werden die Bestandteile des Systems näher erläutert. In diesem Kapitel wird dabei auf die Randbedingungen des Systems und die Definition der Aufgabenbeschreibung eingegangen, die Grundlage für die Module in der Verarbeitungsschicht ist. Außerdem werden die beiden anderen Schichten, die Informations- und Nutzungsschicht, beschrieben. Da die Verarbeitungsschicht den Fokus der Arbeit abbildet, werden dessen Module in dem separaten Kapitel 6 erläutert.

### **5.2 Randbedingungen des Systems**

Gemäß der Zielsetzung des Programmiersystems für die Mensch-Roboter-Kooperation, das flexibel eingesetzt werden können soll, wurde eine Struktur für das System abgeleitet. Aus der Ausgangssituation geht hervor, dass insbesondere in der Montage bisher wenig Automatisierung vorliegt, da einige Vorgänge noch nicht automatisiert werden können. Allerdings besteht aufgrund des demografischen Wandels und steigender Lohnkosten der Bedarf nach Automatisierung in diesem Bereich. Deshalb soll die Mensch-Roboter-Kooperation insbesondere im Bereich der Montage Anwendung finden, weshalb auch das Programmiersystem hierfür konzipiert wird.

Eine Applikation der Mensch-Roboter-Kooperation stellt einige Anforderungen an die Sicherheit. Um diese im Rahmen des Programmiersystems gewährleisten zu



können, müssen Bedingungen an die Applikation definiert werden. Das Programmiersystem kann eingesetzt werden, wenn Roboter und ein Endeffektor vorhanden sind. Der Roboter sollte der Werkkraft gegenüberstehen und der Arbeitsraum sich auf einem Montagetisch befinden. Die Höhe des Roboters soll so gewählt werden, dass dieser den Arbeitsraum erreichen kann, jedoch nicht auf Kopfhöhe der Werkkraft verfährt. Als Komponenten, wie beispielsweise Endeffektoren, sollten solche verwendet werden, die speziell für den Einsatz in einer MRK konzipiert wurden und so das Verletzungsrisiko verringern. Der Anwendungsbereich des Programmiersystems beschränkt sich auf die in Abschnitt 5.4 beschriebenen Montagevorgänge. Dies ergibt sich daraus, dass diese auf ihre Ausführung in einer MRK hin analysiert wurden.

### 5.3 Beschreibung der Aufgaben

In Abschnitt 3.2.2 wurde bereits auf mögliche Beschreibungsformen von Aufgaben eingegangen. Einhergehend mit dem Ziel einer abstrakten Darstellung und Beschreibung der Aufgaben für die aufgabenorientierte Programmierung muss eine Ebene für die Beschreibung von Aufgaben identifiziert werden. In der Literatur finden sich Beschreibungen auf Ebene der Montageschritte (MÜLLER ET AL. 2016), Beschreibungen auf Prozessebene, die weiter untergliedert werden (TSAROUCHI ET AL. 2016a), bis hin zur Beschreibung der elementaren Bewegungen nach MTM (DEUSE ET AL. 2014). Letztere wird im Bereich der Mensch-Roboter-Kooperation verwendet. BEUMELBURG (2005) verwendet für ihre Arbeit eine Zwischenebene zur Beschreibung der Aufgaben, wie in Abschnitt 3.2.2 dargelegt. Dabei werden die Aufgaben jedoch nicht so einheitlich beschrieben, dass sie von einem maschinellen System verarbeitet werden können. Daher erfolgt nachfolgend die Ableitung der Beschreibungsebene.

Als Anforderung an die Beschreibung der Aufgaben für eine vereinfachte Programmierung in der Mensch-Roboter-Kooperation auf einer abstrakten Ebene sind die elementaren Bewegungen zu detailliert. Die Beschreibung auf Montageschrittebene wird jedoch als zu wenig detailliert angesehen, da hierbei der Schritt nicht weiter unterteilt ist und eine Aufteilung zu Mensch und Roboter nicht möglich ist. LOTTER & WIENDAHL (2006) evaluieren die Aufteilung von Montageaufgaben bei Großgeräten. Sie ordnen dabei einem Primär- oder Sekundärvorgang mehrere elementare Grundbewegungen zu. So umfasst der Vorgang Aufnehmen und Platzieren die Grundbewegungen *Hinlangen*, *Greifen* und *Bringen*. Der Fügevorgang kann dann wiederum als elementarer Vorgang oder als

Bewegungszyklus bewertet werden, wobei ein Bewegungszyklus einen sich wiederholenden Vorgang bezeichnet, wie er beim Eindrehen von Schrauben stattfindet (LOTTER & WIENDAHL 2006).

An diese Art der Beschreibung wird im Folgenden angesetzt, wie auch in BERG ET AL. (2017b) vorgestellt. Es erfolgt eine abstrakte Beschreibung der Aufgaben, die es jedoch zulässt, die Aufgaben zu Mensch und Roboter zuzuteilen. Bei Betrachtung eines Produkts kann jedem Bauteil jeweils ein Montageschritt zugeordnet werden. Dieser Montageschritt wird, angelehnt an LOTTER & WIENDAHL (2006), unterteilt in die Aufgabenblöcke *Antransport Fügebauteil* und *Fügevorgang*. Optional ist der Vorgang *Antransport Fügehilfsteil*, der die Entnahme und das Platzieren von Hilfsteilen, wie z. B. Schrauben, darstellt. Die Vorgänge zum Antransport des Fügebauteils bzw. des Fügehilfsteils fassen die Grundbewegungen Hinlangen, Greifen und Bringen nach MTM zusammen (vgl. Abbildung 17).

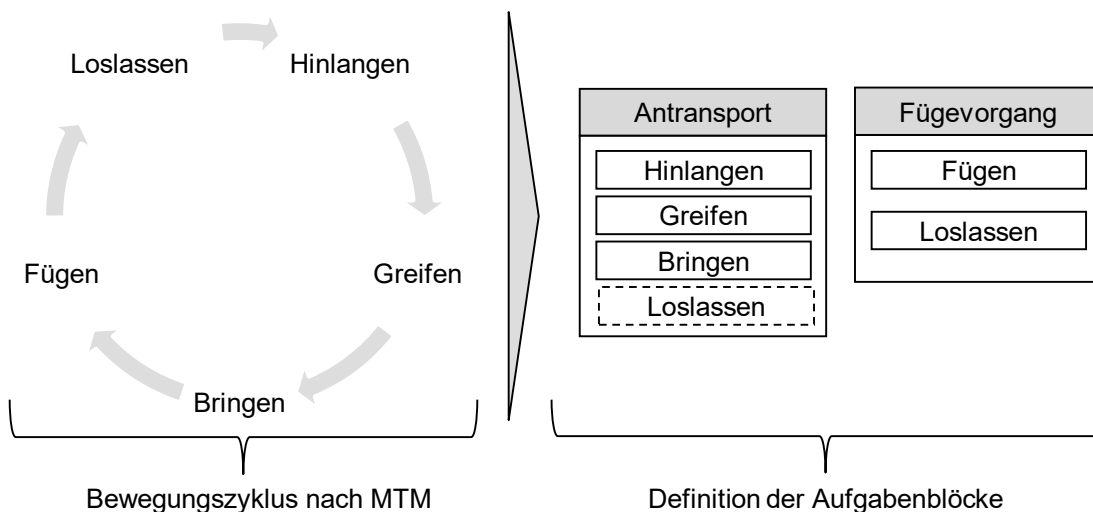


Abbildung 17: Überführung des MTM-Zyklus in Aufgabenblöcke

Wie in Abbildung 17 ersichtlich wird, muss unterschieden werden, ob die gleiche Ressource sowohl den Antransport als auch den Fügevorgang durchführt, das Bauteil ablegt oder an eine andere Ressource übergibt. Wird das Bauteil durch eine Ressource nur antransportiert, so muss die Bewegung Loslassen ausgeführt werden. Wird der Fügevorgang durch die gleiche Ressource unmittelbar angeschlossen, so ist das Loslassen erst nach dem Fügevorgang zu vollziehen. Dies muss bei der Programmerstellung beachtet werden.

Die Eigenschaften der Fügevorgänge und deren Repräsentation im Programmiersystem auf der Benutzerschnittstelle und zur Programmerstellung

variieren je nach Art des Fügevorgangs. Für den Kontext der Mensch-Roboter-Kooperation werden zunächst die Fügevorgänge auf deren Eignung hinsichtlich eines Einsatzes untersucht und basierend darauf die Merkmale zu den Fügevorgängen identifiziert. Die Fügevorgänge und deren Repräsentation werden dem Programmiersystem über das Wissensmodell zur Verfügung gestellt und nachfolgend in Abschnitt 5.5.3 beschrieben.

Die Aufgabenbeschreibung ist Grundlage für die verschiedenen Module des Programmiersystems. Im Planungsmodul werden auf dieser Basis die Aufgaben zu Mensch und Roboter zugeteilt und auf dieser Ebene auf der Benutzerschnittstelle dem Nutzer präsentiert. Im Programmiermodul wird aus diesen Aufgaben das Programm erstellt. Die Identifizierung der Tätigkeiten im Betriebsmodul findet ebenfalls auf dieser Ebene statt.

### **5.4 Analyse von Montageaufgaben für die MRK**

Nach Abbildung 4 gehören zu den Montagevorgängen die Gruppen Fügen, Handhaben, Kontrollieren, Justieren und die Durchführung von Sonderoperationen. Da es sich bei Sonderoperationen, wie bereits durch den Begriff deutlich wird, um besondere, meist einzigartige Operationen handelt, werden diese im Programmiersystem zunächst nicht weiter betrachtet. Die Gruppe Kontrollieren fasst verschiedene Prüf- und Messaufgaben zusammen. Da für diese Aufgabe des Messens meist zusätzliche Messgeräte notwendig sind, werden diese zunächst nicht in das Programmiersystem integriert.

Daher bleiben die Gruppen *Fügen* und *Handhaben* und *Prüfen*, die näher auf Anwendung in einer MRK hin untersucht werden. Hierfür werden zunächst die Roboter betrachtet, die in diesem Kontext Verwendung finden. Es handelt sich um Leichtbauroboter, wie sie in Abschnitt 2.4.3 beschrieben wurden. Der Stand der Technik dieser Roboter weist eine maximale Traglast von 14 kg auf. Basierend auf dieser Tatsache können zunächst einige Fügevorgänge ausgeschlossen werden, bei denen ein Werkzeug mit hohem Gewicht bewegt werden muss oder die Fügevorgänge besonders hohe Kräfte aufweisen.

Die beiden Fügegruppen *Fügen durch Füllen* und *Fügen durch Urformen* werden nicht betrachtet, da in beiden Fällen spezielle Vorrichtungen, wie etwa ein Formwerkzeug, benötigt werden. Durch den zuvor beschriebenen Ausschluss von einigen Fügevorgängen bleiben noch nachfolgende Fügevorgänge zur näheren Betrachtung im Zusammenhang mit dem Programmiersystem für die Mensch-

Roboter-Kooperation: *Zusammensetzen, Anpressen/Einpressen, Fügen durch Umformen, Kleben und textiles Fügen*. Diese Vorgänge können grundsätzlich in einer Mensch-Roboter-Kooperation abgebildet werden. WIELAND (1995) untersucht in seiner Arbeit zur robotergestützten Montage die dort ablaufenden technologischen Vorgänge. Dabei betrachtet er folgende fünf Kategorien: Fügen durch Zusammensetzen, Fügen durch Schrauben, Fügen durch Kleben, Kommissionieren und Handhaben. Beim Vergleich dieser Aufgaben mit den identifizierten Fügevorgängen für die MRK ist festzustellen, dass diese Auflistung mit den identifizierten Fügevorgängen weitestgehend übereinstimmt. Abbildung 18 fasst die in der Arbeit weiter betrachteten Abläufe zusammen.

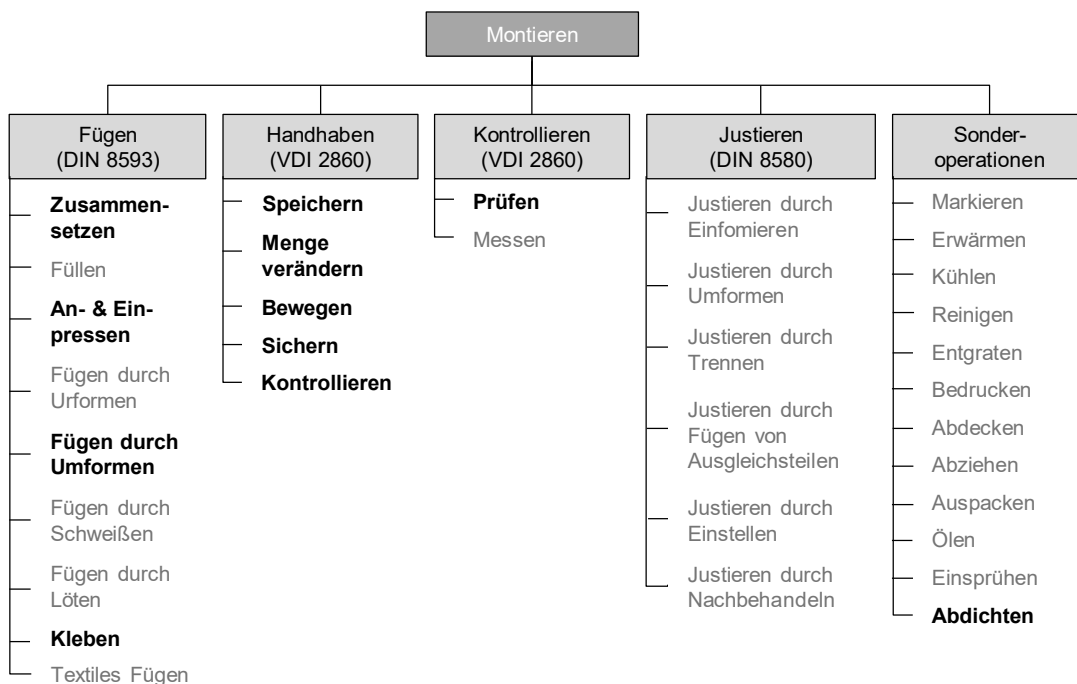


Abbildung 18: Einschränkung der Funktionen der Montage nach LOTTER & WIENDAHL (2006) für das Programmiersystem

## 5.5 Informationsschicht

Zunächst wird das für das Gesamtsystem notwendige und von den einzelnen Modulen verwendete Informationsmodell beschrieben. Nach SCHENCK & WILSON (1994) handelt es sich bei einem Informationsmodell um eine formelle Beschreibung eines Teils der realen Welt durch ein Modell bestehend aus Ideen, Fakten und Prozessen; darüber hinaus beinhaltet es Regeln zur Interpretation. SELIG (2011) ergänzt die Beschreibung dahingehend, dass die Auswahl der darzustellenden Informationen und deren Strukturierung vom Anwendungsfall abhängen. Das Informationsmodell

besteht in diesem System aus einem Aufgabenmodell, in dem die Montageaufgabe modelliert ist, dem Umweltmodell, das die Szene darstellt, und einem Wissensmodell, das notwendiges Wissen über die Ressourcen, deren Parameter und den Aktionen bereitstellt.

### **5.5.1 Aufgabenmodell**

Im Aufgabenmodell enthalten ist die Produkt- oder Baugruppenrepräsentation in einer 3D-CAD-Umgebung. Darin sind die einzelnen Bauteile, deren Eigenschaften und ihre Beziehung zueinander beschrieben. Diese Informationen stehen dem System zur Verfügung, wodurch die Gesamtaufgabe in die einzelnen Montageaufgaben zerlegt werden kann. Aus einem CAD-Produkt können die Montageschritte abgeleitet werden, wobei jedem Bauteil, welches kein Fügehilfsteil ist, ein Montageschritt zugehörig ist. Zu jeder Aufgabe werden die Daten zu dem entsprechenden Bauteil und Informationen zu dem jeweiligen Fügevorgang hinterlegt. Die Methode zur Beschreibung der Aufgaben mit den Daten findet sich im Abschnitt 6.1.1. Die Daten zu den jeweiligen Aufgaben umfassen Eigenschaften zum Bauteil und zum Prozess. Diese Beschreibung wird in einer Form gespeichert und den anderen Modulen des Programmiersystems zur Verfügung gestellt.

Das Aufgabenmodell muss dem System zur Verfügung gestellt werden, indem das CAD-Modell des Produkts auf dem Rechner gespeichert wird. Zur Nutzung des Modells muss das Produkt über bestimmte Beschreibungseigenschaften verfügen. Daher muss der Experte das CAD-Modell hinsichtlich der notwendigen Eigenschaften überprüfen und kann es dann dem System zur Verfügung stellen. Notwendige Anforderungen an das Modell sind beispielsweise die Beziehungen zwischen den Bauteilen.

### **5.5.2 Umweltmodell**

Das Umweltmodell ist nach WECK & BRECHER (2006) die Ausgangsbasis für die aufgabenorientierte Programmierung. Das Umweltmodell umfasst die im Arbeitsbereich auftretenden Ressourcen und simulationsspezifische Inhalte. Für die Applikation einer Mensch-Roboter-Kooperation bedeutet dies, dass das Umweltmodell das Layout des Arbeitsplatzes inklusive der beiden Ressourcen Roboter und Mensch abbildet. Das Umweltmodell liegt in Form einer 3D-Abbildung des Layouts in der später verwendeten Simulationsumgebung vor und wird für die Ableitung von Informationen für die Programmierung als auch die Simulation verwendet. Für die Programmierung und Konfiguration der Tätigkeitserkennung

werden aus dem Umweltmodell die Positionsdaten für die Ausgangsposition des Menschen und die Positionsdaten der Bauteile und der Montageposition abgeleitet.

Das Umweltmodell muss für einen Arbeitsplatz durch einen Experten manuell erstellt werden. Dazu müssen die Komponenten des Arbeitsplatzes als 3D-Modell in das Layout integriert werden bzw. die Positionen der Komponenten oder Ressourcen in dem aktuellen Layout angepasst werden. Um hohe Aufwände für die Anpassung zu vermeiden, ist es zielführend, eine Kombination aus einem Montagetisch und einem Roboter für die meisten Anwendungen zu verwenden, so dass das grobe Layout nicht angepasst werden muss. Um jedoch einen flexiblen Einsatz des Roboters und des Programmiersystems erreichen zu können, können Ansätze, wie von SAND & HENRICH (2014) beschrieben, integriert werden, wobei über eine Kamera-Aufnahme der Umgebung ein 3D-Modell erzeugt wird.

### 5.5.3 Wissensmodell

Wie bereits in Abschnitt 5.3 erwähnt, werden die Bewegungen des Roboters zu den Aufgabenblöcken Antransport und Fügevorgang zusammengefasst. Diese Aufgabenblöcke müssen im System in einem Wissensspeicher hinterlegt werden, um die Aufgaben mit den entsprechenden Parametern zu beschreiben. Dies wird als Wissensmodell bezeichnet. Auf dieses Wissen wird sowohl im Planungsmodul, auf der Benutzerschnittstelle als auch zur Generierung des Roboterprogramms zugegriffen. Es wird als Wissensmodell bezeichnet, da es im Gegensatz zu der Beschreibung von HUMBURGER (1998) nicht aktiv ist, sondern nur die Informationen zur Verfügung stellt. Die Parameter der Aufgaben variieren je nach ihrer Art. Es wird einerseits unterschieden zwischen den Hilfsaufgaben, wie Antransport, und andererseits den Fügevorgängen, wobei bei diesen die Parameter ebenfalls von der Art des Fügevorgangs abhängen.

Für die Anwendung in der Mensch-Roboter-Kooperation werden besondere Anforderungen an die Inhalte des Wissensmodells gestellt. So ist insbesondere die Beschreibung der Parameter für die Mensch-Roboter-Kooperation zu beachten. Hier werden Grenzwerte festgelegt, die nach aktueller Normenlage und unter Gewährleistung der Sicherheit bei der Programmerstellung nicht überschritten werden dürfen. Durch den Zugriff der Benutzerschnittstelle auf das Wissensmodell wird die Eingabe von Parametern außerhalb des Grenzbereichs bereits verhindert. Außerdem hat dies Einfluss auf den Fähigkeitsabgleich, der durch das Planungsmodul ausgeführt wird. Auf der Grundlage der Anforderungen sollen die Bewegungen des Roboters nicht frei wählbar, sondern möglichst vordefiniert sein.

Dies dient einerseits dazu, dass sich der Mensch auf die Bewegungen einstellen kann, und andererseits der Sicherheit der Applikation. Daher werden im Wissensmodell die Bahnen des Roboters zu den jeweiligen Aufgabenblöcken hinterlegt. Im Wissensmodell werden die in Abschnitt 5.4 identifizierten Montagevorgänge abgebildet. Die Beschreibungen zu den Aufgaben im Wissensmodell erfolgen bezogen auf die zur Verfügung stehenden Ressourcen. Für die Roboter bedeutet dies, dass die Beschreibungen je nach Hersteller variieren.

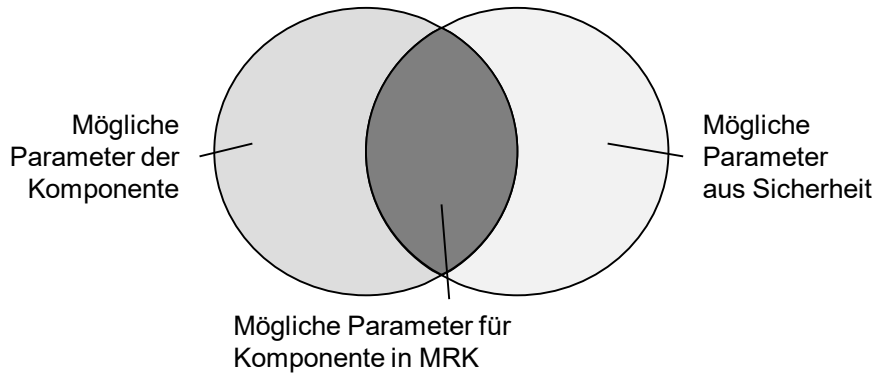
Das Wissensmodell stellt die Informationen zu den identifizierten Aufgaben zur Verfügung. Sollen in das Modell weitere Aufgaben oder Ressourcen integriert werden, so kann das Wissensmodell durch einen Experten erweitert werden. Dazu müssen die neuen Aufgaben in der Bibliothek ergänzt werden oder eine neue Bibliothek für einen anderen Robotertyp erstellt werden. Dies erfolgt nicht über die Benutzerschnittstelle, sondern im Expertenbereich des Programmiersystems.

### **5.5.3.1 Modellierung der Aufgaben im Wissensmodell**

Um die Aufgaben im Wissensmodell zu modellieren, müssen Parameter identifiziert werden. Bei der Untersuchung der technologischen Abläufe nach Ähnlichkeiten stellt WIELAND (1995) Parallelitäten zwischen den Aufgaben fest. Weiterhin beschreibt er für die unterschiedlichen Varianten des Fügens durch Zusammensetzen deren Abläufe und identifiziert Parameter und Endkriterien für verschiedene Fügearten, wie die positions- oder endkraftgesteuerte Montage.

Die Parameter zu den jeweiligen Fügevorgangsarten, die für die Durchführung des Vorgangs relevant sind, müssen im Wissensmodell abgebildet werden. WIELAND (1995) beschreibt in seiner Arbeit Parameter für unterschiedliche Arten der Fügebewegung. So nennt er bei einem positionsgesteuerten Fügevorgang als Parameter die Endposition und Geschwindigkeit; bei einer kraftverlaufsgeführten Fügebewegung sind es der Bewegungsvektor, die maximale Geschwindigkeit und der Kraftverlauf. Die Parameter der Endposition und der Geschwindigkeit werden zur Parametrierung der Roboterbewegung verwendet. Die weiteren Parameter zur Beschreibung der Aufgaben werden durch die Analyse der Aufgaben und die notwendigen Angaben zur Roboterprogrammierung ermittelt. So werden beispielsweise bei der Aufgabe *Antransport* nicht nur die Endposition, sondern auch die Position der Teilebereitstellung benötigt. Die Parameter zur Beschreibung der Aufgaben sind in Abbildung 20, Abbildung 21 sowie im Anhang A1 dargestellt. Hinsichtlich des Einsatzes in der MRK werden diese Parameter in einigen Fällen mit Grenzen belegt, die nicht überschritten werden dürfen, um den Menschen nicht zu

gefährden. Dies wird nachfolgend in Abschnitt 6.4 näher beschrieben. Für die Parameter wird für die jeweilige Ressource und unter Betrachtung der Sicherheit ein Bereich angegeben, in dem Werte angenommen werden können. Dabei wird die Schnittmenge aus der Wertemenge der Ressource und der Wertemenge der Sicherheit gebildet, vgl. Abbildung 19.



*Abbildung 19: Schema zur Festlegung der Parameterbereiche für die Mensch-Roboter-Kooperation*

Die abgebildeten Fügeprozesse mit den jeweiligen Parametern sind im Anhang A1 ausführlich dargestellt. Als Beispiele werden nachfolgend der Fügevorgang des Schraubens und der Antransport betrachtet.

### 5.5.3.2 Beispiele für Modellierung

Ein Schraubvorgang besteht jeweils aus der Bewegung zur Schraubposition und dem Schraubvorgang selbst. Dazu muss am Roboterflansch eine Schraubspindel bzw. Schraubeinrichtung zur Verfügung stehen. Bevorzugt sind dabei solche, die über eine interne Schraubenversorgung verfügen (GROßBERNDT 1988). Abhängig von der Anzahl der Schrauben, erfolgt nach einer Verschraubung eine Bewegung zur nächsten Schraubposition. Aus GROßBERNDT (1988) gehen als Leistungsdaten für die Schraubeinrichtung das Anziehmoment, das Anziehverfahren und die Anzahl und Art der Schrauben als Parameter für einen automatisierten Schraubvorgang hervor. Im Zusammenhang mit der Erstellung des Roboterprogramms wird darüber hinaus die jeweilige Schraubposition ergänzt (vgl. Abbildung 20). Im Roboterprogramm sind die Bahnen zum Anfahren der einzelnen Schraubpositionen enthalten. Die Ansteuerung des Schraubers erfolgt ebenfalls über die im Roboterprogramm enthaltene Ansteuerung des Schraubsystems. Aus der Literatur zur Konzeption einer Mensch-Roboter-Applikation können sicherheitsrelevante Parameter abgeleitet werden. Über die zulässigen Kraft- und Druckwerte aus der ISO TS 15066 kann eine



reduzierte Geschwindigkeit zur Anfahrt der Schraubposition abgeleitet werden. Nach Möglichkeit sollte der Hub zur Bewegung zu einer weiteren Schraubposition kleiner 4 mm oder größer 25 mm betragen, da damit das Risiko des Einquetschens eines Fingers verringert werden kann (BERUFGENOSSENSCHAFT HOLZ UND METALL 2011).

Ablauf	Schematischer Ablauf	Parameter	Sicherheitsrelevante Parameter
Bewegen Schraubposition ↓ Schraubvorgang ↓ Bewegen zur nächsten Schraubposition	<p>Werkzeug</p> <p>Schraubposition A</p> <p>Schraubposition B</p>	Schraubenposition (x/y/z) Drehwinkel Drehmoment (Nm) Anzahl der Schrauben Art der Schraube	Definierte Bewegung; reduzierte Geschwindigkeit bei Anfahrt Schraubposition Hub ab Bauteilposition < 4 mm oder >25 mm

\*abhängig von Greifer

Abbildung 20: Beschreibung der Funktion Schrauben

Neben den Fügevorgängen müssen auch die Handhabungsaufgaben beschrieben werden. Die Aufgabe *Antransport*, die die Grundbewegungen Hinlangen, Greifen und Bringen nach MTM zusammenfasst, wird über die Elementarvorgänge definiert und die jeweiligen Vorgänge werden mit Parametern versehen. Angelehnt an die Darstellung eines prinzipiellen Montageablaufs von WIELAND (1995) wird zur Definition der Roboterbewegung der Vorgang Hinlangen in eine horizontale und vertikale Bewegung unterteilt. Dies entspricht auch den Hinweisen, dass die Anfahrbewegung des Roboters möglichst rechtwinklig zum Bauteil sein sollte (SIHN ET AL. 2017). Als Parameter werden hier die Bewegungsgeschwindigkeit und die Beschleunigung definiert. Der Greifvorgang beschreibt das Schließen des Greifers über die Greifkraft; optionaler Parameter ist hierbei noch die Greifdistanz. Abbildung 21 zeigt die Zusammenfassung der Aufgabe *Antransport*.

Ablauf	Schematischer Ablauf	Parameter	Sicherheitsrelevante Parameter
<p>Bewegen zu Bauteilposition (Hinlangen)</p> <p>↓</p> <p>Bauteil greifen (Greifen)</p> <p>↓</p> <p>Bewegen zu Montageposition (Bringen)</p> <p>↓</p> <p>Greifer öffnen (Loslassen)</p> <p>↓</p> <p>Fügevorgang durchführen (Fügen)</p>		<p>Bauteilposition (x/y/z)</p> <p>Greifkraft (N)* Greifweg (mm)*</p> <p>Montageposition (x/y/z)</p> <p>Abhängig von Fügevorgang</p>	<p>Definierte Bewegung; reduzierte Geschwindigkeit zwischen A und B</p> <p>Greifkraft &lt; 140N</p> <p>Definierte Bewegung; reduzierte Geschwindigkeit zwischen C und D</p>

\*abhängig von Greifer

Abbildung 21: Beschreibung der Funktion Antransport

## 5.6 Nutzungsschicht

Im Kapitel 4 wurden Anforderungen an die Benutzerschnittstelle für das aufgabenorientierte Programmiersystem identifiziert. Wie aus den Anforderungen erkennbar wird, soll die nutzende Person über die Benutzerschnittstelle mit dem System interagieren und das Programmiersystem steuern können. Außerdem stellt die Benutzerschnittstelle die Verbindung zwischen Planungs- und Programmiermodul dar, wobei die Daten aus dem Planungsmodul auf der Benutzerschnittstelle angereichert und im Programmiermodul weiterverarbeitet werden. Die Funktionen der Benutzerschnittstelle umfassen das Starten eines Planungs- und Programmierprozesses basierend auf der CAD-Datei, die Anpassung der Aufgabenzuteilung und Parametrierung der Aufgaben (vgl. Abbildung 22), die Anzeige des simulierten Montageablaufs und die Visualisierung des aktuellen Programmdurchlaufs.

Die Hauptfunktion der Benutzerschnittstelle liegt in der Anzeige und Änderung der Aufgaben von Roboter und Mensch. Um die Entscheidung der nutzenden Person, welche Ressource eine Aufgabe übernehmen soll, zu unterstützen und außerdem den Aufwand zur Eingabe der einzelnen Aufgaben zu verringern, wird durch das vorgelagerte Planungsmodul (vgl. Abschnitt 6.1) ein Vorschlag für die Aufgabenzuteilung erarbeitet. Dieser wird dann auf der Benutzerschnittstelle angezeigt.

Die Verschiebung der Aufgabenblöcke erfolgt durch Drag & Drop-Technik. Diese bietet den Vorteil, dass Aufgaben während der visuellen Darstellung auch zwischen den Ressourcen verschoben werden können. Eine schematische Darstellung des Aufgabenbereichs auf der Benutzerschnittstelle ist in Abbildung 22 zu sehen.

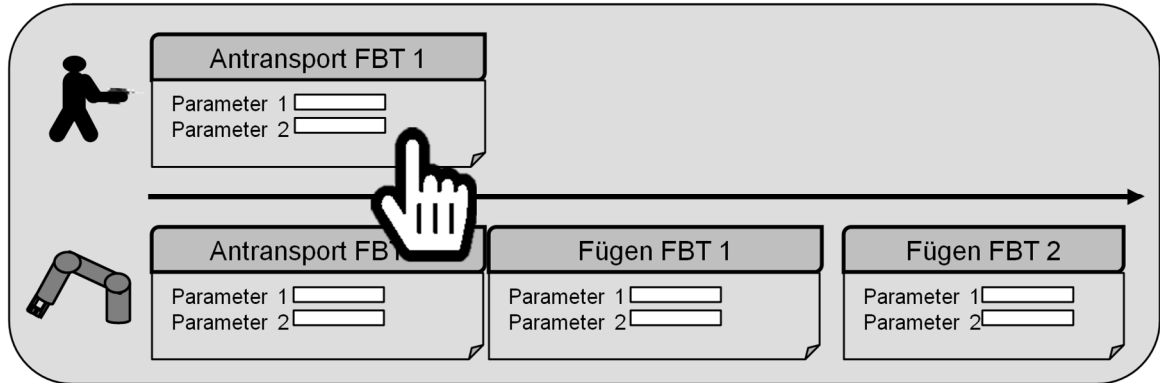


Abbildung 22: Schematische Darstellung der Aufgabenanzeige der Benutzerschnittstelle

Neben der Darstellung und Verschiebung der Aufgabenblöcke muss die Benutzerschnittstelle die Funktion der Parametrierung der Aufgaben bieten. Da es nicht auszuschließen ist, dass zusätzlich zu den Aufgaben, die über die automatisierte Ableitung der Montagereihenfolge identifiziert wurden, Aufgaben notwendig sind, gibt es darüber hinaus die Möglichkeit, neue Aufgaben hinzuzufügen, einer Ressource zuzuordnen und zu parametrieren.



## 6 Hauptmodule des Programmiersystems

### 6.1 Planungsmodul

Das Planungsmodul generiert einen Vorschlag für die Aufgabenzuteilung zu Mensch und Roboter, der dem Nutzer über die Benutzerschnittstelle dargestellt wird. Für die Erarbeitung des Vorschlags stehen drei Untermodule zur Verfügung, die als Montageplanung, Fähigkeitenabgleich und Szenarioplanung bezeichnet werden.

#### 6.1.1 Ableitung des Montageplans und Beschreibung der Aufgaben

Zunächst werden, basierend auf dem Aufgabenmodell, der Montageplan abgeleitet und die einzelnen Aufgaben modelliert (vgl. Abbildung 23), wie in Abschnitt 5.3 beschrieben. Im ersten Schritt wird der Montageplan des Produkts oder der Baugruppe abgeleitet und die Aufgaben werden mit entsprechenden Daten angereichert. Der Montageplan repräsentiert die Reihenfolge der durchzuführenden Aufgaben bis zum fertig montierten Produkt. Da die Informationen aus der Ableitung für den weiteren Verlauf notwendig sind, sollte auch bei vorhandenem Montagevorranggraphen eine Ableitung durchgeführt werden. Die Ableitung der Montagereihenfolge erfolgt anhand des CAD Produkts und ist angelehnt an die Arbeit von MICHNIEWICZ (2019). Dazu wird zunächst ein Bauteil ausgewählt, das im nächsten Schritt auf globale Demontierbarkeit hin überprüft wird. Dieses Bauteil wird auch als Demontagekandidat bezeichnet. Die Auswahl des Demontagekandidaten ist angelehnt an MICHNIEWICZ (2019) und GU & YAN (1995), wobei die Abstandsbestimmung der Bauteile jedoch anhand der Positionen der Koordinatensysteme erfolgt. Dazu werden die Positionen aller Koordinatensysteme der Bauteile ausgelesen und das Bauteil mit dem größten Abstand zum Koordinatensystem des Basisbauteils, das durch den Nutzer bestimmt wird, wird als Demontagekandidat ausgewählt und auf Demontierbarkeit hin überprüft. Für diese Überprüfung stehen zwei Alternativen zur Verfügung: Der erste Ansatz beschreibt eine Variante des Hüllkörperansatzes, basierend auf ENG ET AL. (1999). Die zweite Möglichkeit beschreibt ein schrittweises Verschieben des Montagekandidaten, angelehnt an OU & XU (2013).

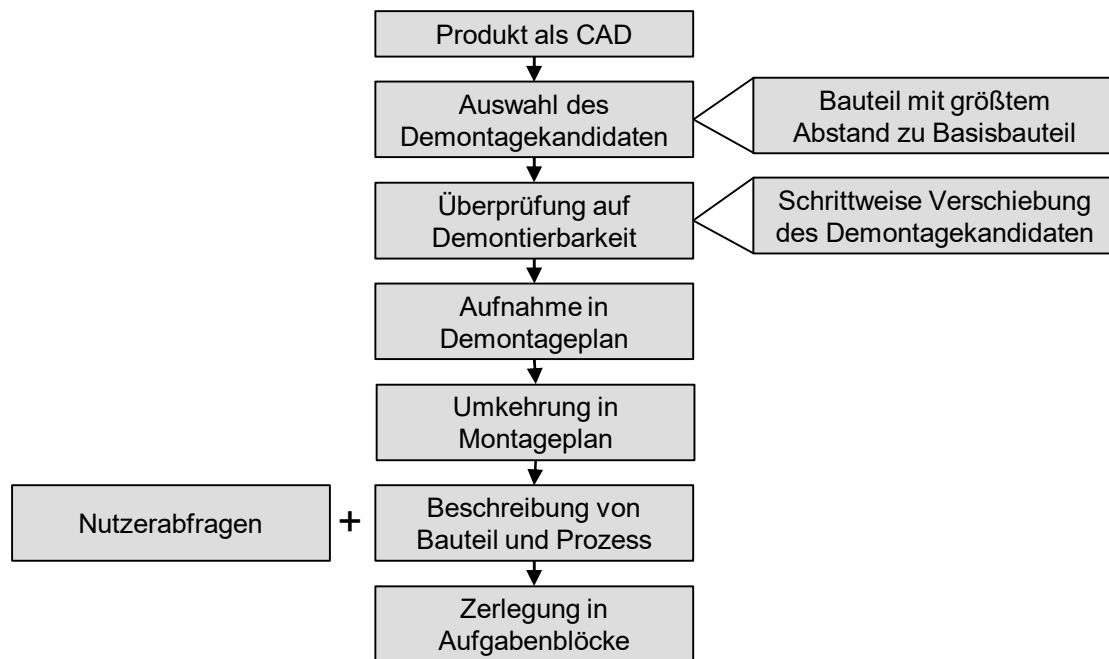


Abbildung 23: Ablauf zur Beschreibung der Aufgabe

Ist der Schritt der Demontageüberprüfung erfolgreich, so wird dieser Montageschritt in den Plan aufgenommen und das entsprechende Bauteil und der zugehörige Montageprozess werden näher beschrieben. Um die Informationen zu identifizieren, werden entsprechende Informationen aus dem CAD ausgelesen. Informationen, die nicht ausgelesen werden können, werden durch Nutzerabfragen in das System aufgenommen, um die Beschreibung zu vervollständigen. Bei der Ableitung der Montagereihenfolge ausgehend vom Produkt in einer CAD-Umgebung werden die Aufgaben auf Ebene der Montageschritte nach Abschnitt 5.3 beschrieben. Wie jedoch erläutert, werden die Aufgaben auf der Ebene eines Vorgangs weiterverwendet, um sie weiter zerlegt zu Mensch und Roboter aufteilen und zuordnen zu können. Dazu werden die einzelnen Montageschritte nach Ableitung der Reihenfolge in Aufgaben zerlegt und mit den Eigenschaften beschrieben (vgl. Abbildung 24).

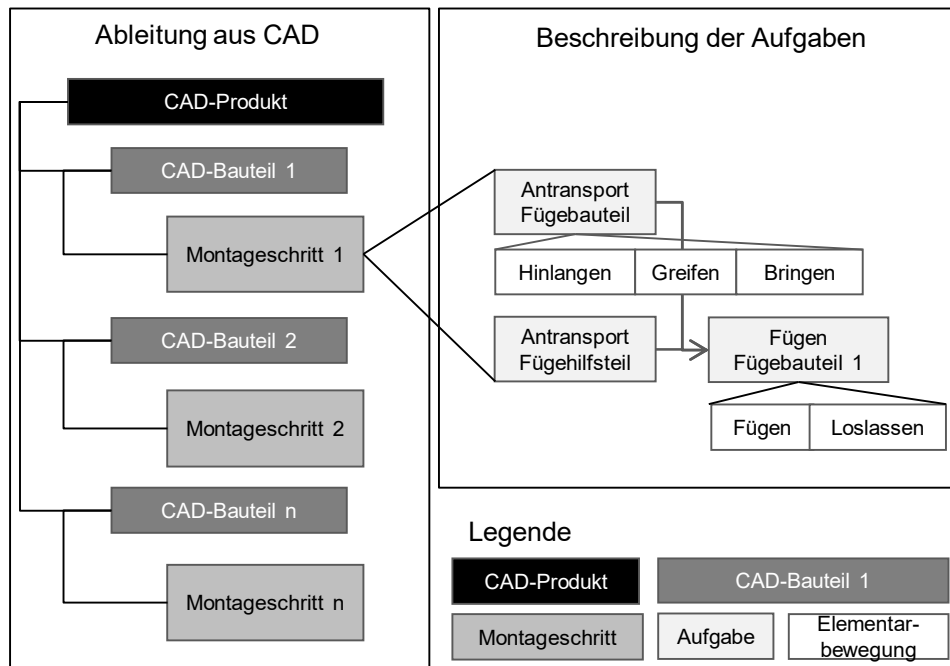


Abbildung 24: Ableitung der Montageschritte und Zerlegung in Aufgaben

Mit den Bauteil- und Prozessinformationen wird bewertet, ob sich die Aufgabe für eine Mensch-Roboter-Kooperation eignet und welche Ressource die Aufgabe besser durchführen kann. In seiner Arbeit zur Bewertung des Aufwands zur Automatisierbarkeit stellt ROSS (2002) Kriterien zusammen, über die der Aufwand zur Automatisierung bestimmt wird. Jedem Kriterium werden vier Ausprägungen zugeteilt, bei denen die erste Ausprägung den geringsten und die vierte Ausprägung den höchsten Aufwand zur Automatisierung darstellt. Eine weitere Methode zur Beurteilung der Automatisierbarkeit beschreibt DEUTSCHLÄNDER (1989). Er unterteilt die Kriterien in objekt- und prozessbezogene, wobei jedes Kriterium drei Ausprägungsformen hat. Aus diesen beiden Arbeiten können daher bereits einige relevante Kriterien zur Prüfung der Automatisierbarkeit abgeleitet werden, mit denen das Produkt und der Prozess beschrieben werden können. Auf Basis der Beschreibung erfolgt dann die Bewertung der Ressourcen hinsichtlich ihrer Eignung. Aus der automatisierten Ableitung können folgende Informationen gewonnen und verwendet werden:

- Abmessungen, Volumen und Masse
- Bauteil- und Prozessname
- Bauteilmaterial, Oberflächenempfindlichkeit
- Formstabilität
- Anziehungsmomente für den Schraubvorgang
- Montagerichtung des Bauteils

Die Bewertung der Aufgaben hinsichtlich des Einsatzes in einer Mensch-Roboter-Kooperation erfordert Kriterien einerseits zur Bewertung der Automatisierung und andererseits zur Überprüfung eines Einsatzes in einer Mensch-Roboter-Kooperation. Dabei handelt es sich insbesondere um Eigenschaften des Produkts, die die Sicherheit betreffen. Daher wird das Kriterium *Scharfkantigkeit des Bauteils* bzw. das *Vorhandensein spitzer Teile* als relevant identifiziert und zur Produktbeschreibung hinzugefügt. Auf Grundlage der ISO TS 15066 beeinflusst dieses Kriterium die Ausführbarkeit der Aufgabe durch den Roboter. Die zulässigen Grenzwerte des Drucks im Falle einer Kollision werden bei einer kleinen Fläche bereits bei geringen Geschwindigkeiten schnell überschritten, was zu einer geringen Bewegungsgeschwindigkeit des Roboters und damit zu langen Ausführungszeiten der Aufgabe führt. Als Kriterien, die den Einsatz in einer MRK beeinflussen, werden folgende hinzugefügt:

- Scharfkantigkeit des Bauteils
- Vorhandensein spitzer Teile
- Art der Montage: Fügevorgang erfordert hohe Kraft
- Feinfühligkeit für den Fügeprozess

### 6.1.2 Abgleich der Fähigkeiten

Zur Prüfung, welche der beiden Ressourcen eine Aufgabe übernehmen sollte, müssen die im vorhergehenden Abschnitt beschriebenen Aufgaben mit den Fähigkeiten der Ressourcen abgeglichen werden. In Abschnitt 3.2.3 wurde der Ansatz von HAMMERSTINGL & REINHART (2017) zum Abgleich der Aufgaben mit den Fähigkeiten der Ressourcen beschrieben. Dieses Modell muss um die Ressource des Menschen und Kriterien für die Mensch-Roboter-Kooperation erweitert und für die Integration in das aufgabenorientierte Programmiersystem angepasst werden.

Bei den spezifizierten Aufgabenblöcken *Antransport* sowie den meisten Fügevorgängen, wie sie in dem Wissensmodell beschrieben sind, handelt es sich um zusammengesetzte Fähigkeiten, da sie eine Kombination elementarer Fähigkeiten darstellen. Die zusammengesetzte Fähigkeit *Antransport* kann im Falle eines Roboters nur in Kombination mit einem vorhandenen Greifer ausgeführt werden. Diese Information wird in dem Fähigkeitenmodell zur Verfügung gestellt. Analog zu den Fähigkeiten der Automatisierungsgeräte können die Fähigkeiten des Menschen beschrieben werden.



Die zusammengesetzte Fähigkeit *Antransport* kann sowohl für den Menschen als auch den Roboter bzw. die Kombination von Roboter und Greifer mit Parametern beschrieben werden. Wie bereits in der Aufgabenmodellierung beschrieben, setzt sich der Antransport aus den Grundbewegungen Hinlangen, Greifen und Bringen zusammen. Basierend auf HAMMERSTINGL & REINHART (2017) werden für die Fähigkeiten Input sowie Output Parameter definiert, die in die Kategorien Bezeichner, Datentyp, Wert oder Wertebereich, Einheit und Notwendigkeit eingeteilt werden.

Im Hinblick auf die zeitliche Zuordnung der Aufgaben zu Mensch und Roboter, für die mehrere Möglichkeiten bestehen, soll als Ergebnis des Abgleichs der Fähigkeiten eine Eignung der jeweiligen Ressource für eine Aufgabe festgestellt werden, so dass eine bessere bzw. schlechtere Eignung einer Ressource erkennbar wird. Daher wird die Beschreibung der Fähigkeiten um das Kriterium Erfüllungsgrad erweitert. Der Erfüllungsgrad gibt einen Wertebereich an, der von dieser Ressource für dieses Kriterium erreicht werden kann. Zur Ermittlung des Eignungsgrads in Abhängigkeit von einem Kriterium wurden sowohl die Checkliste von DEUTSCHLÄNDER (1989) als auch die Methode nach ROSS (2002) herangezogen. Zur Bewertung werden beide Methoden kombiniert. Die Bewertung erfolgt dann, angelehnt an das Vorgehen von ROSS (2002), mit Hilfe der Ausprägungsstufen von 1-4. Die möglichen Erfüllungsgrade zu den einzelnen Kriterien sind in Anhang A3 zusammengefasst.

Des Weiteren wurden den Fähigkeiten MRK-relevante Parameter, wie die Scharfkantigkeit von Bauteilen oder die Feinfühligkeit, hinzugefügt. Diese wurden auf Basis arbeitswissenschaftlicher Erkenntnisse bzw. ausgehend von den möglichen Gefährdungen von Bauteilen identifiziert und in Abschnitt 6.1.1 beschrieben.

Die zusammengesetzte Fähigkeit *Antransport* kann demnach, wie in folgender Abbildung 25 dargestellt, beschrieben werden.

	Bezeichner	Wert	Einheit	Datentyp	Notwendigkeit	Erfüllungsgrad
	Genauigkeit	±0,08	mm	Int	v	1-4
	Max. Gewicht	10000	g	Int	o	1-4
Input	Feinfühligkeit	Ja	-	bool	o	3-4
	scharfkantige Bauteile	Ja	-	bool	v	2-3
	Durchführungszeit	-	s	Int	o	1-4
Output	Eignung			Int	v	1-4

Abbildung 25: Parameter der zusammengesetzten Fähigkeit *Antransport* in Anlehnung an HAMMERSTINGL & REINHART (2017) (Erweiterungen eingerahmt gekennzeichnet)

Für den Abgleich wird der jeweilige Erfüllungsgrad in Abhängigkeit von den produktseitigen Anforderungen berechnet (vgl. Abbildung 26). Dies erfolgt je nach Kriterium auf unterschiedliche Weise: entweder über den Vergleich von Werten oder einer ja/nein-Beurteilung.

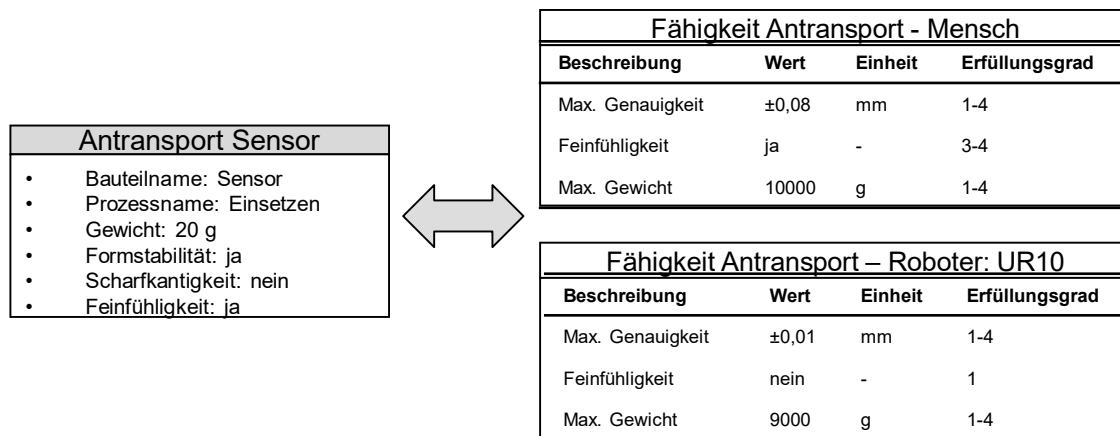


Abbildung 26: Abgleich der Anforderungen der Aufgabe mit den Fähigkeiten der Ressourcen

Der Abgleich erfolgt für die Fähigkeiten der in der Benutzerschnittstelle angegebenen vorhandenen Ressourcen. Aus der Mittelwertbildung der Erfüllungsgrade der einzelnen Kriterien ergibt sich ein Gesamterfüllungsgrad für die Ressourcen je betrachteter Aufgabe. Dieser liegt zwischen eins und vier und dient als Basis für die zeitliche Zuordnung der Aufgaben zu den Ressourcen, die im nachfolgenden Abschnitt unter Szenarioplanung beschrieben ist.

### 6.1.3 Szenarioplanung

Bei der Szenarioplanung können durch die Fokussierung auf bestimmte Kriterien unterschiedliche Szenarien, das heißt unterschiedliche Abläufe und Zuteilungen zu den Ressourcen Mensch und Roboter, resultieren. BEUMELBURG (2005) stellt fest, dass die Unternehmensziele durch Zeit, Kosten und Qualität beschrieben werden können. Bei einer möglichen Zuordnung von Aufgaben und der Planung eines Szenarios können in der Mensch-Roboter-Kooperation jedoch noch weitere Kriterien aus Sicht der Arbeitswissenschaft hinzukommen. So kann als Ziel genannt werden, dass Resttätigkeiten für den Menschen vermieden werden sollen (FRAUNHOFER IAO 2016). Um bei der Szenarioplanung verschiedene Unternehmensziele zu berücksichtigen, werden in Anlehnung an die drei Zielkriterien Kosten, Zeit und Qualität, wie sie auch von THIEMERMANN (2005) beschrieben werden, drei

unterschiedliche Optimierungskriterien identifiziert, die durch den Nutzer ausgewählt werden können.

- Minimale Kosten
- Minimale Durchlaufzeit
- Maximale Automatisierung

Die Szenarioplanung in dieser Arbeit soll automatisiert innerhalb des Planungsmoduls erfolgen und nach den Fähigkeiten bewertete Aufgaben verwenden. Im Stand der Technik wurden von BEUMELBURG (2005), TSAROUCHI ET AL. (2016a) und WANTIA ET AL. (2016) Ansätze zur Ablaufplanung für eine Mensch-Roboter-Kooperation vorgestellt, bei denen die Ablaufplanung an zuvor bewerteten Aufgaben ansetzt. BEUMELBURG (2005) nutzt zur Zuteilung der Aufgaben einen genetischen Algorithmus. TSAROUCHI ET AL. (2016a) berechnen für Abläufe hinsichtlich der Optionen durchschnittliche Auslastung und Durchlaufzeit die Kosten und wählen den Ablauf mit den geringsten Kosten aus. WANTIA ET AL. (2016) nutzen die Planning Domain Definition Language, deren Planer einen Ablauf hinsichtlich der geringsten Kosten generiert. Das Ziel der Szenarioplanung ist es, die Aufgaben basierend auf der Bewertung der Fähigkeiten und mit Bestreben einer parallelen Abarbeitung zuzuteilen. Des Weiteren sollen die Optimierungskriterien verwendet werden können, um dem Nutzer Entscheidungshilfen zu bieten. Im Gegensatz zu den anderen beiden dargestellten Ansätzen kann die Nutzung des genetischen Algorithmus, wie ihn BEUMELBURG (2005) anwendet, dies ermöglichen und durch die iterative Vorgehensweise die beste Option für mehrere Zielkriterien identifizieren.

Daher erfolgt angelehnt an den Ansatz von BEUMELBURG (2005) die Szenarioplanung in dieser Arbeit. Jedoch wird zwischen den drei beschriebenen Optionen unterschieden werden können, die jeweils als Optimierungskriterium in den Algorithmus einfließen. Die Ausgangsbasis für die Anwendung des genetischen Algorithmus sind die mit Erfüllungsgraden versehenen Aufgaben. Damit wird im ersten Schritt eine initiale Population erstellt. Nach SHIFFMAN (2012) liegen typische Werte dafür zwischen 50 und 200. Bei der Erstellung der Individuen der Population wird die fähigkeitsbasierte Bewertung der Aufgaben integriert. Gelten beide Ressourcen für eine Aufgabe geeignet, so wird die Aufgabe zufällig einer Ressource zugeordnet. Gilt eine Ressource als besser geeignet, ist also der Eignungsgrad größer als bei der anderen Ressource, so wird die Aufgabe der besser geeigneten Ressource zugeordnet. Die bestehenden Individuen werden im nachfolgenden Schritt auf ihre Fitness hin bewertet. Diese bestimmt sich in Abhängigkeit von dem ausgewählten Optimierungskriterium. Bei den minimalen Kosten wird davon ausgegangen, dass

sich durch Wartezeiten des Menschen, in denen er keine zusätzliche Aufgabe erledigen kann, zusätzliche Kosten ergeben. Daher ist Ziel dieses Optimierungskriteriums, die Wartezeiten des Menschen zu reduzieren. Dies basiert darauf, dass Kosten dafür berechnet werden, wenn Leerzeiten, also Wartezeiten, durch die Werkskraft entstehen. Ist die Wartezeit jedoch lang genug, so kann angenommen werden, dass die Werkkraft in dieser Zeit einer anderen Tätigkeit nachgehen kann. Der Schwellenwert, ob eine Aufgabe durchgeführt werden kann, wurde auf neun Sekunden festgelegt. Die Berechnung dieses Schwellenwerts basiert auf MTM. Sie setzt sich zusammen aus zwei Stationswechseln (Weg zu einer anderen Station und wieder zurück) und einer durchschnittlichen Fügezeit. Daraus ergibt sich eine Zeitspanne von 9s. Bei der minimalen Durchlaufzeit wird für jedes Individuum die Gesamtzeit zur Durchführung der Aufgaben bestimmt. Diese Zeit soll minimiert werden. Zur Maximierung des Automatisierungsgrades wird der Anteil der durch den Roboter ausgeführten Aufgaben im Verhältnis zur Gesamtzeit bestimmt. Dabei erhalten die Individuen mit einem hohen Automatisierungsgrad eine bessere Fitness.

Bei den ersten beiden Auswahlmöglichkeiten - minimale Kosten bzw. Durchlaufzeit - handelt es sich um ein Minimierungsproblem, wohingegen die dritte Möglichkeit ein Maximierungsproblem beschreibt. Zur einheitlichen Weiterbehandlung werden die beiden ersten Optimierungsmöglichkeiten in ein Maximierungsproblem umgewandelt. Mit Hilfe eines *mating pools* erfolgt die Reproduktion der Individuen. Dafür wird die relative Fitness eines Individuums im Verhältnis zur Summe aller Anteile berechnet. Im *mating pool* sind dementsprechend Individuen mit einer hohen Fitness häufiger vertreten als solche mit einer geringen Fitness. Anschließend wird mit Hilfe des *mating pools* die Reproduktion durchgeführt, indem per Zufall zwei Individuen ausgewählt werden und nach dem Crossover-Prinzip ein neues Kind aus der Kreuzung der Individuen erstellt wird. Beim Crossover werden Teile zweier Genotypen neu zusammengesetzt, wobei die Begriffe Eltern und Kind verwendet werden. Dadurch enthält der neue Genotyp die Eigenschaften der beiden Eltern (MIKKEL T. JENSEN 2001). Durch eine Mutationsrate werden in einigen Fällen Mutationen durchgeführt, um verlorenes genetisches Material wieder zu erhalten oder neu zu gewinnen (MIKKEL T. JENSEN 2001). In dem Fall der Szenarioplanung bedeutet dies eine Änderung der Ressource zur Durchführung einer Aufgabe. Nach der Reproduktion erfolgt ein erneuter Durchlauf des Algorithmus bis zum Erreichen des Abbruchkriteriums.

Das Ergebnis der Szenarioplanung mit der ausgewählten Option wird an die Benutzerschnittstelle weitergegeben, so dass die nutzende Person Änderungen vornehmen kann. Anschließend wird das angepasste Szenario an das

Programmiermodul übergeben. Wird die Anordnung von Aufgaben geändert, so wird überprüft, ob diese neue Anordnung hinsichtlich der Montagereihenfolge zulässig ist. Werden die Ressourcen geändert, so prüft das System, ob beispielsweise die Traglast des Roboters überschritten wird, und meldet dies an den Nutzer.

### **6.2 Programmiermodul**

Im Anschluss an die Planung der Montageaufgaben bei WECK & BRECHER (2006) übernehmen die Greif-, Bahn- und Fügeplaner die Generierung des Roboterprogramms. Analog dazu werden über das Programmiermodul in diesem System die Informationen zu den Aufgaben, inklusive der Zuordnung zu Mensch oder Roboter, in ein lauffähiges Roboterprogramm transformiert und das Modell zur Tätigkeitserkennung konfiguriert. Der Programmcode für den Roboter soll mit Hilfe des Systems offline generiert und dann auf die Robotersteuerung übertragen werden, auf der das Programm im Betrieb abläuft. Das Roboterprogramm kann daher weitestgehend unabhängig von der Konfiguration des Tätigkeitserkennungsmodells für das Betriebsmodul generiert werden. Das Gerüst für das Betriebsmodul stellt das Modell für die Tätigkeitserkennung und die Schnittstelle zum Roboterprogramm zur Weitergabe eines Signals an den Roboter bei erkannter Tätigkeit zur Verfügung. Die Schnittstelle zur Tätigkeitserkennung muss folglich außerdem auf der Robotersteuerung konfiguriert werden. Im Anschluss an die Generierung des Roboterprogramms wird dieses unter Zuhilfenahme einer Simulation auf Funktionsfähigkeit und Kollisionsfreiheit überprüft, die Simulation dem Nutzer auf der Benutzerschnittstelle angezeigt. Erst nach Bestätigung des Nutzers kann der Betrieb erfolgen. Der Ablauf im Programmiermodul ist in der folgenden Abbildung 27 dargestellt.

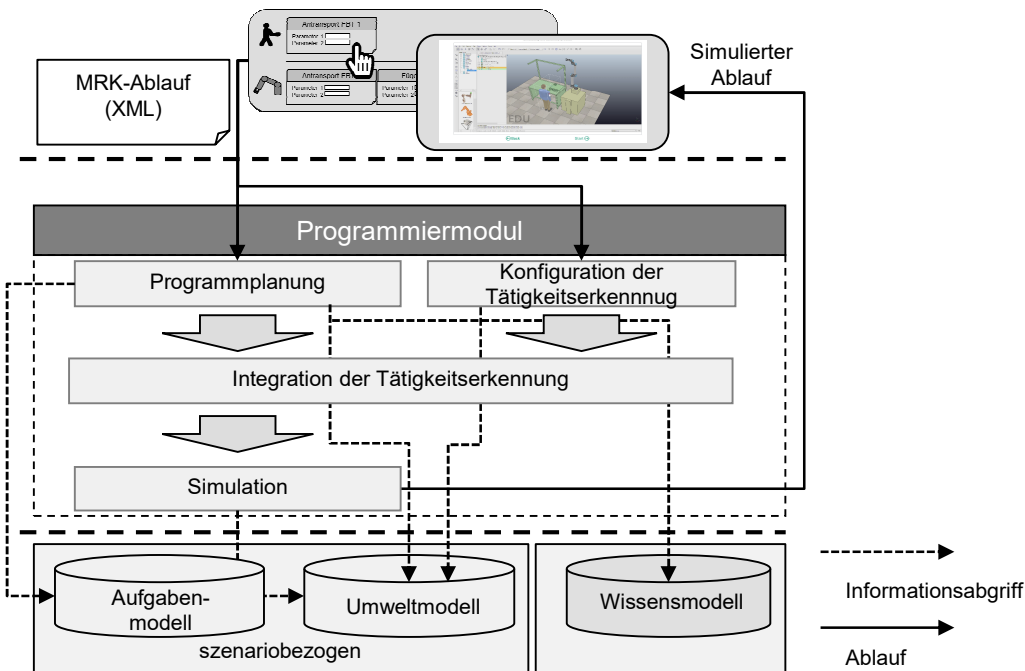


Abbildung 27: Ablauf der Vorgänge im Programmiermodul

Für die Übergabe von Informationen zwischen den einzelnen Submodulen im Programmiermodul muss ein Austauschformat definiert werden. Zum Austausch der Daten stehen die Extended Markup Language (XML) und die JavaScript Object Notation (Json) zur Verfügung. FISCHER-STABEL & GOLLMER (2016) beschreiben XML als eine Methode, um strukturierte Daten in einer Textdatei darzustellen. Durch XML kann im Vergleich zu bspw. HTML eine höhere Flexibilität abgebildet werden, um eine Kommunikation mit unterschiedlichen Anwendungsprogrammen zu ermöglichen. Beispielhafte Anwendungen nach FISCHER-STABEL & GOLLMER (2016) sind der Austausch von Sensordaten in der Meteorologie oder die Beschreibung von Sensoren. ROTTACH & GROß (2002) beschreiben als eine Kerneigenschaft von XML die Eignung zur Maschine-Maschine-Kommunikation. Bei Json handelt es sich um ein Datenaustauschformat, das auf der Programmiersprache JavaScript basiert. Im Vergleich zu XML bietet es den Vorteil, dass die Syntax einfacher ist und es daher leichter zu erstellen ist. Als reines Datenaustauschformat hat es gegenüber XML den Nachteil, dass XML als Markup Language flexibler verwendet werden kann (DIN SPEC 91329:2016-01). Da das Format sowohl zum Austausch von Daten als auch zur Speicherung von Informationen im Wissensmodell dient, wird das flexiblere Format XML gewählt.

So geht aus der Benutzerschnittstelle eine XML-Datei mit den parametrisierten Aufgaben, bereits zu Mensch und Roboter zugeteilt, hervor, die im

Programmiermodul verarbeitet wird. Die Funktionen der einzelnen Submodule sind in den folgenden Abschnitten erläutert.

### **6.2.1 Programmplanung**

Zur Generierung eines herstellereinspezifischen Roboterprogramms müssen dem Programmiermodul entsprechende Informationen über die hinterlegten Bewegungsbefehle mit den zugehörigen Parametern zur Verfügung gestellt werden. Dies erfolgt über das Wissensmodell. Mit deren Hilfe und den in der Oberfläche angegebenen Informationen kann anschließend das Roboterprogramm durch den Postprozessor erstellt werden. Ansätze zur Generierung von Roboterprogrammen über einen Postprozessor wurden von HUMBURGER (1998) und darauf aufbauend von EHRMANN (2007) und BACKHAUS (2016) oder im Falle der automatisierten Programmierung von THOMAS (2008) beschrieben. In den Arbeiten von EHRMANN (2007) und daran anschließend BACKHAUS (2016) war der Postprozessor auf Robotersteuerungen sowie SPS ausgelegt. Während bei diesen Arbeiten über die aufgabenorientierte Programmierung nur ein Ablauf für den Roboter angegeben ist, sind im betrachteten Falle der Mensch-Roboter-Kooperation ein Ablauf für den Menschen und ein Ablauf für den Roboter vorhanden, die zusammen einen Gesamtablauf darstellen. Daraus ergibt sich der Unterschied, dass kein sequentieller Ablauf vorhanden ist, wie er bei BACKHAUS (2016) als Basis für den Postprozessor gilt. Daher muss der Postprozessor aus den vorangegangenen Arbeiten entsprechend modifiziert werden. Auch wird in diesem Ansatz keine SPS-Steuerung integriert, sondern es muss die Signalübergabe zwischen der Robotersteuerung und der Tätigkeitserkennung konfiguriert werden.

Ähnlich der Übergabe der Aufgaben durch das Planungsmodul an die Benutzerschnittstelle zur Anzeige der Aufgaben, gibt die Benutzerschnittstelle die durch den Nutzer parametrisierten Aufgaben über eine XML-Datei an das Programmiermodul weiter. Diese Datei enthält sowohl die Informationen zu den Aufgaben des Roboters als auch zu den Aufgaben des Menschen. Zur Generierung des Roboterprogramms werden die Informationen zu den Aufgaben des Roboters extrahiert und verarbeitet. Die Aufgaben des Menschen werden zur Konfiguration des Modells zur Tätigkeitserkennung und zur Abbildung der Schnittstellen zwischen den Aufgaben auf Seite der Tätigkeitserkennung und der Robotersteuerung verwendet. Zusätzlich notwendige Informationen zur Generierung des Roboterprogramms stellt das Wissensmodell zur Verfügung. Darin enthalten sind

herstellerspezifische Bewegungsbefehle des Roboters und sonstiger Komponenten. Ein Auszug aus dem Wissensmodell ist in Anhang A2 dargestellt.

Auf der Benutzerschnittstelle werden die Aufgaben zu Mensch und Roboter zugeordnet. Durch den Montagevorranggraphen ergeben sich Zwangsbedingungen, die bedeuten, dass gewisse Aufgaben nicht vor Beendigung einer vorherigen Aufgabe begonnen werden können. Diese Abhängigkeiten, die in diesem Fall aus der Aufteilung der Aufgaben zu Mensch und Roboter resultieren, sind auf Seite des Roboterprogramms vergleichbar mit der Anbindung und der Weitergabe von Signalen an eine SPS-Steuerung, wie durch EHRMANN (2007) betrachtet.

Die Abhängigkeiten zwischen den Aufgaben werden bereits in der Datei mit den Abläufen von Mensch und Roboter, basierend auf den Angaben in der Benutzerschnittstelle, beschrieben und können im Postprozessor dann verwendet werden. Auf der Grundlage der Informationen, die mit Hilfe der Benutzerschnittstelle eingegeben werden, und dem Montagevorranggraphen wird das XML erstellt, in dem der Ablauf zwischen Mensch und Roboter beschrieben ist. Nach Fertigstellen der Eingaben der nutzenden Person wird dieses XML generiert. Dabei wird auf das Wissensmodell zugegriffen, um die eingegebenen Parameter in die Struktur zu bringen, so dass sie durch den Postprozessor weiterverarbeitet werden können (vgl. Abbildung 29).

In den Ablauf der Montageaufgabe durch Mensch und Roboter müssen, wie bereits beschrieben, die Abhängigkeiten zwischen den Ressourcen hinzugefügt werden, da diese sowohl im Roboterprogramm als auch für die Tätigkeitserkennung zur Verfügung gestellt werden müssen. Daher müssen bei der Erstellung der XML-Datei mögliche Abhängigkeiten identifiziert werden und zu den jeweiligen Aufgaben hinzugefügt werden. Abbildung 28 zeigt schematisch die Abhängigkeit zwischen zwei Aufgaben.

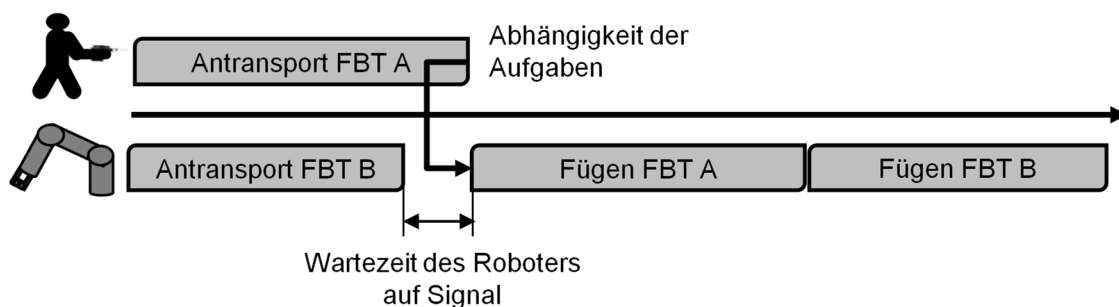


Abbildung 28: Abhängigkeit der Aufgaben



Dies bedeutet, dass zur Ausführung der beispielhaften Aufgabe *Fügen FBT A* durch den Roboter die Aufgabe *Antransport FBT A* durch die Werkkraft abgeschlossen sein muss. Dies muss durch ein entsprechendes Signal an das Robotersystem gemeldet werden. Diese Information, dass ein Signal notwendig ist, wird bei Erstellen der XML-Datei integriert und im Postprozessor (vgl. Abbildung 29) verarbeitet. Für die Tätigkeitserkennung muss diese Information gleichermaßen im XML verankert werden, versehen mit der Information der Schnittstelle zum Robotersystem.

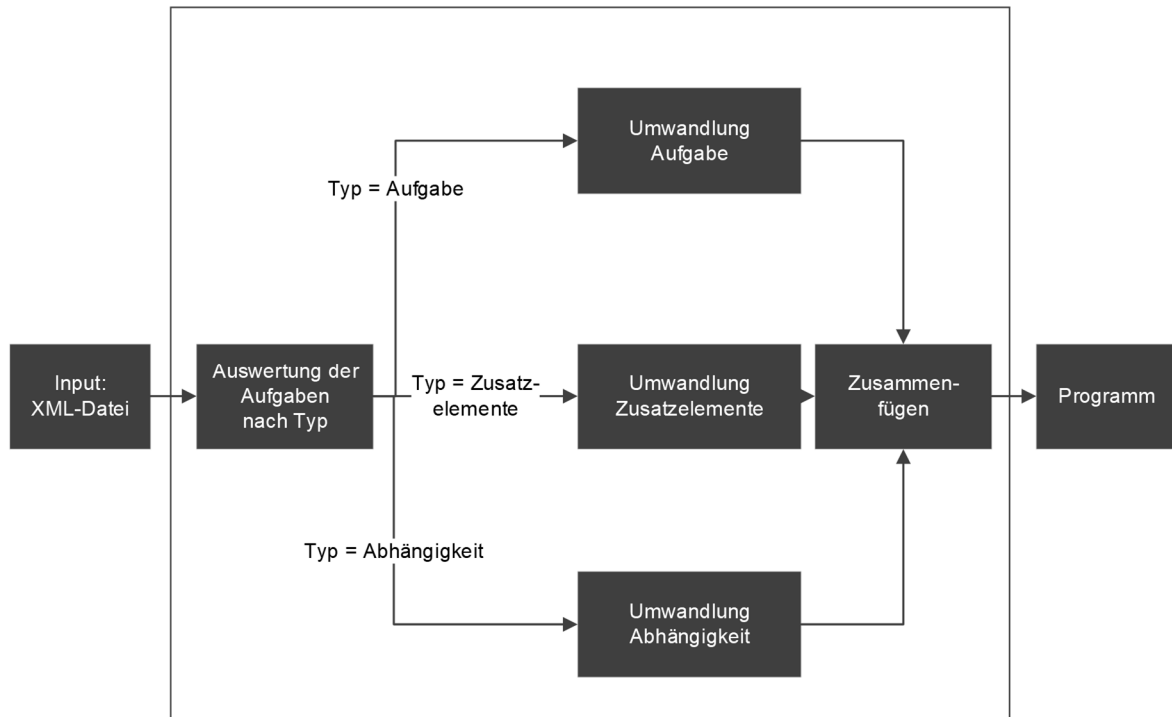


Abbildung 29: Postprozessor zur Programmgenerierung

In der Abbildung 29 sind mehrere Typen ersichtlich. Handelt es sich um eine Aufgabe, die von Roboter oder Mensch durchgeführt wird, so wird hierfür ein entsprechender Bewegungsbefehl für den Roboter integriert. Ein Zusatzelement kann eine zusätzliche Informationsquelle darstellen, die bspw. für die Initialisierung eines Greifers notwendig ist. Über die Abhängigkeiten wird eine Bedingung in das Programm eingefügt, die als Input das Signal der Tätigkeitserkennung erhält. Das Ergebnis ist folglich ein Roboterprogramm mit Greifpunkten und Schnittstellen zur Tätigkeitserkennung.

### 6.2.2 Konfiguration des Tätigkeitserkennungsmodells

In den Anforderungen wurde festgestellt, dass für den Betrieb in einer Mensch-Roboter-Kooperation eine technische Schnittstelle für die Abhängigkeiten zwischen

Mensch und Roboter geschaffen werden muss. Um keinen Taster verwenden zu müssen und demnach der Forderung nach Antizipation des Roboters nachzugehen, soll ein System verwendet werden, mit dem berührungslos die Information über den Stand der Abarbeitung des Menschen detektiert werden kann. Dies kann durch die Erkennung der durchgeführten Tätigkeiten erfolgen. Im Stand der Technik werden Ansätze zur Erkennung von menschlichen Tätigkeiten beschrieben, die meist auf Wahrscheinlichkeitsmodellen basieren. Ansätze im Bereich der Mensch-Roboter-Kooperation werden z. B. von SHEN (2015) oder ROITBERG ET AL. (2014) beschrieben. SHEN (2015) evaluierte Verfahren zur Tätigkeitserkennung. Diese umfassten die Fuzzy-Logik, Neuronale Netze, Markov Ketten, Kalman-Filter, Hidden Markov Modelle und Petri-Netze. In seiner Bewertung erfüllen Hidden Markov Ketten und Petri-Netze die Anforderungen gleichermaßen, wobei die Wahl aufgrund der Ausgeglichenheit zwischen mittlerer Genauigkeit, hoher Geschwindigkeit und guter Trainierbarkeit auf die Hidden Markov Modelle fällt. Auch nach KIM ET AL. (2010) stellen Hidden Markov Modelle für die Erkennung menschlicher Tätigkeiten eine der meist verwendeten Wahrscheinlichkeitsmodelle dar. Daher wird darauf aufbauend in dieser Arbeit ein Hidden Markov Modell verwendet.

Die Voraussetzung für die Erkennung von Tätigkeiten ist insbesondere deren vorherige Definition; im Übrigen müssen sie dem System bekannt sein (SHEN 2015). Im Gegensatz zu der Arbeit von SHEN (2015), der einen Satz von Trainings durchführen muss, um das Modell zu nutzen, geht aus den Anforderungen in Abschnitt 4.6 hervor, dass das Training des Modells möglichst automatisiert erfolgen und minimiert werden soll. Dies setzt voraus, dass ein Training des Modells weitestgehend eliminiert und das Modell stattdessen konfiguriert wird. Wie das Modell konfiguriert wird, stellt nachfolgender Abschnitt dar. Zunächst werden jedoch das verwendete Modell und dessen Eigenschaften erläutert, so dass danach auf die automatische Konfiguration des Modells eingegangen werden kann.

ALZHRANI & KAMMOUN (2016) beschreiben in Anlehnung an BULLING ET AL. (2014) den allgemeinen Aufbau eines Systems zur Erkennung menschlicher Tätigkeiten in fünf Schritten (vgl. Abbildung 30).

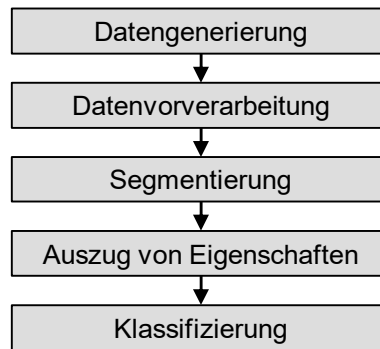


Abbildung 30: Vorgehen zur Erkennung von Tätigkeiten angelehnt an ALZAHIRANI & KAMMOUN (2016)

Im ersten Schritt werden Daten aufgenommen, die anschließend vorverarbeitet werden. Im dritten Schritt werden die für das Modell relevanten Informationen extrahiert. Darauf folgend werden aus den verbleibenden Informationen die Merkmale identifiziert, die für die Erkennung notwendig sind. Bei der Klassifizierung werden dann zunächst im Training die Modelle eingelernt und im Betrieb die erkannten Tätigkeiten klassifiziert. Nach diesem Vorgehen werden, angelehnt an SHEN (2015), für die einzelnen Schritte entsprechende Funktionen konzipiert. Zur Klassifizierung der Tätigkeiten wird ein Hidden Markov Modell verwendet.

### 6.2.2.1 Modell zur Tätigkeitserkennung

Wie im vorherigen Abschnitt beschrieben, soll die Klassifizierung der Tätigkeiten entsprechend der Aufgabenblöcke erfolgen. Um dies zu erreichen, müssen Merkmale der Tätigkeiten definiert werden, die über die Informationen aus den gewonnenen Daten abgeleitet werden können.

Dabei ist festzustellen, dass die Tätigkeiten auf der beschriebenen Ebene zu abstrakt sind für eine maschinelle Erkennung. Daher werden sie in ihre auf MTM basierende Struktur zerlegt. Für die einzelnen Grundbewegungen können Bewegungen der Hände festgestellt werden, die in nachfolgender Tabelle 1 aufgelistet sind.

Tabelle 1: Analyse der Handbewegungen für die Grundbewegungen in Anlehnung an MTM (REFA BUNDESVERBAND E.V. 2002)

Grundbewegung	Bewegung der Hand
<b>Hinlangen</b>	Hand bewegt sich von der Grundposition zur Speichereinheit
<b>Greifen</b>	Hand steht über der Speichereinheit und greift das Bauteil
<b>Bringen</b>	Hand bewegt sich mit dem Bauteil zum Montageplatz
<b>Fügen</b>	Hand steht über dem Montageplatz und fügt das Bauteil
<b>Loslassen</b>	Hand steht über dem Montageplatz und lässt das Bauteil los

Bei den Grundbewegungen Hinlangen und Bringen kann eine merkbare Positionsveränderung festgestellt werden, bei den Grundbewegungen Greifen, Fügen und Loslassen ist hingegen kaum eine Positionsveränderung feststellbar.

Die Positionsveränderungen bei den Bewegungen Hinlangen und Bringen bieten daher Potenzial, maschinell erkannt und unterschieden zu werden.

Unter der Voraussetzung, dass die Positionen der Speichereinrichtungen und die der Montageposition bekannt sind, können Trajektorien für die Grundbewegungen Hinlangen und Bringen berechnet werden. Trajektorien werden beschrieben als eine geradlinige oder gekrümmte Bahnkurve oder als ein Pfad im zwei- oder dreidimensionalen Koordinatensystem (BASHIR ET AL. 2007). Bewegungstrajektorien, insbesondere Handbewegungen, sind grundsätzlich für eine Bewegungserkennung geeignet (RAO ET AL. 2002).

Eine schematische Abbildung von Trajektorien zeigt die nachfolgende Abbildung 31.

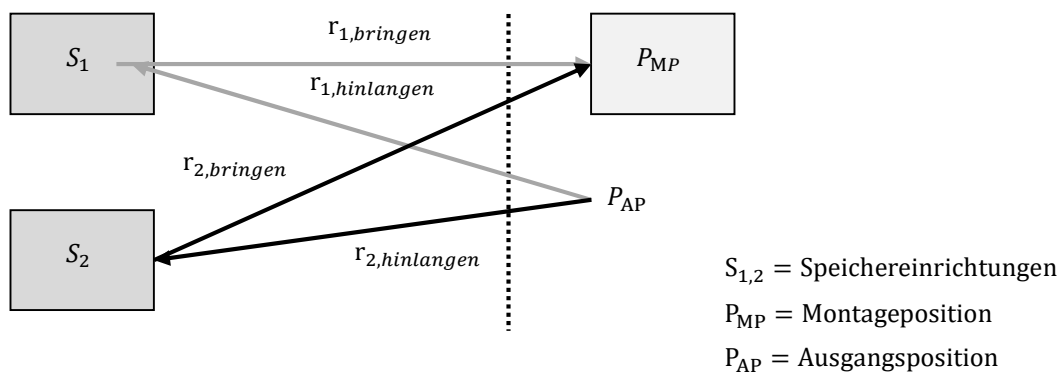


Abbildung 31: Trajektorien zwischen Speichereinrichtungen und Montageposition

Aus Abbildung 31 geht hervor, dass sich die Bewegungen neben der Positionsänderung in ihren Richtungen unterscheiden. Dies ist ein weiteres Merkmal, das für die Erkennung der Grundbewegung verwendet werden kann. Demnach ist festzustellen, dass die Grundbewegungen Hinlangen und Bringen über die Positionsänderung und Bewegungsrichtung entlang der Trajektorie erkannt werden können. Das in BERG ET AL. (2018a) vorgestellte Modell wurde daher um die Bewegungsrichtung der Hand erweitert, wie in folgender Abbildung 32 dargestellt.

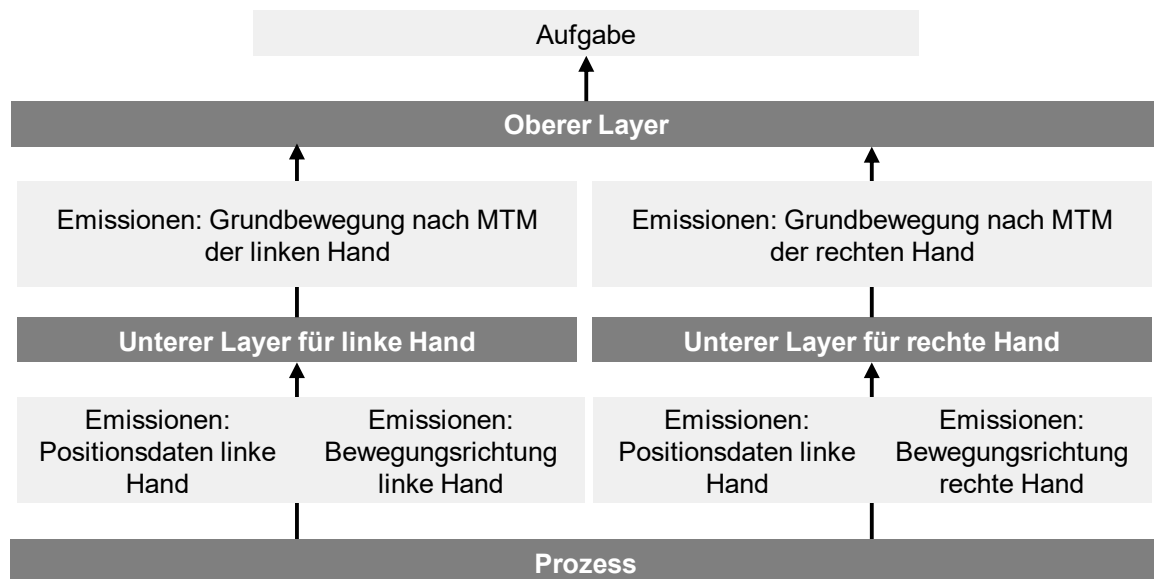


Abbildung 32: Layered Hidden Markov Modell zur Erkennung von Tätigkeiten

Da es sich bei den berechneten Trajektorien um zweidimensionale Trajektorien handelt, die Hand des Werkers sich jedoch im dreidimensionalen Raum bewegt, ist festzustellen, dass die reine Berechnung zwischen den beiden Positionen keine ausreichend gute Trajektorie für die Erkennung ergibt. Auch POLLICK & ISHIMURA (1996) stellen fest, dass Bewegungen der menschlichen Hand nicht geradlinig sind, sondern eine Krümmung aufweisen. Daher wurden die realen Handbewegungen dahingehend untersucht, ein Modell für die Abbildung realer Handbewegungen zu entwickeln, um damit ein genaueres Modell für die Bewegungserkennung liefern zu können.

In Abbildung 33 dargestellt sind die geradlinige Trajektorie zwischen den zwei Punkten und drei Verläufe der realen Handbewegung. Dabei ist die Krümmung der Trajektorie in z-Richtung und damit die Abweichung von der geradlinigen Trajektorie erkennbar. Des Weiteren ist auch eine Unterscheidung der Bewegungstrajektorien untereinander erkennbar.

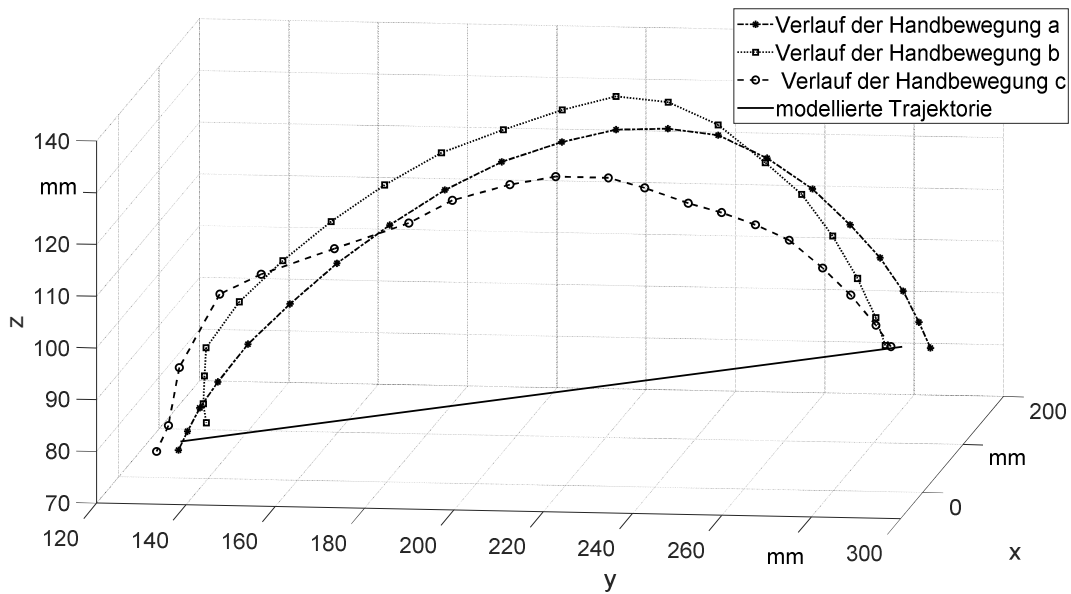


Abbildung 33: Vergleich der modellierten geradlinigen Trajektorie mit den realen Handbewegungen

Um diese Bewegungen realitätsnah abzubilden, werden zwei Schritte vorgenommen:

- 1) Um die Krümmung abzubilden, wird der Mittelpunkt der Grundtrajektorie um einen Wert in Richtung z-Achse verschoben. Dadurch entstehen zwei Teiltrajektorien:
  - a. vom Ausgangspunkt zum erhöhten Mittelpunkt
  - b. vom Mittelpunkt zum Endpunkt

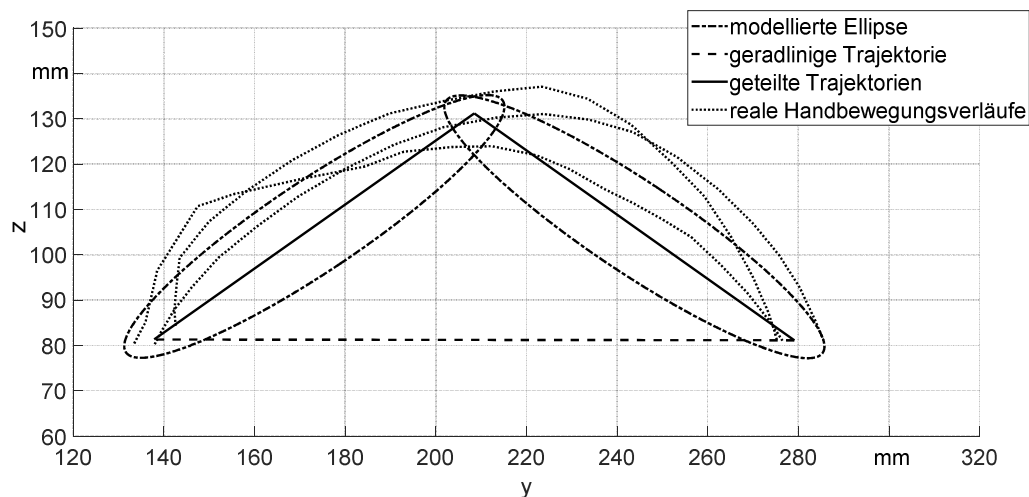
Für die Erhöhung um einen Wert in Richtung der z-Achse wurde in Versuchen festgestellt, dass der Versatz abhängig ist von der Länge der Trajektorie. Die Längen der Trajektorien können dabei in verschiedene Klassen eingeteilt werden. Das Ergebnis stellt nachfolgende Tabelle 2 dar.

Tabelle 2: Erhöhung des Mittelpunkts der Trajektorie abhängig von Länge

Klasse	Distanz	Handerhöhung
1	0 – 100 mm	39 mm
2	100 – 200 mm	42 mm
3	200 – 300 mm	63 mm
4	300 – 400 mm	105 mm

- 2) Zum Ausgleich der Varianz zwischen einzelnen Bewegungsabfolgen wird eine Normalverteilung um beide Teiltrajektorien gelegt, wie es bereits bei der geradlinigen Bewegung vorgenommen wurde (vgl. BERG ET AL. 2018).

Das Ergebnis der Modellierung ist in folgender Abbildung 34 visuell dargestellt.



*Abbildung 34: Modellierung der Trajektorie mit Verschiebung in z-Richtung und Darstellung der Ellipsen*

Diese Modellierung der Bewegungstrajektorie bildet die Basis für die spätere Erkennung der Grundbewegungen Hinlangen und Bringen.

Neben den Trajektorien als Erkennungsmerkmal einer Tätigkeit wird die Bewegungsrichtung betrachtet und in der unteren Ebene als Markov Modell integriert. Bei Untersuchung unterschiedlicher Bewegungsabläufe wurde festgestellt, dass für den größten Teil der Strecke eines Hinlang-Vorgangs bspw. die Bewegungstrajektorie annähernd als linear identifiziert werden kann, so dass eine Bewegungsrichtung erkennbar ist. Bei Annäherung an den Zielpunkt ist keine eindeutige Bewegungsrichtung mehr erkennbar. Daher wird die Bewegungsrichtung nur bei Überschreiten eines Geschwindigkeitsschwellenwerts in die Erkennung der Tätigkeit einbezogen, da dann davon auszugehen ist, dass eine Richtung detektiert werden kann. Zur Berücksichtigung von Abweichungen vom berechneten Sollwinkel wird auch hier eine Normalverteilung verwendet.

### 6.2.2.2 Mathematische Beschreibung von Hidden Markov Modellen

Hidden Markov Modelle (HMM) sind zeit-diskrete Prozesse der Statistik und dienen dazu, nicht direkt erkennbare Zustände, die auch als verdeckt bzw. hidden bezeichnet werden, über Ereignisse zu erfassen. Hidden Markov Modelle basieren auf der Markov-Kette, die wiederum einen Prozess definiert, der zu unterschiedlichen Zeitpunkten verschiedene Zustände annimmt (RABINER 1989).

Über folgende Gleichung ist die Wahrscheinlichkeit des Eintretens einer bestimmten Zustandsfolge mit der Anfangswahrscheinlichkeit  $\pi$  definiert.

$$P(X) = \pi(X_{t_0})P(X_{t_1}|X_{t_0})P(X_{t_2}|X_{t_1}) \dots P(X_{t_T}|X_{t_{T-1}}) \quad (\text{Gl. 6-1})$$

mit:

$P$ : Wahrscheinlichkeit zum Eintritt der Zustandsfolge

$X_t$ : Zustand zum Zeitpunkt  $t$

$\pi$ : Anfangswahrscheinlichkeit

$P$  bildet die Wahrscheinlichkeit des Übergangs von Zustand  $i_n$  in den Nachfolgezustand  $i_{n+1}$  ab. Das System ist dadurch „gedächtnislos“, was bedeutet, dass ein nachfolgender Zustand nur von dem zuvor beobachtbaren Zustand abhängt (RABINER 1989). Die Markov-Kette ist in Abbildung 35 dargestellt.

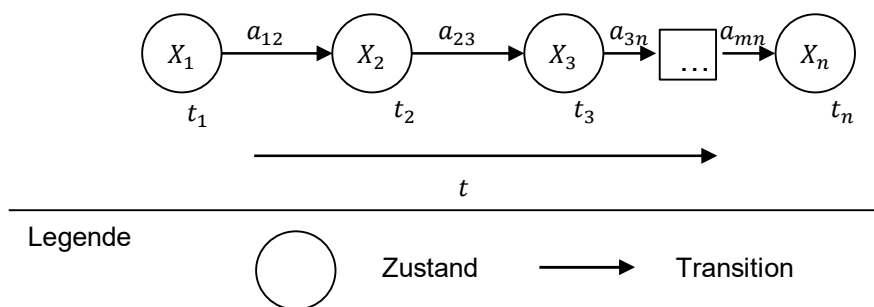


Abbildung 35: Markov-Kette nach DRAKE (1967)

Die Wahrscheinlichkeiten, zum Zeitpunkt  $t$  in den nächsten Zustand überzugehen, werden mit Hilfe der Transitionsmatrix  $A$  beschrieben:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \quad (\text{Gl. 6-2})$$



mit:  $A$ : Transitionsmatrix

$a_{nm}$ : Wahrscheinlichkeit zum Übertritt in nächsten Zustand

Im Gegensatz zum Markov-Modell, können die Zustände in einem Hidden Markov Modell nicht unmittelbar erkannt werden. Von jedem versteckten Zustand werden allerdings Emissionen  $O$  ausgesendet, die beobachtbar sind. Die Wahrscheinlichkeit, dass eine bestimmte Emission zu einem Zustand zu beobachten ist, wird durch die Emissionsmatrix  $B$  beschrieben.

$$B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \quad (\text{Gl. 6-3})$$

mit:  $B$ : Emissionsmatrix

$b_{nm}$ : Wahrscheinlichkeit der Emission  $m$  zum Zustand  $n$

Dabei ist  $b_{nm}$  die Wahrscheinlichkeit, dass die Emission  $m$  zum Zustand  $n$  eintritt (RABINER 1989). Die beiden Schichten mit den Zusammenhängen zwischen den Zuständen und Emissionen sind in der folgenden Abbildung 36 dargestellt.

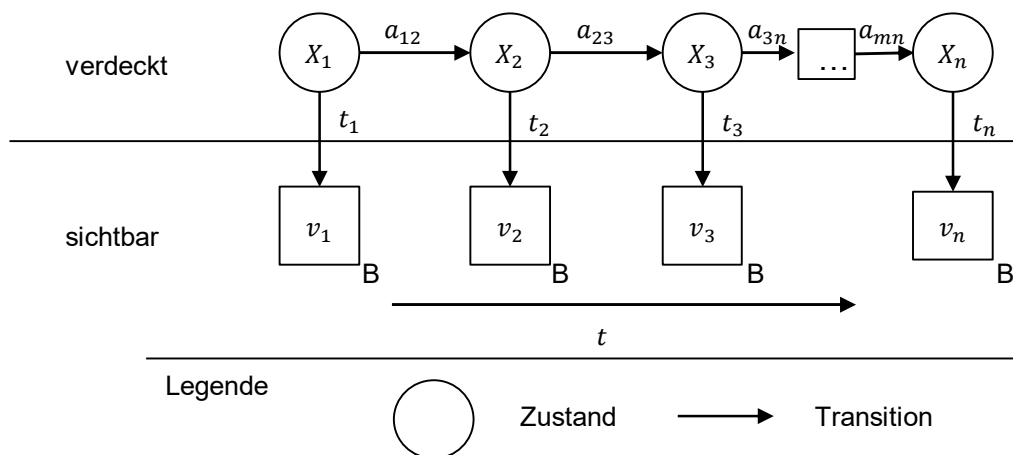


Abbildung 36: Hidden Markov Modell nach RABINER (1989)

Für die Wahrscheinlichkeit einer Emissionssequenz  $O$  mit der Anfangswahrscheinlichkeit  $\pi$ , der Transitionsmatrix  $A$  und der Emissionsmatrix  $B$  bei gegebener Zustandsreihenfolge  $X$  gilt (RABINER 1989):

$$P(\mathbf{X}, \mathbf{O}) = \pi(X_{t_0})P(O_{t_0}|X_{t_0})P(X_{t_1}|X_{t_0})P(O_{t_1}|X_{t_1})\dots P(X_T|X_{T-1})P(O_T|X_T) \quad (\text{Gl. 6-4})$$

mit:

$P$ : Wahrscheinlichkeit zum Eintritt der Zustandsfolge

$X_t$ : Zustand zum Zeitpunkt  $t$

$\pi$ : Anfangswahrscheinlichkeit

$O$ : Emission zum Zeitpunkt  $t$

Im Kontext der Erkennung von menschlichen Tätigkeiten verknüpfen PERDIKIS ET AL. (2008) mehrere HMM auf unterschiedlichen Ebenen zu Layered Hidden Markov Modellen (LHMM). Die Verwendung dieser Modelle ermöglicht eine gesteigerte Genauigkeit und höhere Robustheit des Modells.

Zur Erkennung von Tätigkeiten müssen Positions- und Geschwindigkeitsdaten aufgenommen und im HMM verarbeitet werden. Da hierbei kontinuierliche und nicht diskrete Daten generiert werden, müssen diese zunächst diskretisiert werden, um im HMM verarbeitet werden zu können. Dazu werden Continuous Mixture Hidden Markov Modelle verwendet, bei denen die Emissionswahrscheinlichkeiten nicht über die Emissionsmatrix  $B$ , sondern eine Dichtefunktion  $N$  beschrieben werden. Als Dichtefunktion finden dabei meist multivariante Gaußverteilungen Anwendung (RABINER 1989):

$$N(\mathbf{O}, \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|(2\pi)^d}} e^{-\frac{1}{2}(\mathbf{O}-\mu)^T \Sigma^{-1}(\mathbf{O}-\mu)} \quad (\text{Gl. 6-5})$$

mit:  $\mu$ : Mittelwert

$N$ : Dichtefunktion

$\Sigma$ : Kovarianzmatrix

$d$ : Dimension von  $\Sigma$

Das Training von HMM erfolgt durch Algorithmen, wie von ROITBERG ET AL. (2014) beschrieben.

### 6.2.2.3 Parametrierung des Tätigkeitserkennungsmodells

Als Basis für die Konfiguration des Hidden Markov Modells für die betrachtete Montageaufgabe dient der übergebene Ablauf mit den jeweiligen Aufgaben von Mensch und Roboter. Die Schritte zur Konfiguration des Modells sind nachfolgend

in Abbildung 37 aufgezeigt. Mit Hilfe des Ablaufs der Montageaufgabe, in dem die Aufgaben beschrieben und den Ressourcen zugeordnet sind, können die Transitions- und Emissionsmatrizen ermittelt werden. Des Weiteren werden anhand der Positionsdaten des Arbeitsplatzes (Ausgangsposition der Hände, Position der Teilebereitstellung und der Montageposition) sowohl die Trajektorien als auch die Sollwinkel der Trajektorien abgeleitet.

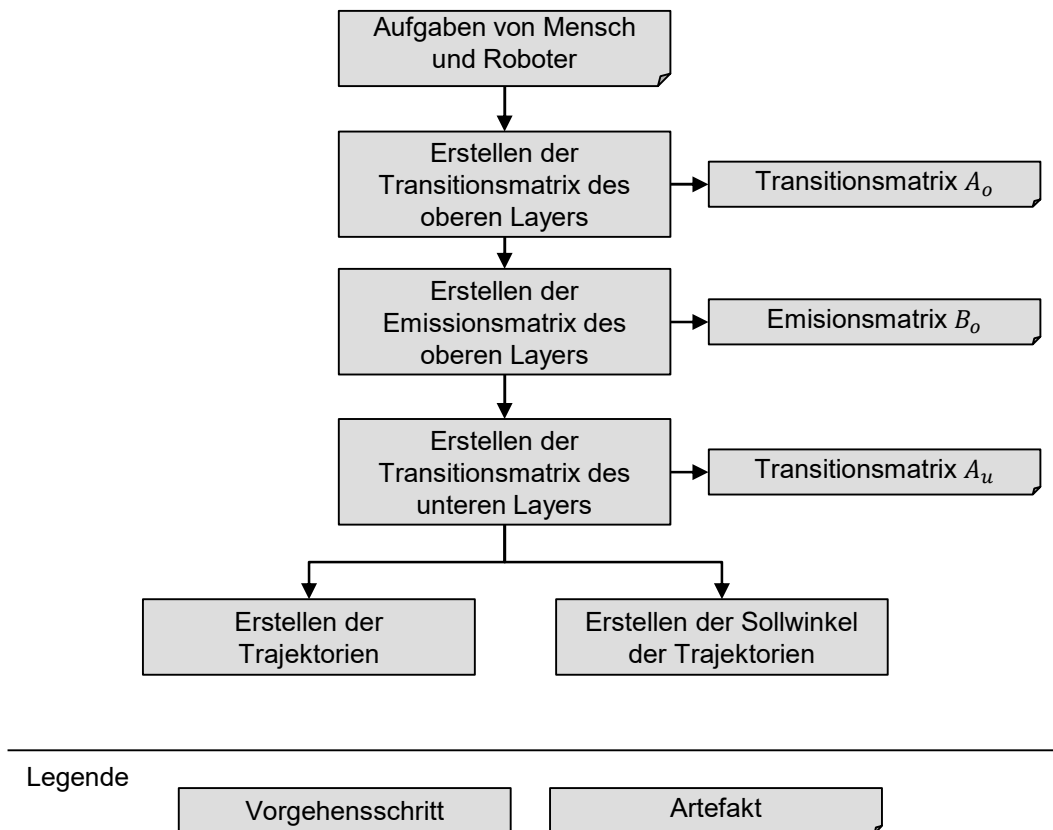


Abbildung 37: Vorgehen zur Konfiguration des Modells zur Tätigkeitserkennung

Der obere Layer des Hidden Markov Modells repräsentiert die Aufgaben. Wie aus der Herleitung hervorgeht, kann die Aufgabe *Antransport* mit Hilfe der Grundbewegungen Hinlangen und Bringen näher beschrieben werden, die, wie zuvor erwähnt, über Positionsänderungen erkannt werden können. Da bei einem Fügevorgang nur wenige Positionsänderungen der Hände festzustellen sind, wird dieser über die Restzeitbestimmung erkannt, wie später vorgestellt wird.

Die Transitionsmatrix  $A_o$  des oberen Layers, wie sie in Abschnitt 6.2.2.2 beschrieben wurde, wird bestimmt aus der Anzahl der Aufgaben des Menschen, die den Zuständen des Modells entsprechen. Hinzu kommt ein Grundzustand, in dem der Mensch keine Aufgabe erledigt und der von jedem anderen Zustand aus erreichbar sein muss. Daher

wird aus dem übergebenen Ablauf die Anzahl der Aufgaben des Menschen bestimmt. Anschließend können die Werte für die einzelnen Einträge ermittelt werden. Abbildung 38 zeigt schematisch, wie die Matrizen basierend auf den Informationen befüllt werden. Die Berechnung der einzelnen Werte der Matrix kann im Anhang A4 eingesehen werden.

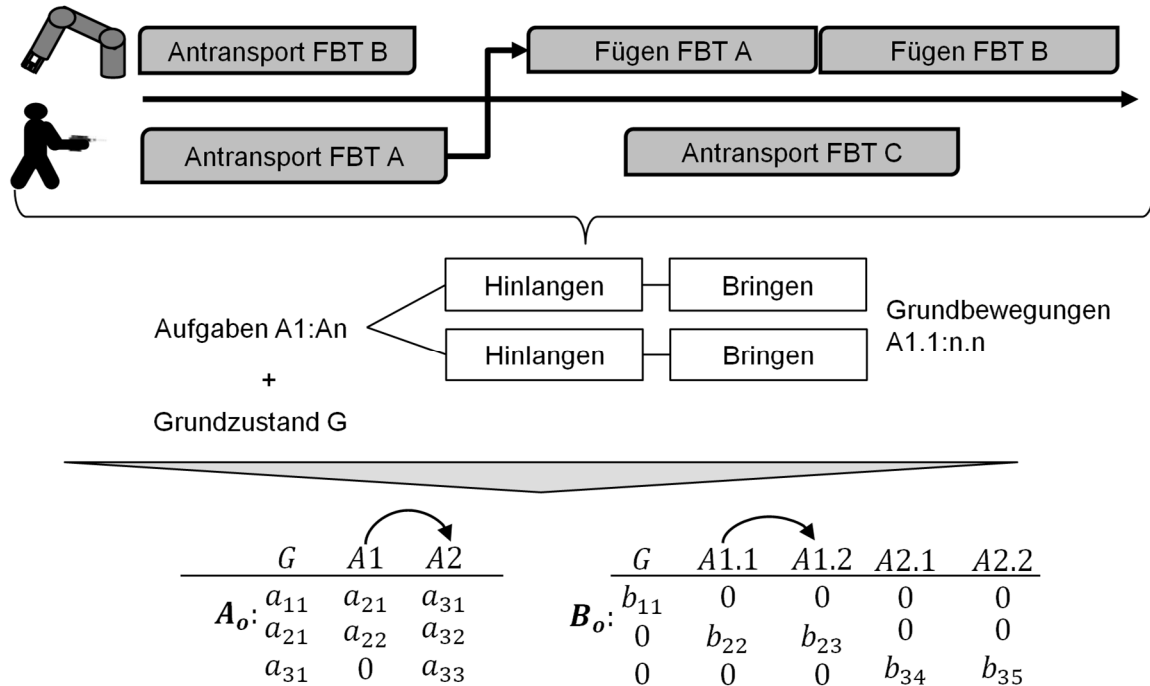


Abbildung 38: Ableitung der Matrizen basierend auf Aufgaben

Die Transitionsmatrix des unteren Layers  $A_u$  wird ähnlich zu der des oberen Layers bestimmt. Hierbei wird nicht die Anzahl der Aufgaben, sondern die Anzahl der Grundbewegungen als Basis verwendet.

Die Emissionsmatrix des oberen Layers  $B_o$  beschreibt die Wahrscheinlichkeit, dass zu einem Zustand bestimmte Emissionen ausgesandt werden. Dies entspricht der Wahrscheinlichkeit, dass bei einer Aufgabe die Grundbewegungen ausgeführt werden. Umgekehrt bedeutet dies, dass bei Auftritt einer Hinlang-Bewegung zu einer Speichereinrichtung die Wahrscheinlichkeit besteht, dass diese zu einer Aufgabe gehört. Um diese Wahrscheinlichkeiten in der Emissionsmatrix zu beschreiben, werden zunächst die Anzahl der Grundbewegungen in Bezug auf die Aufgaben bestimmt.

Die Emissionsmatrix  $B_U$  im unteren Layer wird über die Dichtefunktion der Trajektorien der Handpositionen und der Bewegungsrichtung beschrieben. So kann zu jedem Zeitpunkt bestimmt werden, mit welcher Wahrscheinlichkeit eine

Grundbewegung vorliegt. Zunächst werden dazu die Trajektorien der einzelnen Grundbewegungen zu den Aufgaben berechnet. Unter Beachtung der Gleichungen 6-6 und 6-7 können die Trajektorien berechnet werden:

$$\overrightarrow{r_{hinlangen}} = \overrightarrow{S_n} - \overrightarrow{P_{AP}} \quad (Gl. 6-6)$$

$$\overrightarrow{r_{bringen}} = \overrightarrow{P_{MP}} - \overrightarrow{S_n} \quad (Gl. 6-7)$$

mit:  $r$ : Distanzvektor der Grundbewegung

$S_n$ : Position der Speichereinrichtung

$P_{AP}$ : Ausgangsposition

$P_M$ : Montageposition

Da es sich, wie bereits erläutert, bei natürlichen Handbewegungen nicht um gradlinige Trajektorien handelt, wird die berechnete Trajektorie geteilt. Der Mittelpunkt der Trajektorie wird wie folgt bestimmt:

$$m_n = \frac{W_2 + W_1}{2} \quad (Gl. 6-8)$$

mit  $m_n$ : Mittelpunkt der Trajektorie n

$W_n$ : Anfangs- bzw. Endpunkt der Trajektorie n

Dieser wird um einen Wert in Richtung z-Achse verschoben, sodass sich zwei Trajektorien für die Bewegung ergeben. Über die Länge der Trajektorie wird der Wert der Erhöhung ermittelt. Um Abweichungen auf der Bewegungsbahn der Hand von der berechneten Trajektorie zu integrieren, werden Gaußsche Verteilungen um die Trajektorien gebildet. Diese Abweichungen der Bewegungsbahn werden normalerweise über ein Training bestimmt. Da dies in diesem Ansatz jedoch entfallen soll, um die flexible Einsetzbarkeit des Systems zu gewährleisten, werden Zufallspunkte um die Trajektorie berechnet. Dazu werden die Trajektorien in  $Z$  Zwischenpunkte zerlegt. Um die Zwischenpunkte werden jeweils Zufallspunkte generiert, die in einer Matrix abgespeichert werden. Damit kann abschließend für jede Teiltrajektorie eine Dichtefunktion berechnet werden, die im Betrieb Anwendung findet.

Neben den Trajektorien wird für jede Grundbewegung deren Bewegungsrichtung bestimmt, die verwendet wird, um eine größere Robustheit des Systems zu erreichen. Dies ist vor allem bei Überschneidungen der Trajektorien relevant, da dort mehrere Grundbewegungen mit einer ähnlich großen Wahrscheinlichkeit detektiert werden

und nicht eindeutig erkannt werden. Für jede Grundbewegung wird ein Sollwinkel in der x-y-Ebene berechnet. Abbildung 39 zeigt eine schematische Darstellung der Grundbewegung mit ihrem Winkel.

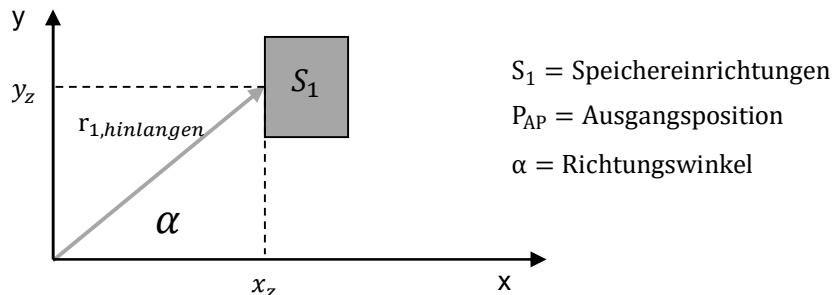


Abbildung 39: Ableitung des Richtungswinkels der Grundbewegung

Die Berechnung des Winkels erfolgt mit nachfolgender Formel:

$$\alpha_t = \arctan\left(\frac{y_t - y_{t-1}}{x_t - x_{t-1}}\right); t = 1, 2, 3 \dots, T \quad (\text{Gl. 6-9})$$

mit  $\alpha_t$ : Winkel der Trajektorie  $t$

$x_t$ : x-Koordinate der Trajektorie

$y_t$ : y-Koordinate der Trajektorie

Mit dieser Vorbereitung des Modells kann das HMM folglich verwendet werden. Bei Detektion einer Aufgabe durch das HMM erfolgt die Weitergabe dieser Information an das Robotersystem, das auf diesen Input wartet. Um eine möglichst reibungslose Zusammenarbeit zwischen den Partnern zu ermöglichen, muss die Ausführungsdauer einer Tätigkeit abgeschätzt werden. Nach HUBER ET AL. (2013) stellen die Erkennung der Tätigkeit und der Ausführungsdauer eine menschliche Schlüsselkompetenz in der Zusammenarbeit dar. Die Abschätzung der Ausführungsdauer als zusätzliche Information zur reinen Erkennung der Tätigkeit des Menschen ermöglicht, dass das Robotersystem seine Aufgaben auch zur richtigen Zeit starten kann, ohne auf ein physisches Signal warten zu müssen.

MTM bietet die Möglichkeit zur Vorberechnung von Montagezeiten und wird in der Sollzeitbestimmung von Montagezeiten eingesetzt (vgl. LOTTER & WIENDAHL (2006)). Da sowohl die Beschreibung der Aufgabenblöcke als auch die Erkennung der Tätigkeiten auf den Grundbewegungen nach MTM basieren, bietet sich dieses System als Grundlage für die Berechnung der Sollzeiten an. Die in MTM-Normwertkarten angegebenen Zeiten hängen von einigen, die Montage betreffenden

Umweltfaktoren ab. Dazu gehören die Bewegungslänge, die Position der Speichereinrichtung, die Art der Speichereinrichtung oder die Präzision beim Fügevorgang. Damit können für alle 5 Grundbewegungen Zeiten für deren Dauer bestimmt werden.

Um die Ausführungsdauer für die jeweilige Tätigkeit bestimmen zu können, sind die oben aufgeführten Faktoren notwendig. Diese können jedoch dem Aufgaben- und Umweltmodell und dem übergebenen Ablauf der Aufgaben entnommen werden. Die Informationen zur Berechnung der Zeiten der einzelnen Aufgaben sind im Wissensmodell hinterlegt, auf das hier zugegriffen wird. In diesem Schritt werden dann für jede durchzuführende Tätigkeit jeweils die Zeiten für die Grundbewegungen aus dem Abstand zwischen einer festgelegten Ausgangsposition und der Position der Speichereinrichtung bzw. dem Abstand zwischen der Position der Speichereinrichtung und der Montageposition berechnet.

Nachdem die Elemente des Modells konfiguriert und erstellt sind, müssen die Schnittstellen für die Tätigkeitserkennung zwischen Robotersteuerung und Tätigkeitserkennung definiert und aufgebaut werden. Dies erfolgt im nachfolgenden Schritt.

### **6.2.3 Integration der Tätigkeitserkennung durch Schnittstellendefinition**

Zur beidseitigen Kommunikation zwischen Roboter und Tätigkeitserkennung müssen die Schnittstellen definiert werden. Aus dem Ablauf, der an das Programmiermodul übergeben wird, gehen die Tätigkeiten hervor, die über die Tätigkeitserkennung abgebildet werden und damit als Schnittstelle vorliegen müssen. Diese Schnittstelle steht auf Seite der Tätigkeitserkennung zur Verfügung und muss auf der Robotersteuerung sowohl im Programm integriert als auch als Eingang definiert sein. Die Integration in das Roboterprogramm selbst wurde in Abschnitt 6.2.1 erläutert. Für die Tätigkeiten wird auf beiden Seiten eine Variable definiert, die der Tätigkeit entspricht. Die Informationen zu den Namen der Tätigkeiten inklusive Schnittstelleninformationen der Robotersteuerung werden zusätzlich zu dem Ablauf an das Programmiermodul übergeben.

Eine zu erkennende Tätigkeit bzw. deren Variable wird im Roboterprogramm in die Kontrollstruktur integriert, die zu der Aufgabe gehört, bei der die Abhängigkeit zur Tätigkeit des Menschen besteht. Darüber hinaus muss dieser Eingang auf der Robotersteuerung mit dem gleichen Namen als Variable definiert werden, so dass das Signal entsprechend empfangen werden kann.

Auf der Seite der Tätigkeitserkennung müssen die Variablen zu den Tätigkeiten ebenfalls bekannt sein, so dass bei Erkennen einer Tätigkeit das entsprechende Signal an den Roboter gesendet werden kann. Daher werden auf Basis der übergebenen Informationen zu den Schnittstellen auch die Schnittstellen in der Tätigkeitserkennung definiert (vgl. Abbildung 40).

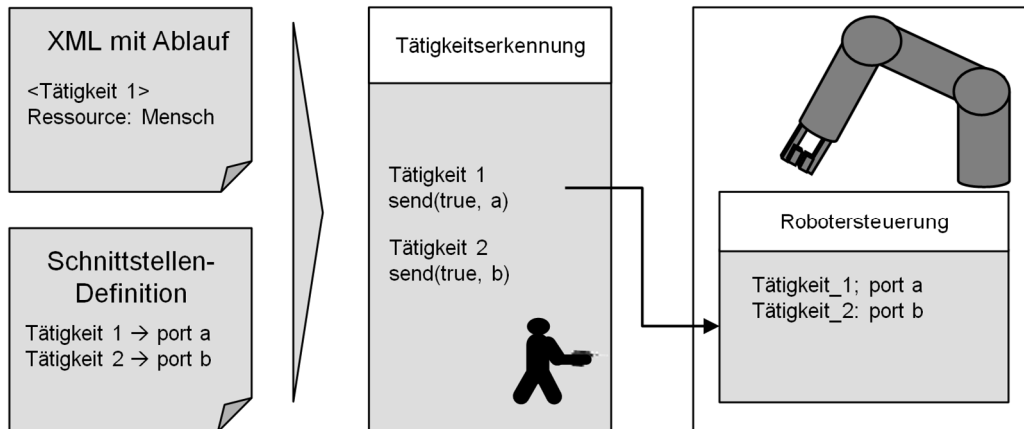


Abbildung 40: Definition der Schnittstellen in Tätigkeitserkennung und Robotersteuerung

### 6.2.4 Simulation

Mit Hilfe einer der Programmgenerierung nachgelagerten Simulation wird der Gesamtablauf der Montageaufgabe überprüft. Dazu werden sowohl das generierte Roboterprogramm als auch die Aufgaben des Menschen an das Simulationssystem übergeben, das entsprechend einen Simulationsablauf generiert (vgl. Abbildung 41). Neben der reinen Überprüfung des Ablaufs dient die Simulation des Szenarios auch der Überprüfung des Ablaufs auf sicherheitskritische Bewegungen. Außerdem kann der Mensch über die Visualisierung des Ablaufs auf den Betrieb vorbereitet werden. Damit kann der Anforderung begegnet werden, dass die Bahnen für den Menschen vorhersehbar sein sollen.

Zur Simulation eines Szenarios einer Mensch-Roboter-Kooperation wurde in Abschnitt 3.2.5 bereits festgestellt, dass hierfür wenige Simulationsumgebungen und Ansätze zur Verfügung stehen. Die Anforderung an die hier verwendete Anwendung ist insbesondere die automatische Generierung des Simulationsszenarios und damit die Einbindung in das Gesamtsystem. Die Grundvoraussetzung für die Auswahl ist jedoch die Möglichkeit der kombinierten Darstellung von Roboter und Mensch.

Die Herausforderung bei der Simulation des Szenarios für die Mensch-Roboter-Kooperation liegt in der Simulation des Menschen. Viele Programme zur



Robotersimulation bieten bisher nicht die Möglichkeit zur Simulation eines Menschen. Um ein geeignetes Programm zu identifizieren, wurden Simulationsprogramme untersucht, die zur Robotersimulation oder zur Simulation von Montagearbeitsplätzen eingesetzt werden. Einen Vergleich möglicher Simulationsprogramme enthält nachfolgende Tabelle 3.

*Tabelle 3: Vergleich von Simulations-Software*

<b>Programm</b>	<b>Hersteller</b>	<b>Industrie-roboter</b>	<b>Digitales Menschmodell</b>
v-rep	Copellia Robotics	X	X
Webots	Cybertotics Ltd.	X	-
RoboDK	RoboDK Inc	X	-
ROS		X	-
HumanBuilder	Dessault Systems	-	X
Jack	Siemens PLM	-	X
Ramsis	Human Solutions	-	X

Legende: X: Simulation möglich

Aus dem Vergleich der verschiedenen Simulationsprogramme geht hervor, dass nur v-rep die Möglichkeit bietet, sowohl den Roboter als auch den Menschen darzustellen und anzusteuern, so dass ein Ablauf abgebildet werden kann. Im Gegensatz jedoch zu den Programmen, die auf die Darstellung digitaler Menschmodelle spezialisiert sind, ist die Variabilität des Menschen in v-rep eingeschränkt. Es kann daher nur ein Mensch verwendet werden. Des Weiteren kann nur ein vorher definierter Ablauf des Menschen abgebildet werden. Ungeplantes Verhalten des Menschen, das in einer Mensch-Roboter-Kooperation zu Gefährdungen führen kann, kann nicht untersucht werden. Jedoch reicht diese Darstellung zunächst aus, um den groben Ablauf zwischen Mensch und Roboter darzustellen. Das Vorgehen zur Simulation des Arbeitsablaufs von Mensch und Roboter innerhalb des Programmiermoduls zeigt nachfolgende Abbildung 41.

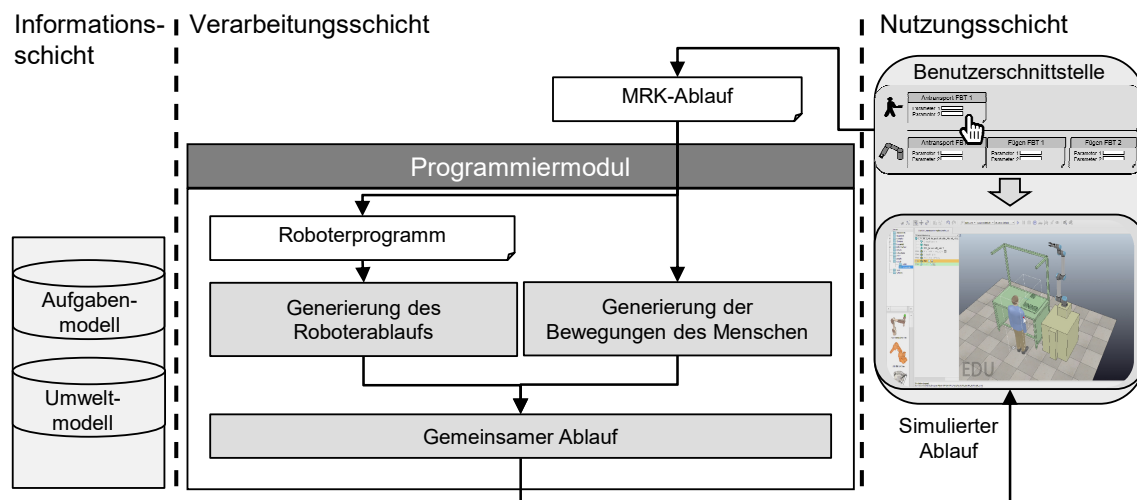


Abbildung 41: Vorgehen zur Simulation des Arbeitsablaufs

Basierend auf dem Roboterprogramm, das durch den Programmplaner erstellt wurde, wird der Ablauf des Roboters in einer Simulationsumgebung generiert. Ähnlich dazu werden aus der Datei, die den Ablauf von Mensch und Roboter enthält, die Aufgaben des Menschen identifiziert, für die die Bewegungen des Menschen in der Simulationsumgebung generiert werden. Daraus folgend gibt es für beide Partner einen Ablauf in der Simulation. Durch die Startzeiten, die in der Szenarioplanung angenommen werden, können die Abläufe beider Ressourcen in der Simulation aufeinander abgestimmt werden, so dass die Darstellung des Gesamtablaufs erfolgen kann.

Der generierte Simulationsablauf wird anschließend in der Benutzerschnittstelle angezeigt. Dadurch kann einerseits der Mensch auf den gemeinsamen Ablauf vorbereitet werden, so dass die Gefahr nicht vorhersehbarer Bewegungen sinkt. Andererseits wird die Simulation für die in Abschnitt 6.4 dargestellte Sicherheitsbetrachtung verwendet.

### 6.3 Betriebsmodul

Im Anschluss an die Erstellung des Roboterprogramms und die Konfiguration des Modells zur Tätigkeitserkennung erfolgt der Betrieb in der Mensch-Roboter-Kooperation. Das Betriebsmodul läuft im Gegensatz zum Planungs- und Programmiermodul online, was bedeutet, dass Daten fortwährend des Betriebs übertragen und verarbeitet werden. Das Betriebsmodul gliedert sich in die beiden Bereiche Tätigkeitserkennung und Kollisionsvermeidung.

### 6.3.1 Tätigkeitserkennung

Durch die Tätigkeitserkennung werden die durchgeführten Bewegungen des Menschen detektiert und darüber die Tätigkeit erkannt. Die Information einer erkannten Tätigkeit wird an den Roboter weitergeleitet. Der Aufbau und die Konfiguration des Modells wurden in Abschnitt 6.2.2 bereits erläutert. Im Betriebsmodus wird das Modell angewendet.

Wie zuvor erwähnt, dienen die Positions- und Richtungsdaten der Hände des Menschen als Eingangsdaten für das Modell. Diese müssen daher detektiert und verarbeitet werden, wozu Kamerasysteme verwendet werden. Kamerasysteme bieten dabei den Vorteil, dass der Nutzer keine zusätzlichen Geräte, wie z. B. Datenhandschuhe oder ein Armband, tragen muss. Die Kamerasysteme sind fest am Montagearbeitsplatz verbaut und nehmen kontinuierlich die Positionen der Körperteile, insbesondere der Hände, auf. Um möglichst genaue Positionsdaten zu erhalten, werden mehrere Kamerasysteme eingesetzt. Wie in Abschnitt 6.2.2 dargelegt, können insbesondere für die Grundbewegungen *Hinlangen* und *Bringen* Bewegungen der Hand festgestellt werden. Bei einem Fügevorgang bewegt sich die Hand nur in geringem Maße, meist in z-Richtung, über der Montageposition. Aufgrund dieses Sachverhalts wird im Folgenden aus der Detektion des Antransports eines Bauteils, bestehend aus Hinlangen, Greifen und Bringen, auf den Fügevorgang geschlossen. Die Zeiten für die einzelnen Grundbewegungen wurden im Rahmen der Konfiguration berechnet und an das System weitergegeben. Somit liegt für jede Tätigkeit ein Zeitenmodell vor. Diese Zeiten sind relevant für die Übergabe des Signals an den Roboter, das die Information einer erkannten Tätigkeit enthält. Wird eine Tätigkeit erkannt, so kann das Signal entweder unmittelbar oder verzögert an den Roboter gesendet werden. Soll das Signal verzögert versendet werden, so kann nach Erkennung einer Tätigkeit die Signalübergabe um die berechnete Restzeit der Tätigkeit verzögert werden.

Da die Erkennung des Fügevorgangs selbst mit dem Modell nicht möglich ist, wird die berechnete Zeit darüber hinaus zur Abbildung des Fügevorgangs verwendet. So werden der Restzeit einer Tätigkeit Antransport, die erkannt werden kann, die berechneten Zeiten für die Grundbewegungen Fügen und Loslassen hinzugefügt. Dieses Vorgehen gilt, wenn der Mensch sowohl die Aufgabe Antransport als auch Fügen eines Bauteils übernimmt. Ein beispielhafter Ablauf einer zu erkennenden Tätigkeit ist in der nachfolgenden Abbildung 42 dargestellt.

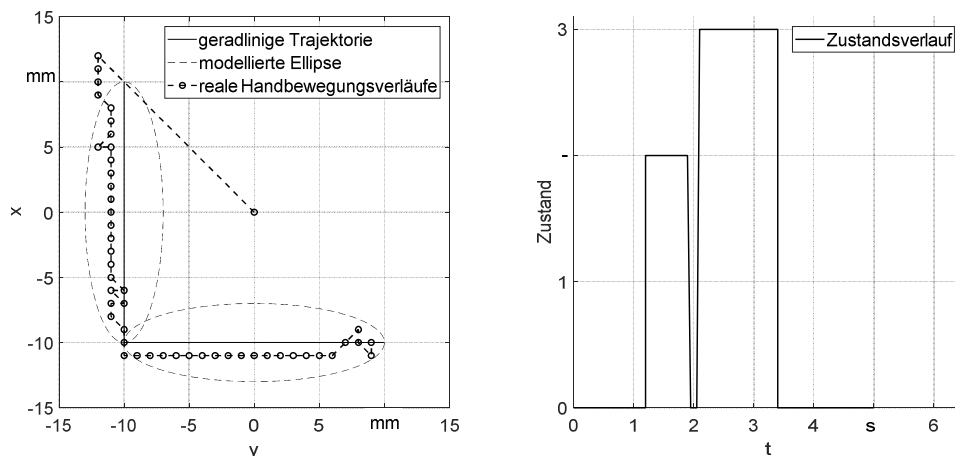


Abbildung 42: Zustandsänderung bei Erkennung einer Tätigkeit

Darin zu erkennen sind die Übergänge zwischen einzelnen Zuständen während der Tätigkeitserkennung. Im linken Bild sind die Solltrajektorien und die aufgenommenen Punkte der Hand zu erkennen. Auf der rechten Seite sind die Zustände abgebildet. Zustand 1 zeigt den Grundzustand, Zustand 2 entspricht dem Hinlangen und Zustand 3 stellt die restlichen Grundbewegungen Greifen, Bringen, Fügen und Loslassen dar.

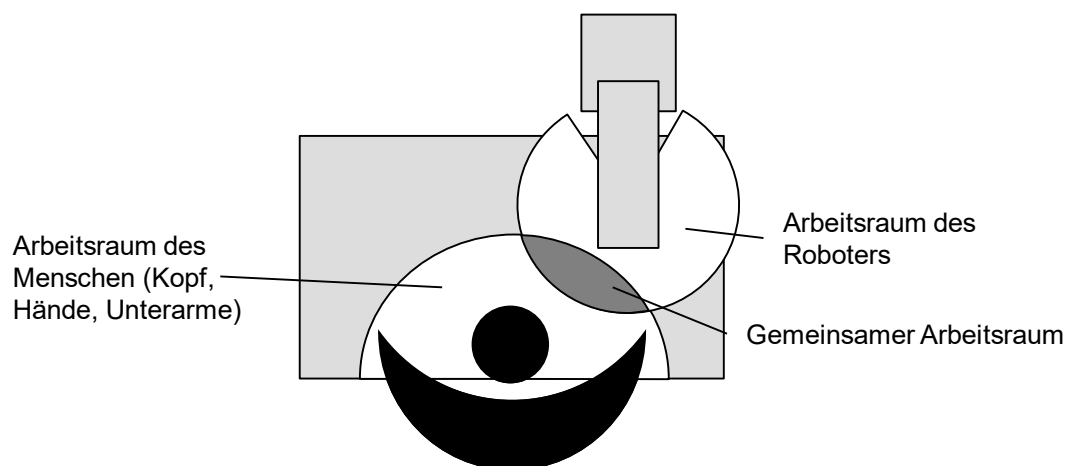
### 6.3.2 Kollisionsvermeidung

Da im Rahmen dieser Arbeit zur Programmierung inklusive Tätigkeitserkennung bereits Kamerasysteme zur Detektion des Menschen zur Anwendung kommen, kann eine kamerabasierte Kollisionsvermeidung integriert werden. Dies erscheint zielführend, da eine dynamische Umplanung der Roboterbahn abhängig von der Bewegungsbahn des Werkers nicht beabsichtigt ist.

Für die Tätigkeitserkennung während des Betriebs werden die Bewegungen des Menschen, also die Positionsänderungen, aufgenommen und ausgewertet. Diese Daten stehen folglich auch einer Kollisionsvermeidung zur Verfügung, so dass die Position des Menschen im Arbeitsraum bekannt ist. Des Weiteren können Informationen, die den Roboter betreffen, über die Robotersteuerung ausgelesen werden. Stehen sich Roboter und Mensch gegenüber, sind vor allem die Position des Endeffektors und der letzten Achse von Interesse, da hier eine Kollision auftreten kann.

Für den Fall einer Kollision zwischen Mensch und Maschine findet insbesondere das Bounding Box Verfahren Anwendung. GOTTSCHALK (2000) beschreibt in seiner Arbeit eine Anwendung dieses Verfahrens. Das Bounding-Box-Verfahren basiert auf Hüllkörpern (engl. Bounding Volumes), die einen Körper umhüllen (ENGESER 2014). Dabei können unterschiedliche Hüllkörperformen verwendet werden, die unterschiedliche Eigenschaften haben. Verwendet werden kugelförmige Hüllkörper, an Achsen ausgerichtete Quader, willkürlich orientierte Quader oder konvexe Hüllen (GOTTSCHALK 2000).

Das Bounding-Box Verfahren mit Hüllkörpern von GOTTSCHALK (2000) wird auf die Anwendung in dieser Arbeit übertragen. Davon ausgehend, dass der Roboter gegenüber dem Menschen oder schräg seitlich zu ihm angeordnet ist, kann hier vor allem von Kollisionen der Hände bzw. Unterarme oder des Kopfes mit dem Endeffektor bzw. der letzten Achse des Roboters ausgegangen werden (vgl. Abbildung 43).



*Abbildung 43: Schematische Abbildung der Positionen und Arbeitsräume von Mensch und Roboter*

Daher werden für den Endeffektor ein Gefahrenraum und für die genannten Körperteile, die kollidieren könnten, ein Sicherheitsraum modelliert. Für den Gefahrenraum des Endeffektors kann je nach Art des Endeffektors auch die Form und Größe des Gefahrenraums eingestellt werden. Als Information für die Berechnung der Position und Orientierung des Gefahrenraums um den Endeffektor wird die Position und Orientierung des Tool Center Points verwendet. Ähnlich der Modellierung des Gefahrenraums um den Endeffektor des Roboters erfolgt die Modellierung der Sicherheitsräume um Hände, Unterarme und den Kopf des Menschen. Zur Modellierung dieser Sicherheitsräume werden als Eingangsinformation ebenfalls die Positionen der jeweiligen Gliedmaße verwendet.

Dabei wird für jedes Gliedmaß ein eigener Sicherheitsbereich, bestehend aus einem quader- oder kugelförmigen Hüllkörper, modelliert. Diese Sicherheitsbereiche in ihrer Gesamtheit gelten als Sicherheitsraum des Menschen. Die Positionsinformation kann aus den Kamerabildern abgeleitet werden. Sind die Gefahrenräume des Roboters und die Sicherheitsräume des Menschen modelliert, so werden diese stetig abgeglichen und auf Kollisionen überprüft.

Die Kollisionsprüfung unterscheidet sich für die Fälle zweier Quader, einer Kugel und eines Quaders und zweier Kugeln. Zur Kollisionsprüfung zweier Quader wird das dreidimensionale Problem als mehrere zweidimensionale betrachtet. Hierfür wird das Koordinatensystem parallel zu den Seiten eines Quaders ausgerichtet, woraufhin an den Eckpunkten geprüft wird, ob dort eine Überschneidung vorliegt. Nur wenn an drei Achsen Überschneidungen vorliegen, kann eine Kollision detektiert werden. Bei der Prüfung auf Kollision zwischen einem Quader und einer Kugel wird die Kugel ebenfalls als Quader betrachtet, wobei der Quader um die Kugel gelegt wird. Der Quader der Kugel wird dabei so modelliert, dass er bereits parallel zu dem zweiten Quader steht, was dazu führt, dass nur noch drei Achsen auf Überschneidungen überprüft werden müssen. Bei zwei Kugeln wird lediglich der Abstand der Mittelpunkte mit der Summe der Radien verglichen. Ist der Abstand größer, so liegt keine Kollision vor (ECKSTEIN 1999).

### **6.4 Sicherheitsbetrachtung**

Wie in den Abschnitten 2.4.4, 3.3.1 und 4.5 ersichtlich wird, bildet die Betrachtung der Sicherheit die Grundlage für die Inbetriebnahme einer Applikation, in der Mensch und Roboter im gleichen Arbeitsraum arbeiten. Durch festgesetzte Rahmenbedingungen zum Einsatz des Programmiersystems, wie sie in Abschnitt 5.2 beschrieben sind, kann der Aufwand der Risikoanalyse vermindert werden. Im Planungsprozess wird die Sicherheit insbesondere bei der Arbeitsplatzgestaltung einbezogen, wobei darauf geachtet wird, dass der Roboter möglichst wenig scharfe Bauteile handhabt oder Bewegungen in nach ISO TS 15066 kritischen Bereichen des Menschen ausführt. Als besonders kritische Bereiche gelten hier der Kopf- und Halsbereich. Im Anschluss an den Aufbau der Applikation wird die Gefährdungsanalyse, wie in Abschnitt 2.4.4 erläutert, durchgeführt.

Das Ziel einer flexibleren Anwendung einer Mensch-Roboter-Kooperation, bei der die Bewegung des Roboters und das bewegte Objekt verändert werden müssen, stellt die Herausforderung dar, dass der Prozess zur Beurteilung der Gefährdungen neu

durchlaufen werden muss (SCHIEMANN ET AL. 2018). Dies schränkt die Flexibilität und damit auch die Anwendung einer MRK-Lösung ein (SIHN ET AL. 2016).

Daher wird in dieser Arbeit betrachtet, wie der Aufwand zur Inbetriebnahme einer Applikation durch Integration von Sicherheitsaspekten in das Programmiersystem und einer unterstützten Risikoanalyse minimiert werden kann, so dass die flexible Anwendung der Mensch-Roboter-Kooperation möglich werden könnte. Dazu wird versucht, einige Elemente der Sicherheitsbetrachtung in das Vorgehen zur Planung und Programmierung zu integrieren, um den Aufwand zur Risikoanalyse zwischen Programmerstellung und Inbetriebnahme des Roboters zu verringern. Eine unterstützte Risikoanalyse muss dennoch erfolgen, um die Applikation in Betrieb zu bringen. Das allgemeine Vorgehen zur Risikobeurteilung zeigt folgende Abbildung 44.

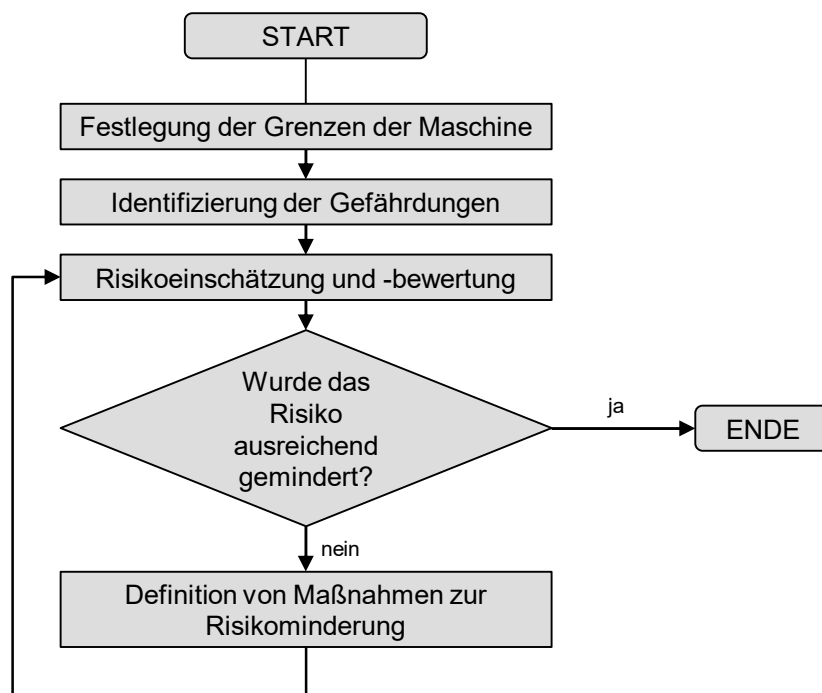


Abbildung 44: Vorgehen zur Risikoanalyse nach DIN ISO 12100

Zur Definition möglicher Maßnahmen zur Vereinfachung des Vorgehens werden zunächst die Gefährdungen diskutiert, die im besonderen Fall einer Mensch-Roboter-Kooperation auftreten können. Dabei liegt der Fokus auf den Gefährdungen, die im normalen Betrieb auftreten können. Weniger werden mögliche Gefährdung anderer Personengruppen, wie Techniker, Reinigungspersonal oder externe Besucher, wie sie von SIHN ET AL. (2017) beschrieben sind, einbezogen. Zunächst liegt eine Gefährdung durch das Werkstück vor. Je nach Beschaffenheit des Werkstücks, wie zum Beispiel beim Vorhandensein scharfer Kanten, geht von ihm eine erhöhte Gefahr

aus. Die möglichen Kontaktarten zwischen Person und Roboter werden in transienten und quasistatischen Kontakt unterteilt. Der Grad der Gefährdung der Person hängt auch von der Art des betroffenen Körperteils ab. Für die einzelnen Körperteile werden in der ISO TS 15066 Kraft- und Druckgrenzen definiert, deren Einhaltung durch eine Messung abgesichert sein muss. Eine Gefährdung in der Kopfregion muss ausgeschlossen werden. Neben dem Werkstück kann auch der Arbeitsplatz zu Gefährdungen führen. So können beispielsweise Quetschungen zwischen Roboter und Objekten im Arbeitsraum auftreten.

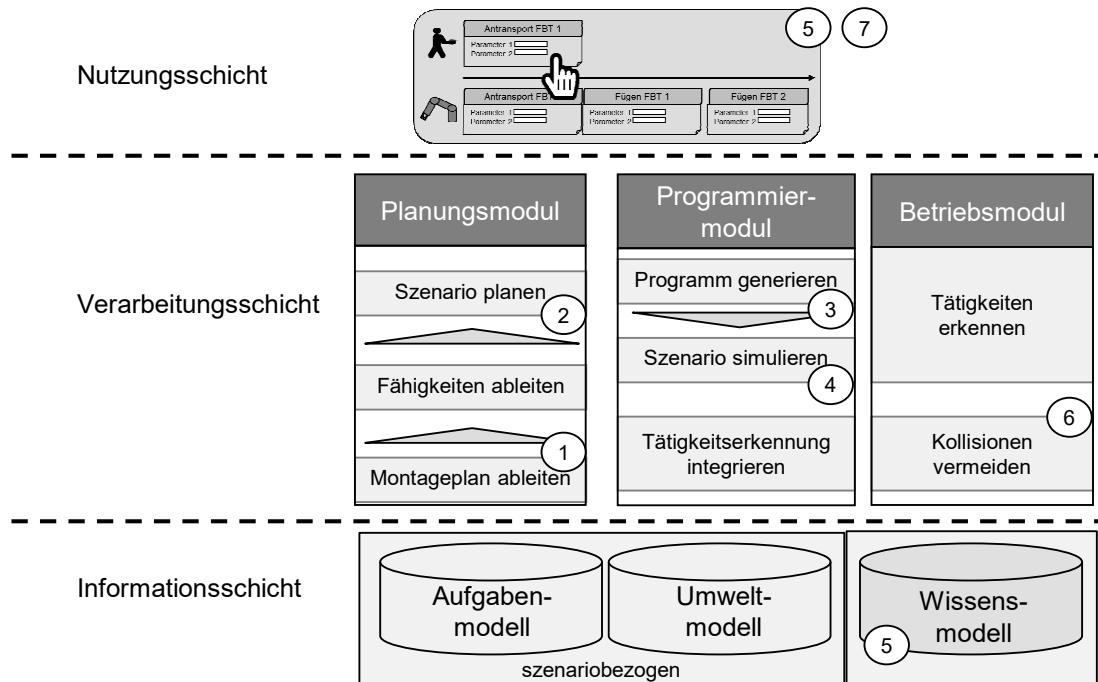
SIHN ET AL. (2017) identifizieren als Gefährdungen bei einer beispielhaften kollaborativen Anwendung:

- Quetschung zwischen Greifer und Werkstück in Behälter
- Quetschung zwischen Arbeitsplatzaufbau und Roboterarm
- Transienter Stoß zwischen Greifer und Werkkraft
- Quetschung von Fingern zwischen Gelenk 1 und Werkzeugschnittstelle
- Quetschung zwischen Roboterfuß und Basisgelenk
- Quetschung zwischen Arbeitsfläche und gegriffenem Werkstück

Aus den beschriebenen Gefährdungen ergeben sich Maßnahmen, die teilweise in einer frühen Phase der Planung des Layouts und der Zuteilung der Aufgaben bzw. Programmierung getroffen werden können. Diese sind beispielsweise die Überprüfung der Beschaffenheit der Werkstücke in einer frühen Phase, die Anordnung der Ressourcen und Komponenten im Arbeitsraum oder die Definition von Arbeitsräumen für den Roboter.

Daraus ergeben sich Angriffsstellen zur Integration von Sicherheitsmaßnahmen in das aufgabenorientierte Programmiersystem. Diese sind in folgender Abbildung 45 dargestellt und wurden teilweise in BERG ET AL. (2018b) vorgestellt.





- ① Bauteilbeschaffenheit
- ② Vermeidung gleichzeitiger Bewegung Richtung Arbeitspunkt
- ③ Definition von Roboterbewegungen mit Einschränkungen
- ④ Simulation der Applikation
- ⑤ Definition von Grenzbereichen und Einschränkung in Benutzerschnittstelle
- ⑥ Kollisionen vermeiden
- ⑦ Unterstützte Risikoanalyse

Abbildung 45: Integration von Sicherheitselementen in das aufgabenorientierte Programmiersystem nach BERG ET AL. (2018)

Einerseits werden Elemente bzgl. der Sicherheit in die Submodule integriert und andererseits soll der Prozess zur Risikoanalyse unterstützt werden, so dass dieser mit weniger Aufwand durchgeführt werden kann, um zur sicheren Applikation zu gelangen. Durch die Qualifizierung der Werkkräfte, wie sie in Abschnitt 7.2.3 beschrieben ist, wird die Basis dafür gelegt, dass der Roboter nur bestimmungsgemäß eingesetzt wird. Die Beschreibung der Elemente für die einzelnen Module erfolgt im nachstehenden Abschnitt. Daran anschließend wird der Prozess zur unterstützten Risikobeurteilung erläutert.

### 6.4.1 Integration in die Module

Die vom Werkstück ausgehende Gefährdung wird bereits in das Planungsmodul integriert. Dabei wird überprüft, inwieweit scharfe Kanten, also Kanten mit einem

Radius weniger als 1mm, am Werkstück vorliegen, die bei Handhabung des Roboters zu einer erhöhten Gefährdung führen. Die mögliche Gefährdung durch Kontakt des Roboters mit der Person wird bei der Layoutprüfung im Planungsmodul bereits abgeschätzt und weiterhin bei der Simulation im Anschluss an die Programmerstellung überprüft. Wird hier festgestellt, dass erhöhte Gefahren vorliegen, können die Aufgaben den Partnern anders zugeteilt oder die Geschwindigkeit des Roboters verringert werden. Über die maximalen Druck- und Kraftgrenzen für die betroffenen Körperregionen kann die Geschwindigkeit des Roboters berechnet werden. Damit kann für eine bestimmte Aufgabe mit Hilfe der Berechnungsvorschrift aus der ISO TS 15066 die Geschwindigkeit des Roboters entsprechend angepasst werden. Durch diese Anpassung soll der Aufwand bei der Kraft- und Druckmessung verringert werden.

Für die Programmierung werden im Wissensmodell sicherheitsrelevante Parameter zu den einzelnen Funktionsbausteinen hinterlegt, wie in Abschnitt 5.5.3 beschrieben. Diese Parameter führen zu Grenzbereichen, die den Anforderungen der ISO TS 15066 entsprechen. Bei einem Greifer werden beispielsweise die maximalen Greifkräfte festgelegt. Ein Schraubvorgang soll zum Beispiel so abgebildet werden, dass der Schraubkopf nur um maximal 4mm angehoben wird, was der Grenze zum Einbringen eines Fingers zwischen Schraubkopf und Bauteil entspricht. Lässt das Bauteil dies nicht zu, so wird die senkrechte und geschwindigkeitsreduzierte Verfahrensbewegung, wie beim Anfahren von Positionen, verwendet. Gemäß SIHN ET AL. (2017) sollen außerdem Greifpunkte langsamer angefahren werden und Annäherungen möglichst im rechten Winkel erfolgen, um eine Gefahr durch Scherung zu verringern. Die Bahnen des Roboters für die unterschiedlichen Aufgabenblöcke sind im Wissensmodell fest definiert. Dies ergibt den Vorteil, dass unvorhersehbare Bewegungen des Roboters nicht eintreten sollten. Des Weiteren kann über diese Definition der Bahnen die Risikoanalyse unterstützt werden, da die möglichen Gefährdungen durch den Roboter bekannt sind. Die Geschwindigkeit des Roboters kann über die Beschaffenheit des Bauteils und der angestrebten Bahn berechnet werden.

Die im Betriebsmodul verwendeten Kamerasysteme entsprechen nicht den Anforderungen an ein sicheres Kamerasystem. Dennoch kann über die Redundanz von mehreren nicht-sicheren Kamerasystemen ein sicheres System erreicht werden. Da zur Kollisionsvermeidung zwei unterschiedliche Kamerasysteme verwendet werden, bietet das System eine Sicherheit an. Dadurch verhilft das System im Betrieb zu einer Minderung des Risikos von Quetschungen zwischen Endeffektor mit gegriffenem Werkstück und der Ablageposition des Bauteils. Durch die Einstellung

der Größe der Sicherheits- bzw. Risikoräume bei der Kollisionsvermeidung abhängig vom Werkstück und durch die Anpassung der Geschwindigkeit soll gewährleistet werden, dass der Druck oder die Kraft im Falle einer Kollision nicht zu hoch wird.

### 6.4.2 Unterstützte Risikoanalyse

Die Integration von Sicherheitselementen in das Planungs- und Programmiermodul unterstützt die planende Person dabei, dass die Applikation eine CE-Kennzeichnung erhalten kann. Dennoch muss die Gesamtapplikation durch eine befähigte Person abgenommen werden, wobei überprüft wird, ob die Risiken, die aus der Gefährdungsanalyse hervorgehen, entsprechend gemindert wurden.

Durch die unterstützte Risikoanalyse soll der Aufwand zur Überprüfung einer Änderung und der damit verbundenen Anpassung der CE-Konformitätserklärung verringert werden, um damit einen flexibleren Einsatz des Roboters zu ermöglichen. Nach den Vorgaben des BUNDESMINISTERIUMS FÜR ARBEIT UND SOZIALES (2015) muss bei einer Änderung der Maschine untersucht werden, ob durch die Veränderung neue Gefährdungen betrachtet werden müssen oder sich ein Risiko erhöht. Für die Mensch-Roboter-Kooperation geht die Risikoanalyse nach aktueller Normenlage mit der Messung oder Berechnung der Kraft- und Druckwerte einher.

Eine neue Gefährdung, insbesondere für den Menschen, kann unter anderem in folgenden Fällen auftreten:

- Eine Änderung der Geometrie des Bauteils kann zu veränderten Druckwerten führen, da die Flächenpressung verändert wird.
- Eine Änderung des Gewichts des Bauteils führt zu einer Veränderung der Kraftwerte durch eine höhere bewegte Masse.
- Eine Änderung des Roboterprogramms bei einer veränderten Aufgabe kann zu anderen Gefährdungen, wie bspw. Quetschstellen, führen.
- Eine Änderung des Endeffektors kann zu verschiedenen anderen Gefährdungen führen.
- Eine Änderung der Umgebung des Arbeitsplatzes kann zu Quetschgefahren zwischen einer Wand und dem Robotersystem führen.

Demnach führen die meisten Produktänderungen, Ressourcenänderungen oder der Einsatz an einem anderen Arbeitsplatz zu einer neuen Gefährdung oder einem erhöhten Risiko. Um den flexibleren Einsatz daher zu ermöglichen, bedarf es der Vereinfachung der Durchführung der Gefährdungs- und Risikoanalyse. Dazu kann unter anderem der Ansatz von SCHIEMANN ET AL. (2018) herangezogen werden.

Dabei wird die Risikoanalyse für unterschiedliche Werkzeuge durchgeführt, die im Rahmen in einer MRK eingesetzt werden können. Im Ansatz von Plug & Play können diese Werkzeuge gewechselt und ohne weitere ausführliche Gefährdungsanalyse eingesetzt werden.

Dieser Ansatz wird in die unterstützte Risikoanalyse integriert und es dürfen im Rahmen des vorgestellten Programmiersystems nur Komponenten, wie beispielsweise Greifer, verwendet werden, die als sicher gelten. Zur Durchführung der Risikoanalyse müssen zunächst allgemeine Fragen zur Applikation beantwortet werden. Dazu gehört die Abfrage nach den verwendeten Komponenten und ob sich keine nicht verwendeten Objekte im Arbeitsraum befinden. Außerdem soll das System nur im freien Raum und nicht an einer Wand eingesetzt werden. Anschließend werden die Aufgabenblöcke hinsichtlich der Gefährdungen untersucht. Da bei den Funktionsbausteinen die Gefährdungen bekannt sind und die Geschwindigkeit des Roboters basierend auf der Beschaffenheit des Werkstücks eingestellt wird, werden die Gefährdungen gemeinsam mit der Person überprüft. Der Ablauf zur unterstützten Risikoanalyse ist im Anhang A5 einzusehen. Die Analyse der Gefährdungen der Aufgaben *Antransport* und *Fügen* zeigt folgende Abbildung 46.

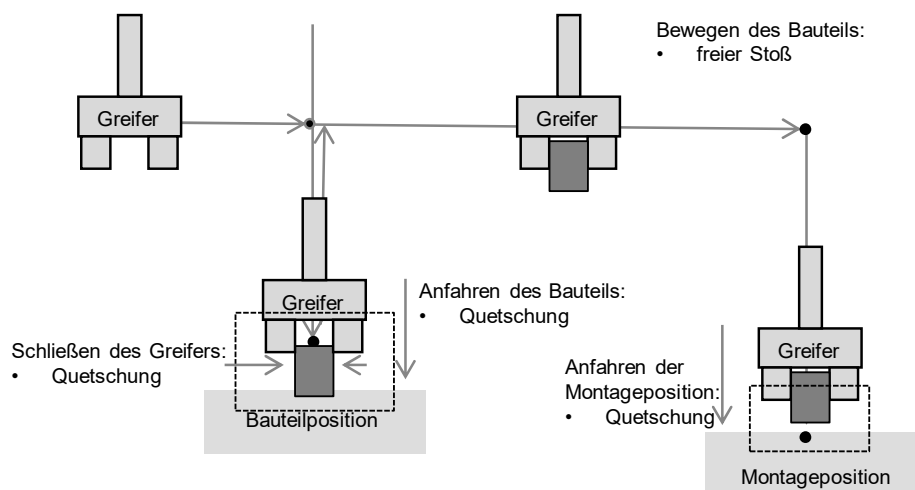


Abbildung 46: Identifikation der Gefährdungen für die Funktion *Antransport*

Neben der Betrachtung der Sicherheit ist die Qualifizierung zum Umgang mit dem Roboter- und Programmiersystem ein wesentliches Element bei der Einführung der Applikationen. Daher wird im nächsten Kapitel ein Vorgehensmodell zur Einführung des Programmiersystems im Unternehmen beschrieben, das die Definition der Rollen für den Einsatz des Programmiersystems und die Qualifizierung der Personen behandelt.

## 7 Vorgehen zur Integration des Programmiersystems

### 7.1 Beschreibung des Vorgehensmodells

Zur Einführung eines aufgabenorientierten Programmiersystems in einem Unternehmen entwickelt BACKHAUS (2016) ein Vorgehensmodell. Dazu erweitert er das Vorgehensmodell nach NEBDAL (2013) um Elemente, die sich konkret auf den Fall der aufgabenorientierten Programmierung beziehen. Ein besonderer Fokus bei der Erweiterung des Modells durch BACKHAUS (2016) liegt auf der Entwicklung eines Reifegradmodells für die Orientierungsphase. Das Ziel des Reifegradmodells ist es, durch eine Bewertung der aktuellen Situation im Unternehmen das Potenzial für den Einsatz eines aufgabenorientierten Programmiersystems abschätzen zu können. Weitere Ergänzungen des Modells liegen in der Analyse- und Designphase, wobei, bezogen auf das Programmiersystem, eine Analyse mittels Informationsmodell und typischer Montagesysteme vorgenommen wird. Darauf aufbauend werden im Design-Schritt das Aufgabenmodell und das Wissensmodell definiert. Da dieses Vorgehensmodell zur Integration eines Programmiersystems als gute Basis für eine Weiterentwicklung in Bezug auf die Mensch-Roboter-Kooperation gilt, wird es als Grundlage herangezogen.

Im Hinblick auf die Einführung der Mensch-Roboter-Kooperation im Unternehmen beschreiben MÜLLER ET AL. (2019) eine Einführungsstrategie. Dabei stellen sie fest, dass die Werkkräfte bei dem Einführungsprozess einbezogen werden müssen. Ein wichtiges Element in diesem Verfahren ist die Kommunikation der Veränderung gegenüber den Werkkräften. In ihrem Vorgehen zur Mitarbeitendeneinbindung ist daher der erste Schritt eine frühzeitige Information an die Werkkräfte, verbunden mit der Schaffung von Transparenz. Außerdem sollen Probleme aufgenommen und Bedürfnisse geweckt werden. Die Werkkräfte sollen Entscheidungen treffen können, die dann auch umgesetzt werden. Des Weiteren spielt die Qualifikation der Werkkräfte eine zentrale Rolle. Um einer Gefahr der Überforderung entgegen zu wirken, sollten eine Basis-Qualifizierung durchgeführt und mit einer spezifischen Qualifizierung frühzeitig begonnen werden. Im Zusammenhang mit der Einführung von MRK beschreiben MÜLLER ET AL. (2019) in Anlehnung an RICHTER ET AL. (2015) vier wesentliche Bedürfnisse von Werkkräften. Diese sind: Kompetenz, Verbundenheit, Autonomie und Abwechslungsreichtum.

Zur Anpassung des Modells für das hier betrachtete Programmiersystem wurden im Ausgangsmodell die einzelnen Phasen dahingehend untersucht, welche Ergänzungen

im Vorgehensmodell getroffen werden müssen, um es für das betrachtete System einsetzen zu können. Dabei wurden die Erkenntnisse zur Einführung der Mensch-Roboter-Kooperation beachtet und auf das Modell zur Einführung des Programmiersystems übertragen. Ein Überblick und die Beschreibung des angepassten Modells sind im nachfolgenden Abschnitt dargestellt.

## 7.2 Anpassung des Vorgehensmodells

Zur Anpassung des Modells für den Einsatz in der Mensch-Roboter-Kooperation wurden die fünf Phasen Orientierung, Analyse, Design, Realisierung und Betrieb übernommen. Die Elemente in den Phasen wurden teilweise basierend auf dem bestehenden Modell erweitert oder es wurden weitere Elemente hinzugefügt. Das Modell mit seinen Erweiterungen ist in nachfolgender Abbildung 47 dargestellt.

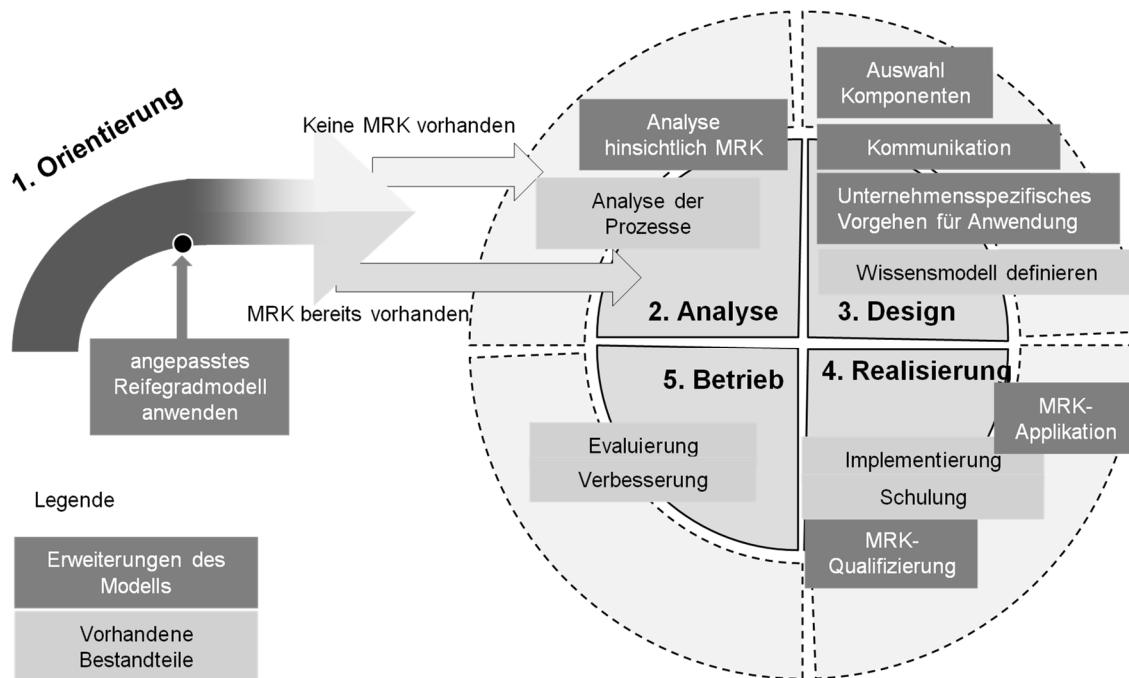


Abbildung 47: Vorgehensmodell zur Einführung des aufgabenorientierten Programmiersystems in einen Betrieb

In der *Orientierungsphase* werden mit Hilfe eines angepassten Reifegradmodells in Form von Fragen das Potenzial und der Aufwand für den Einsatz des Programmiersystems ermittelt. Dabei muss zwischen den beiden Fällen unterschieden werden, dass bereits MRK-Applikationen im Unternehmen vorhanden sind oder noch nicht. Im ersten Fall wird das Programmiersystem zusätzlich dazu eingeführt. Wird noch keine MRK eingesetzt, muss zunächst das MRK-Potenzial im

Zusammenhang mit dem Programmiersystem ermittelt werden. Je nach Reifegrad unterscheiden sich die zusätzlichen Schritte in den weiteren vier Phasen. Daher wird entweder der äußere oder der innere Ring durchlaufen. Bei der *Analyse* wird überprüft, welche Prozesse sich mit MRK bzw. mit dem Programmiersystem abbilden lassen, und daraus abgeleitet, an welchen Stellen das Programmiersystem gegebenenfalls erweitert oder angepasst werden muss. In der *Designphase* werden die Komponenten analysiert, die verwendet werden sollen und die dann in das Informationsmodell integriert werden müssen. Dies hängt vor allem von ihrer Eignung hinsichtlich ihres MRK-Einsatzes ab. Außerdem wird in dieser Phase das unternehmensspezifische Vorgehen für die Anwendung des Systems im Unternehmen ermittelt. Dies beinhaltet insbesondere die Festlegung, welche Rollen für den Einsatz des Programmiersystems definiert werden. Darüber hinaus sollte in dieser Phase die Kommunikation gegenüber den Beteiligten stattfinden. Je nach Ergebnis der Analyse wird in dieser Phase außerdem das Wissensmodell angepasst, was zum Beispiel die Integration weiterer Ressourcen bedeutet. Bei der *Realisierung* wird, im Falle eines erstmaligen Einsatzes einer MRK, die Applikation aufgebaut und die Werkkräfte werden für den Umgang mit der Applikation qualifiziert. In beiden Fällen wird in dieser Phase jedoch das Programmiersystem integriert und die Werkkräfte werden dafür entsprechend geschult. Während der *Betriebsphase* findet eine kontinuierliche Verbesserung statt. Dabei können bspw. neue Anforderungen an die unterstützende Aufgabenzuteilung oder weitere Fügeprozesse abgeleitet werden, die in einem späteren Schritt integriert werden können.

Die Rollen in diesem Vorgehensmodell sind nach NEBDAL (2013) folgende: Projektleiter, Koordination, Umsetzungspartner, Management, Kernteam, Beta-Benutzer, Endbenutzer und Administrator bzw. Experte. Da im Rahmen der Mensch-Roboter-Kooperation der Mensch in das System integriert ist und ein wesentliches Element des Systems die Sicherheit ist, wird als weitere Rolle die Arbeitssicherheit in das Vorgehen einbezogen. Die betriebliche Arbeitssicherheit ist vor allem bei der Definition des unternehmensspezifischen Vorgehens relevant, da dort festgelegt wird, wer die Abnahme der programmierten Applikation übernimmt. Des Weiteren sollte der Betriebsrat als weitere Rolle einbezogen werden, um in jeder Phase mitzuentcheiden, welche Daten zur Verbesserung des Systems aufgenommen werden dürfen oder ob Elemente des Systems ausgeschlossen werden müssen. Nach STOLZENBERG & HEBERLE (2009) sollen Meister und Ausbilder in den Prozess einbezogen werden, da sie die ablaufenden Prozesse kennen und somit einen wertvollen Beitrag liefern können.

### 7.2.1 Orientierung

In der *Orientierungsphase* wird mit Hilfe des angepassten Reifegradmodells in Form von Fragen identifiziert, ob der Einsatz des aufgabenorientierten Programmiersystems für die Mensch-Roboter-Kooperation rentabel für das Unternehmen sein könnte. Setzt das Unternehmen bisher noch keine Mensch-Roboter-Kooperation ein, so wird dies ebenfalls festgestellt. Um zunächst zu erkennen, ob im Unternehmen grundsätzlich Potenzial für eine MRK besteht, können Methoden, wie bspw. nach THOMAS (2017) oder BLANKEMEYER ET AL. (2018), verwendet werden. Geht daraus hervor, dass Potenzial für den Einsatz von MRK besteht, so kann mit dem angepassten Reifegradmodell fortgefahren werden, um in Zusammenhang mit der Einführung einer MRK die Rentabilität des Einsatzes des aufgabenorientierten Programmiersystems abzuschätzen. Die Fragen des angepassten Reifegradmodells umfassen die in folgender Tabelle enthaltenen:

*Tabelle 4: Fragen zur Bestimmung des Reifegrads hinsichtlich des Einsatzes des Programmiersystems*

Frage	Reifegrad
Werden im Unternehmen bereits Leichtbauroboter eingesetzt?	3
Gibt es Experten zur Roboterprogrammierung?	2
Sind die Werkkräfte skeptisch gegenüber dem Einsatz von Robotern?	1
Werden die Aufgaben in der Montage überwiegend manuell gelöst?	3
Besteht eine hohe Variantenvielfalt?	3
Ist es notwendig, die Montageanlage oft umzurüsten?	3
Besteht ein schwankender personalseitiger Kapazitätsbedarf?	3

Zur Bestimmung des Reifegrads beantwortet das Unternehmen die Fragen mit einer ja/nein-Entscheidung. Wird eine positive Antwort gegeben, so wird die Punktzahl des Reifegrads gegeben. Diese Punkte werden nachfolgend aufsummiert.



### 7.2.2 Analyse

Bei der *Analyse* wird, wie auch in den anderen Vorgehensmodellen, geprüft, inwiefern das Programmiersystem für die ausgewählten Montageprozesse einsetzbar ist und welcher Aufwand für eine Erweiterung des Systems anfallen würde. Außerdem werden die ausgewählten Prozesse hinsichtlich ihrer Einsatzfähigkeit für die Mensch-Roboter-Kooperation überprüft. Dabei wird beispielsweise beurteilt, ob die meisten Bauteile Gefährdungspotenziale, wie beispielsweise scharfe Kanten, enthalten, so dass die Umsetzung mit einer Mensch-Roboter-Applikation schwer möglich ist. Während der Analyse wird ein Abgleich durchgeführt, welche Prozesse und Komponenten durch das bereits bestehende System zur Programmierung abgedeckt werden können. Dies bedeutet, dass überprüft wird, welche der Komponenten, z. B. Roboter oder Greifer, bereits im Wissensmodell vorhanden sind. Unter Berücksichtigung der Ergebnisse des Abgleichs wird abgeleitet, an welchen Stellen bzw. für welche Komponenten das Wissensmodell erweitert werden muss. Dieser Schritt erfolgt in der Designphase.

### 7.2.3 Design

Geht aus der Analysephase hervor, dass das Wissensmodell erweitert werden muss, so werden in dieser Phase Ressourcen und Komponenten beschrieben und in das Wissensmodell integriert. Die zu integrierenden Ressourcen und Komponenten, die für das Programmiersystem bzw. das Robotersystem zur Verfügung stehen sollen, werden zunächst einer Prüfung hinsichtlich der Einsatztauglichkeit in einer Mensch-Roboter-Kooperation unterzogen. Damit soll vermieden werden, dass Komponenten, die bereits selbst nicht tauglich sind, in der Anwendung verwendet werden. Dies zielt auf die Integration von Sicherheitselementen ab, wie sie in Abschnitt 6.4 dargelegt wurde. Durch diesen vorgelagerten Prozess wird sichergestellt, dass das Ziel eines geringeren Aufwands für die Risikoanalyse erreicht werden kann. Bei dieser Prüfung der Komponenten wird sowohl deren Gestaltung als auch deren technische Spezifikation betrachtet. Im Hinblick auf die Gestaltung lassen sich in dieser Phase noch Änderungsbedarfe ableiten, die bei der Realisierung eingebracht werden können, so dass die Komponente einsatzfähig wird. Bei den technischen Eigenschaften der Geräte wird überprüft, ob deren Arbeitsbereiche in den Grenzbereichen liegen, so dass sie in einer Mensch-Roboter-Kooperation einsetzbar sind. Für den Fall, dass vorher noch keine MRK eingesetzt wurde, werden in dieser Phase die Komponenten, wie Roboter und Endeffektor, basierend auf den in der

Analyse identifizierten Montageprozessen beschafft. Neue Komponenten sowie in der Analysephase identifizierte Vorgänge werden in das Wissensmodell integriert.

Des Weiteren wird in dieser Phase das unternehmensspezifische Rollenverständnis für die Mensch-Roboter-Kooperation geschaffen. Dies bedeutet, dass im Unternehmen festgelegt wird, welche Aufgaben und Verantwortlichkeiten die Mitarbeitenden im Umgang mit der MRK haben sollen. Um eine Akzeptanz für den Einsatz der Mensch-Roboter-Kooperation zu erhalten, stellen RICHTER ET AL. (2015) fest, dass die Bedürfnisse des Menschen bestmöglich erfüllt sein sollen. Die Bedürfnisse der Mitarbeitenden können je nach Definition der Rollen für den Einsatz der Mensch-Roboter-Kooperation mehr oder weniger erfüllt werden. Um das Rollenverständnis zu schaffen, müssen zunächst folgende drei Fragen beantwortet werden:

- I. Wer soll in der Anwendung die Programmierung der Applikation vornehmen? Ist die Werkskraft in Personalunion die programmierende Person?
- II. Wer übernimmt die unterstützte Risikoanalyse und verleiht der Applikation die CE-Kennzeichnung? Kann dies die Werkkraft tun?
- III. Wie gestalten sich dann der Ablauf des Programmiervorgangs und die Abnahme der Applikation? Wie wird sie in die unternehmensspezifischen Vorgänge integriert?

Die Abbildung 48 zeigt drei verschiedene Möglichkeiten zur Anwendung des Programmiersystems für die MRK im Unternehmen auf und stellt dabei die entsprechenden Vor- und Nachteile und die Bewertung der Erfüllung der Bedürfnisse der Optionen dar. Bei den Bedürfnissen wird überprüft, inwiefern die nach RICHTER ET AL. (2015) beschriebenen Bedürfnisse Kompetenz, Verbundenheit, Autonomie und Abwechslungsreichtum erfüllt werden. Dabei bedeutet Kompetenz, Einfluss auf das Arbeitsergebnis zu haben; Autonomie, dass Entscheidungen selbst getroffen werden können und Abwechslungsreichtum, ein breites Tätigkeitsspektrum zu haben. Die ausgewählte Option muss in einen Geschäftsprozess überführt werden. Von der gewählten Option hängt die Qualifizierung der Mitarbeitenden im nächsten Schritt ab.

Bei der ersten Option ist die Rollenverteilung so gestaltet, dass die Werkkraft die Programmierung durchführt, sich die Simulation ansieht und diese bewertet. Die Arbeitssicherheit als zusätzliche Ressource nimmt anschließend die Applikation ab. Dies bietet den Vorteil, dass einerseits kein Programmierer notwendig ist, aber andererseits die Verantwortung für die Abnahme der Applikation nicht auf die

Werkskraft übertragen wird. Die Bedürfnisse nach Kompetenz, Verbundenheit und Abwechslungsreichtum können hiermit weitestgehend erfüllt werden. Im zweiten Fall werden alle drei Rollen durch unterschiedliche Personen besetzt. Die Programmierung wird durch eine Person durchgeführt, die nicht der Werkkraft entspricht. Des Weiteren erfolgt die Abnahme durch die Arbeitssicherheit. Dabei ist nachteilig, dass alle Rollen teilweise gleichzeitig eingebunden werden müssen und somit ein hoher Aufwand zu verzeichnen ist. Des Weiteren kann die Werkkraft in diesem Fall nicht bei der Zuteilung der Aufgaben mitentscheiden. Dies führt dazu, dass die beiden Bedürfnisse nach Autonomie und Kompetenz kaum erfüllt werden. Im dritten Fall übernimmt die Werkkraft sowohl die Programmierung des Systems als auch die Abnahme der Applikation. Die Verantwortung für die Applikation wird somit an die Werkkraft übertragen. Zugleich trägt dies dazu bei, dass die Werkkraft bei der Aufgabenzuteilung mitentscheiden kann, wodurch die Bedürfnisse nach Autonomie und Kompetenz gut erfüllt werden. Auch trägt dies zum Abwechslungsreichtum bei, da die Werkkraft unterschiedliche Aufgaben zu erfüllen hat. Durch die Übernahme der Aufgaben durch die Werkkraft liegt eine geringe Kapazitätsbindung vor. Die Art der Verteilung der Rollen führt zu unterschiedlichen Anforderungen an die Qualifizierung der Personen, die die Rollen einnehmen. Dieser Bedarf wird in Abschnitt 7.2.4 dargelegt.

## 7 Vorgehen zur Integration des Programmiersystems

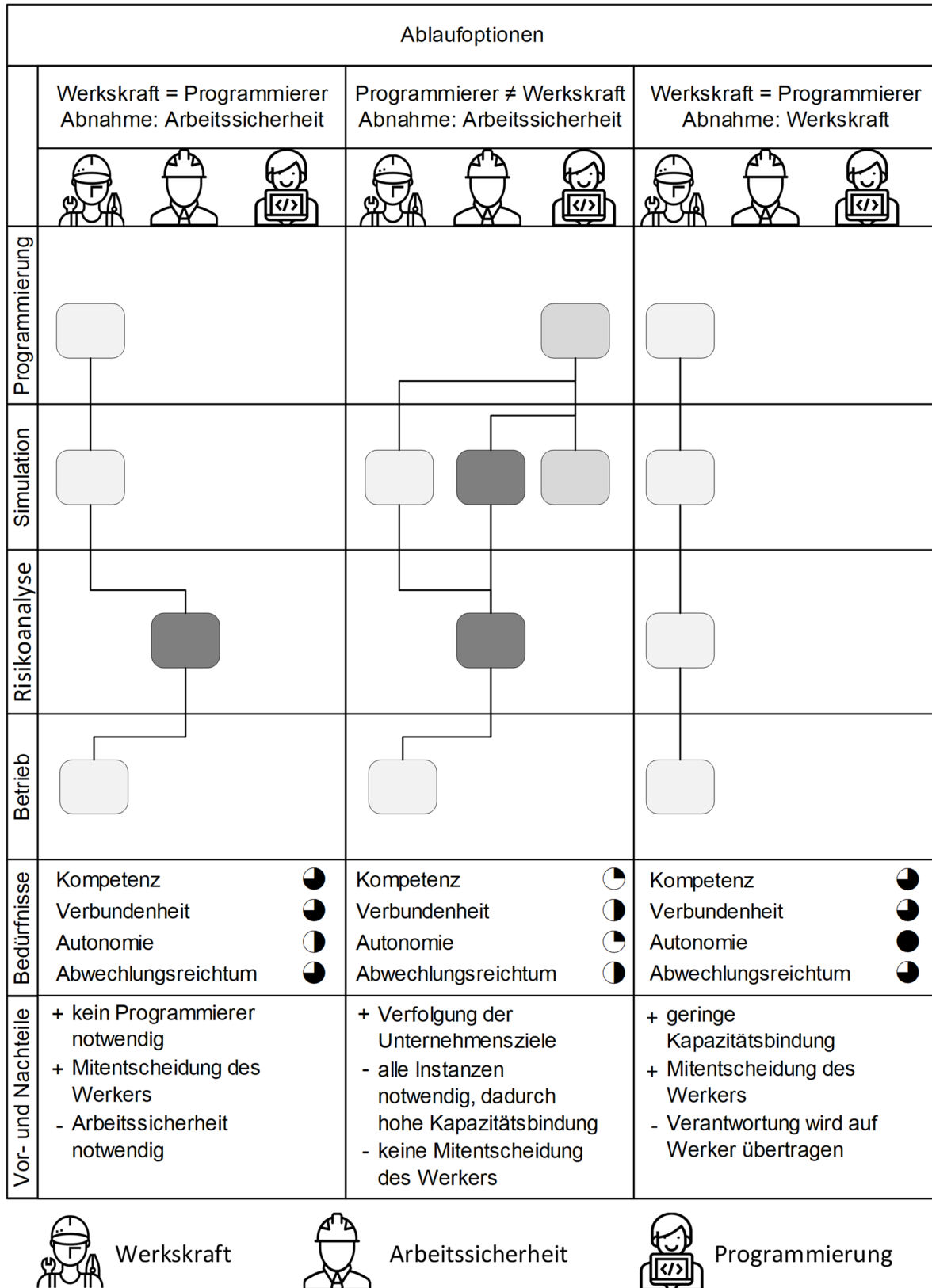


Abbildung 48: Optionen zur Rollenverteilung

### 7.2.4 Realisierung

In der *Realisierungsphase* steht neben der Implementierung des Programmiersystems im Unternehmen bzw. der Implementierung der MRK-Applikation die Qualifizierung des Personals im Fokus. Dabei sollen die Werkkräfte sowohl für die Arbeit in der Mensch-Roboter-Kooperation als auch für den Umgang mit dem Programmiersystem qualifiziert werden.

Die Qualifizierung des Personals im Hinblick auf den Einsatz des aufgabenorientierten Programmiersystems hängt von dem gewählten Ablauf in der Designphase ab. Bei der ersten Option übernimmt die Werkkraft die Aufgabe des Programmierers und entscheidet über die Zuordnung der Aufgaben zu den beiden Ressourcen. Die Abnahme erfolgt durch die Arbeitssicherheit. Dies bedeutet, dass die Werkkraft dafür qualifiziert werden muss nachzuvollziehen, ob bestimmte Aufgaben durch den Roboter durchgeführt werden dürfen, beispielsweise aufgrund von scharfkantigen Bauteilen. Ebenso sollte die Werkkraft sich mit den unterschiedlichen Werkzeugen auskennen.

In der zweiten Option gibt es zusätzlich zu dem Werker eine Person, die die Programmierung des Systems vornimmt. Dies kann beispielsweise ein Vorarbeiter sein. Die Abnahme der Applikation erfolgt durch die Arbeitssicherheit. Bei dieser Konstellation muss die Person, die die Programmierung vornimmt, die Kenntnisse zu einer möglichen Zuteilung der Aufgaben, basierend auf den Randbedingungen, erlangen und das Wissen zu den unterschiedlichen Endeffektoren erhalten. Der Werker muss in diesem Fall die grundlegende Qualifizierung einer MRK-Arbeitskraft durchlaufen.

Bei Option drei übernimmt die Werkkraft alle Tätigkeitsbereiche. In diesem Fall muss die Werkkraft sowohl im Bereich der Grundlagen qualifiziert sein als auch das Programmiersystem beherrschen und außerdem die Qualifizierung für die Abnahme der Applikation vor Inbetriebnahme erhalten.

Für alle Werkkräfte, die an dem MRK-Arbeitsplatz arbeiten, sollte eine Grundqualifizierung durchgeführt und dokumentiert werden. In der Grundqualifizierung müssen die Darstellungen möglicher Gefährdungen bei einer Zusammenarbeit und der Umgang mit dem Roboter enthalten sein. Dazu gehört beispielsweise, dass der Werkkraft bewusst sein muss, dass sie sich bei bewegendem Roboter in dessen Nähe nicht bücken sollte.

Die Zusammenfassung der unterschiedlichen Qualifizierungen für die Optionen enthält nachfolgende Abbildung 49.










Ablaufoptionen			
	Werkskraft = Programmierer Abnahme: Arbeitssicherheit	Programmierer ≠ Werkkraft Abnahme: Arbeitssicherheit	Werkkraft = Programmierer Abnahme: Werkkraft
	  	  	  
Werkkraft	<ul style="list-style-type: none"> <li>• Basisqualifizierung MRK</li> <li>• Programmiersystem</li> <li>• Gefährdungen in der MRK</li> <li>• Endeffektoren</li> </ul>	<ul style="list-style-type: none"> <li>• Basisqualifizierung MRK</li> </ul>	<ul style="list-style-type: none"> <li>• Basisqualifizierung MRK</li> <li>• Programmiersystem</li> <li>• Gefährdungen in der MRK</li> <li>• Endeffektoren</li> <li>• Abnahme der Applikation</li> </ul>
Programmierer	<ul style="list-style-type: none"> <li>• -</li> </ul>	<ul style="list-style-type: none"> <li>• Programmiersystem</li> <li>• Gefährdungen in der MRK</li> <li>• Endeffektoren</li> </ul>	<ul style="list-style-type: none"> <li>• -</li> </ul>
Arbeitssicherheit	<ul style="list-style-type: none"> <li>• Gefährdungen in der MRK</li> <li>• Endeffektoren</li> <li>• Abnahme der Applikation</li> </ul>	<ul style="list-style-type: none"> <li>• Gefährdungen in der MRK</li> <li>• Endeffektoren</li> <li>• Abnahme der Applikation</li> </ul>	<ul style="list-style-type: none"> <li>• -</li> </ul>



Abbildung 49: Bedarf der Qualifizierung je nach Vorgehen

### 7.2.5 Betrieb

In der *Betriebsphase* erfolgen Evaluation, Abnahme, Weiterentwicklung und weitere Nutzungsschulungen (BACKHAUS 2016; NEBDAL 2013). Außerdem sollte eine kontinuierliche Verbesserung des Systems im Unternehmen verfolgt werden. Dabei können einerseits technische Verbesserungen durchgeführt werden, wie z. B. die Anpassung von Roboterbahnen oder die Integration weiterer Prozesse. Andererseits kann das Planungssystem, insbesondere die Szenarioplanung, um Kriterien erweitert werden.

## 8 Umsetzung und Erprobung

Umgesetzt und evaluiert wird das Programmiersystem anhand einer Getriebemontage an einem Montagearbeitsplatz mit einem kollaborierenden Roboter. Der Anwendungsfall umfasst die gemeinsame Montage eines Kleintriebes. Dazu steht ein Arbeitsplatz mit einem Universal Robot, der mit einem elektrischen Zweibackengreifer ausgestattet ist, zur Verfügung. Die Gehäuseteile sowie eine Zahnrad- und Ritzelwelle werden in einem Kleinladungsträger (KLT) an die Montagestation angeliefert. Die O-Ringe, ein Sensor und Dichtungen werden gesammelt in Behältern am Montagearbeitsplatz bereitgestellt. Um die Anforderungen an das Programmiersystem zu identifizieren, wurde dieser Anwendungsfall als MRK-Applikation zunächst ohne Hilfsmittel geplant und der Roboter anschließend dafür programmiert. Die Ausführung dieses Szenarios wurde zunächst ohne Anpassung des Roboters an den Menschen durch Kommunikation über Anstoßen des Roboters ausgeführt. Der Anwendungsfall kann als repräsentativ für die Montage von Produkten an einem Montagearbeitsplatz beschrieben werden, an dem die Bauteile an dem Arbeitsplatz angeliefert werden, so dass der Mensch an seinem Platz verweilen kann.

### 8.1 Versuchsaufbau

Der Versuchsaufbau zur Getriebemontage ist in der nachfolgenden Abbildung 50 dargestellt. Für die Aufgabe steht ein Roboter UR 10 zur Verfügung, der mit einem elektrischen Greifer ausgestattet ist. Dieser kann eine Greifkraft zwischen 1,5 N und 50 N aufbringen und besitzt einen Hub von 50 mm.

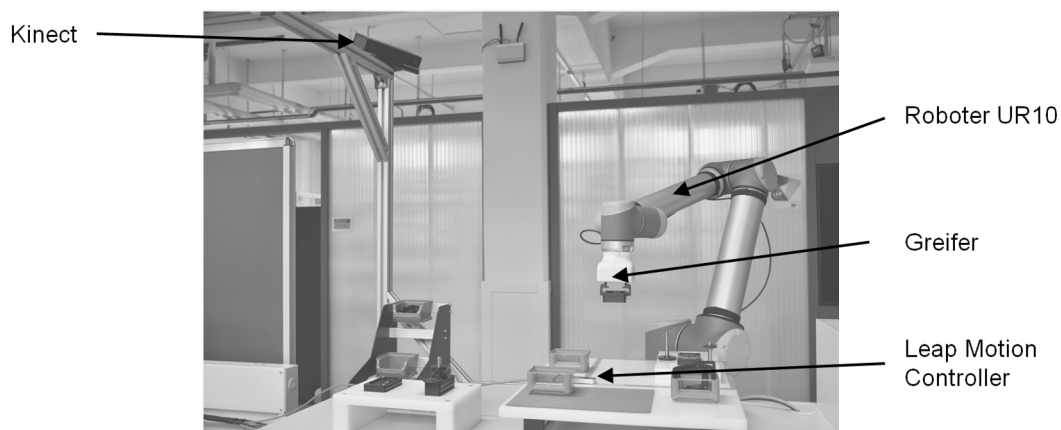


Abbildung 50: Versuchsaufbau zur aufgabenorientierten Programmierung

An dem Ausleger des Montagetisches ist eine Kinect® angebracht, mit der die Bewegungen des Menschen aufgenommen werden können. Außerdem befindet sich im Arbeitsbereich ein Leap Motion Controller, mit dem die Bewegungen der Hände detektiert werden.

Das Kamerasystem Kinect® ist mit Farb- und Tiefensensoren ausgestattet und besitzt ein vertikales Sichtfeld von 60 Grad sowie ein horizontales von 70 Grad. Über die Software kann die Kinect® bis zu 6 Körper erkennen, wobei für jeden Körper Daten zu den einzelnen Gelenken, wie bspw. die Position, abgefragt werden können. Der Leap Motion Controller ist ein Kamerasystem, mit dem eine genaue Detektion von Händen möglich ist. Er ist mit zwei Infrarotkameras und drei Infrarot-LEDs ausgestattet. Der Arbeitsbereich reicht von 25 bis 600 mm über dem Gerät und deckt einen Bereich von 150 Grad ab. Der Leap Motion Controller mit seiner Software liefert Positionen der Hände mit einer Genauigkeit von etwa 0,01 mm (WEICHERT ET AL. 2013). Der Vorteil dieser Komponenten liegt darin, dass beide Systeme Daten zu den Positionen der Gelenke liefern können, die für das Vorgehen zur Tätigkeitserkennung notwendig sind.

### **8.2 Umsetzung des Systems**

Zentraler Bestandteil des Systems ist die Benutzerschnittstelle, die der nutzenden Person die Möglichkeit gibt, mit dem System zu interagieren. Die Inhalte der Benutzerschnittstelle (vgl. Abbildung 51) sind dabei die einzigen für die nutzende Person sichtbaren Inhalte; die restlichen Bestandteile laufen im Hintergrund ab.



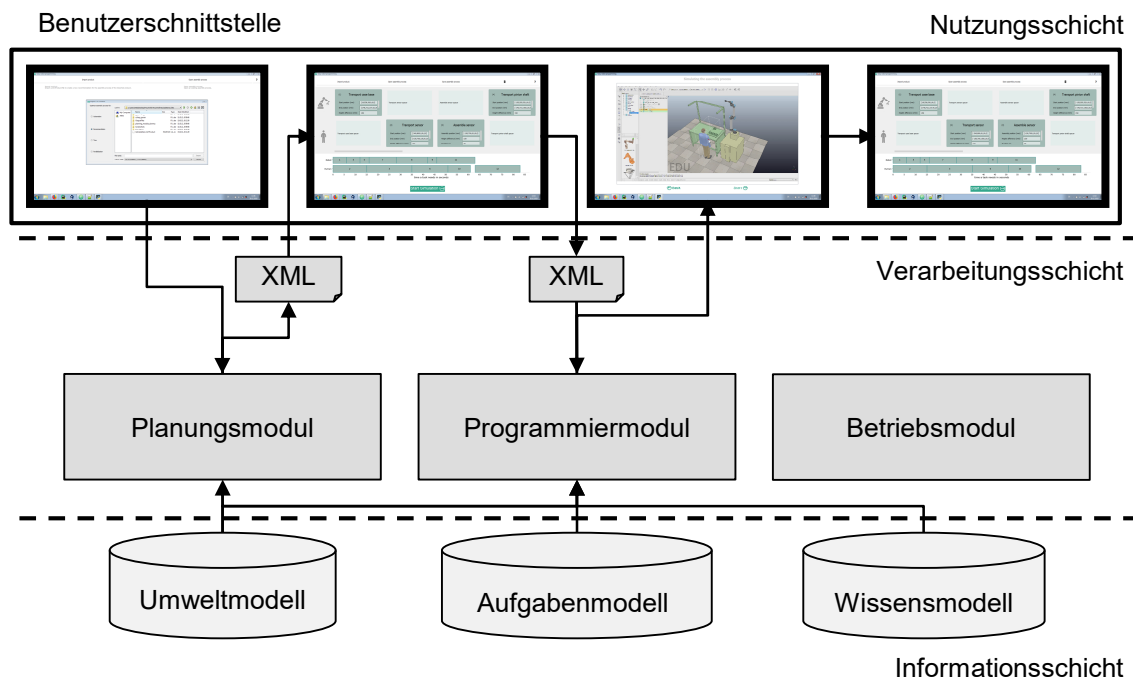


Abbildung 51: Schichten des Programmiersystems und Darstellung der Ausprägungen der Benutzerschnittstelle

Die Benutzerschnittstelle wird über eine Python-Schnittstelle zur Verfügung gestellt. Wie in Abbildung 51 zu erkennen, gibt es verschiedene Darstellungen in der Benutzerschnittstelle je nach Phase des Programmierprozesses.

Die Verarbeitungsschicht besteht aus den Hauptmodulen des Programmiersystems. Zum Austausch von Daten zwischen der Nutzungs- und der Verarbeitungsschicht steht das XML-Format zur Verfügung. Die Funktionen des Planungs- sowie des Betriebsmoduls wurden in Matlab mit Hilfe der Toolbox *Statistics and Machine Learning* umgesetzt. Im Programmiermodul wird für die Konfiguration des Modells zur Tätigkeitserkennung Matlab verwendet. Der Postprozessor zur Erstellung des Roboterprogramms wurde in C++ entwickelt. Die Bestandteile und deren Umsetzung beschreiben die folgenden Abschnitte.

### 8.2.1 Planungsmodul

Über die Benutzerschnittstelle wird es der nutzenden Person ermöglicht, eine Montageaufgabe auf Basis des Produkts bzw. der Baugruppe auszuwählen. Daraufhin wird das Planungsmodul angestoßen, das, wie in Abschnitt 6.1 beschrieben, zunächst die Montagereihenfolge ableitet, sodann einen Fähigkeitenabgleich durchführt und basierend darauf einen Szenariovorschlag erarbeitet.

Aus der CAD-Repräsentation des Produkts bzw. der betrachteten Baugruppe werden im Programm Catia die einzelnen Bauteile und deren Eigenschaften ausgelesen. Durch den in Abschnitt 6.1.1 beschriebenen Assembly-by-Disassembly-Ansatz wird jeweils das mögliche Bauteil demontiert und seine Beschreibungen werden in eine Struktur aufgenommen. Daraus ergibt sich eine Montagereihenfolge, wobei jedem Bauteil ein Prozessschritt zugeordnet ist. Für jedes Bauteil werden für das weitere Vorgehen notwendige Eigenschaften abgeleitet und dem Prozessschritt zugeordnet. Diese sind in Abschnitt 6.1.1 beschrieben.

Unter Beachtung der beschriebenen Aufgabenmodellierung entspricht die Beschreibung eines Montageschritts einem Gesamtzyklus von MTM. Um zu der definierten Beschreibungsebene der Aufgabenblöcke mit der Unterteilung in *Antransport* und *Fügevorgang* zu gelangen, muss dieser abgeleitete Montageschritt zunächst zerlegt werden. Nach der Zerlegung liegen die einzelnen Aufgaben mit ihren Eigenschaften bzgl. Bauteil und Prozess vor. Die Beschreibung des Prozessschrittes *Antransport* des Gehäusebodens mit seinen Eigenschaften ist in nachfolgender Abbildung 52 dargestellt.

Antransport Gehäuseboden	
<b>Bauteil</b>	<b>Prozess</b>
<ul style="list-style-type: none"> <li>• Bauteilname: Gehäuseboden</li> <li>• Gewicht: 2000g</li> <li>• Art: Bauteil</li> <li>• Position Koordinatensystem: [-10 0 0]</li> <li>• Ausdehnung: [96.1600 58 11.6300]</li> <li>• Schwerpunkt: [5.6100 0 -3.5000]</li> <li>• Volumen: 4.2024e+04</li> <li>• Material: ‚Plastik‘</li> <li>• Formstabilität: 1</li> <li>• Scharfkantigkeit: 0</li> </ul>	<ul style="list-style-type: none"> <li>• Prozessart: Antransport</li> <li>• Fügeteil: Gehäusedeckel</li> <li>• Feinfühligkeit: 0</li> <li>• Montagemittel: -</li> <li>• Anziehmoment: -</li> </ul>

Abbildung 52: Darstellung der abgeleiteten Eigenschaften für die Aufgabe *Antransport*

Mit dieser Beschreibung werden die Aufgaben dargestellt, die mit den Fähigkeiten der vorhandenen Ressourcen abgeglichen werden. Dabei wird jede Aufgabe mit den beschriebenen Fähigkeiten von Roboter und Mensch verglichen, um damit die Eignung der Ressource für die jeweilige Aufgabe zu identifizieren. Dazu werden die Werte der einzelnen Kriterien mit den möglichen erreichbaren Eignungsgraden abgeglichen. Somit ergibt sich je Kriterium einer Aufgabe ein Einzeleignungsgrad. Von den Einzeleignungsgraden der Kriterien wird anschließend der Mittelwert gebildet, der den Eignungsgrad der Aufgabe darstellt (vgl. Abbildung 53). Die

empfohlene Ressource beschreibt, welche der beiden Ressourcen die höhere Bewertung erhalten hat. Sind beide Bewertungen gleich, gibt es keine besondere Empfehlung.

	Aufgabe	Bewertung Mensch	Bewertung Roboter	Empfohlene Ressource
1	Antransport Getriebehaelfte_2	4.00	4.00	keine
2	Fuegen Getriebehaelfte_2	3.40	3.80	Roboter
3	Antransport Welle_Lang	4.00	4.00	keine
4	Fuegen Welle_Lang	3.40	3.80	Roboter
5	Antransport Welle_kurz	4.00	4.00	keine
6	Fuegen Welle_kurz	3.40	3.20	Mensch
7	Antransport Sensor	3.50	2.50	Mensch
8	Fuegen Sensor	3.20	2.60	Mensch
9	Antransport Dichtung	3.50	2.50	Mensch
10	Fuegen Dichtung	3.20	2.60	Mensch
11	Antransport Getriebehaelfte_1	4.00	4.00	keine
12	Fuegen Getriebehaelfte_1	3.40	3.80	Roboter

Abbildung 53: Ergebnis der Fähigkeitsbewertung

Orientiert an den Ausprägungen der Kriterien nach ROSS (2002), liegen die Eignungsgrade zwischen 1 und 4, wobei 4 die beste Eignung darstellt. Es wird definiert, dass eine Ressource mit einem Wert von 3,5 für die Aufgabe geeignet ist. Damit wird der nächste Schritt der Szenarioplanung durchgeführt. Wie in Abschnitt 6.1.3 beschrieben, erfolgt die Szenarioplanung mit Hilfe eines genetischen Algorithmus, für den als Optimierungskriterium die drei Optionen zur Verfügung stehen. Das Optimierungskriterium wird durch den Benutzer zu Beginn des Vorgangs auf der Benutzerschnittstelle ausgewählt und entsprechend im Planungsmodul verarbeitet. Auf Basis der bewerteten Aufgaben wird, wie in Abschnitt 6.1.3 erläutert, zunächst eine Population der Größe 50 gebildet. Dabei entstehen 50 Individuen mit unterschiedlicher zeitlicher Zuordnung der Aufgaben zu den Ressourcen. Im nachfolgenden Schritt wird die Fitness der einzelnen Individuen, je nach Auswahl des Optimierungskriteriums, bewertet und daraufhin die Reproduktion durchgeführt. Die durchschnittliche Durchlaufzeit zur Abarbeitung der Aufgaben der ersten Population nach dem ersten Durchlauf lag bei 206 s. Nach 27 Generationen lag der Durchschnitt bei 62 s, was aufzeigt, dass die Durchlaufzeit mit Zunahme der Durchläufe sinkt.

In der nachfolgenden Abbildung 54 ist das Ergebnis der Szenarioplanung für das Optimierungskriterium *minimale Durchlaufzeit* am Beispiel der Getriebemontage dargestellt. Die Ergebnisse der beiden anderen Möglichkeiten und die Zuordnung der Nummern zu den Aufgaben sind im Anhang A6 zu finden.

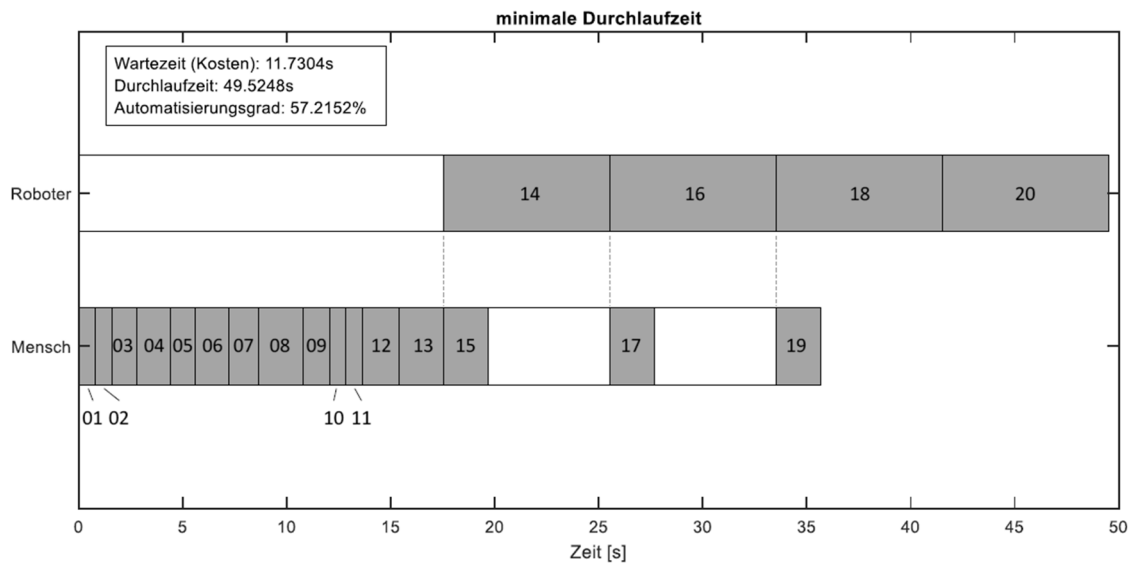


Abbildung 54: Szenario für das Optimierungskriterium minimale Durchlaufzeit für den Anwendungsfall (Nummern kennzeichnen die Aufgaben)

Das Ergebnis der Szenarioplanung und damit die Zuordnung der Aufgaben zu den beiden Ressourcen Mensch und Roboter wird in einer XML-Datei festgehalten, deren Informationen für die Anzeige auf der Benutzerschnittstelle verwendet werden. Die XML-Datei enthält einen Ablauf mit den Aufgaben und der jeweilig zugeordneten Ressource Roboter oder Mensch. Zur zeitlichen Anordnung der Aufgaben auf dem Zeitstrahl wird darüber hinaus die voraussichtliche Startzeit der Aufgabe weitergegeben.

### 8.2.2 Benutzerschnittstelle

Wie zuvor beschrieben, wird das geplante Szenario, das in der XML-Datei festgehalten ist, auf der Benutzerschnittstelle angezeigt und kann dort angepasst und parametrisiert werden. Aus der XML-Datei werden zur Anzeige die Aufgaben und deren Ressource und Startzeit ausgelesen. Die Anzeige der Aufgaben und deren Zuordnung zu Mensch und Roboter auf der Benutzerschnittstelle stellt Abbildung 55 dar.

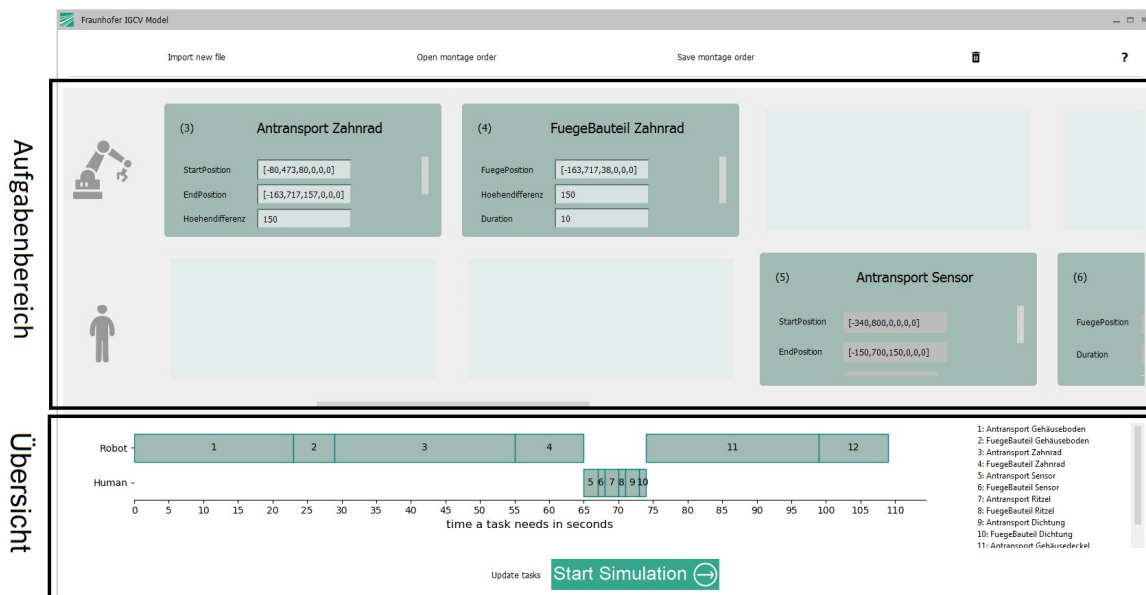


Abbildung 55: Benutzerschnittstelle des Programmiersystems

Es sind dabei ein Aufgabenbereich und ein Übersichtsbereich zu unterscheiden. Im Aufgabenbereich können die Aufgaben zeitlich und zwischen beiden Ressourcen verschoben und die Parameter in den entsprechenden Feldern angepasst werden. Im Übersichtsbereich wird die zeitliche Zuordnung zu Mensch und Roboter der nutzenden Person nochmals im Ganzen angezeigt. Dies ist insbesondere dann relevant, wenn die Anzahl der Aufgaben zu groß wird, so dass im Aufgabenbereich die Lesbarkeit der Aufgaben nicht mehr gewährleistet wäre.

Über die Textfelder erhält die nutzende Person die Möglichkeit zur Eingabe von Parametern. Da auch die Benutzerschnittstelle zur Gewährleistung der Sicherheit beitragen soll, können nur solche Parameter eingegeben werden, die im Rahmen der Sicherheit möglich sind. Zur Überprüfung der Eingabewerte greift die Benutzerschnittstelle auf das Wissensmodell zu. Nach Eingabe aller notwendigen Parameter kann die weitere Verarbeitung zur Generierung des Roboterprogramms gestartet werden. Dazu werden die Informationen über die Aufgaben und deren Zuordnung, wie sie in der Benutzerschnittstelle angegeben sind, in einer XML-Datei gespeichert. In diesem XML werden auch die entsprechenden Abhängigkeiten zwischen Aufgaben, die ein Warten des Roboters auf eine Aufgabe des Menschen erfordern, integriert und an das Programmiermodul weitergegeben.

### 8.2.3 Programmiermodul

Das Programmiermodul ist zuständig für die Generierung des Roboterprogramms und der Konfiguration des Modells für die Tätigkeitserkennung. Die Umsetzung dieser beiden Teilbereiche wird nachfolgend beschrieben.

#### 8.2.3.1 Generierung und Simulation des Roboterprogramms

Das Programmiermodul erhält als Eingangsdaten die XML-Datei mit den Informationen zu den Aufgaben, deren Zuordnung und Abhängigkeiten. Diese Datei wird bei der Generierung des Roboterprogramms einerseits und bei der Konfiguration des Modells zur Tätigkeitserkennung andererseits verarbeitet. Wie in Abschnitt 6.2 beschrieben, wird mit den Informationen ein herstellerspezifisches Roboterprogramm generiert, das nach Prüfung durch die Simulation auf die Robotersteuerung übertragen wird, wo es im Betrieb abläuft. Die Erstellung des Roboterprogramms erfolgt wie in Abschnitt 6.2.1 dargelegt. Dazu wird die XML-Datei mit dem Ablauf eingelese. Anschließend werden aus den Aufgaben die Parameter zur Programmgenerierung, wie bspw. die Greifpositionen, ausgelesen. Das Programmiermodul greift auf das Wissensmodell zu, um die für die Aufgabe passende Roboterbewegung zu identifizieren. Diese Bewegungsbefehle, wie eine lineare Bewegung des Roboters, werden mit den entsprechenden Parametern befüllt, so dass das Programm verwendet werden kann. Zur Ansteuerung des Greifers ist beispielsweise dessen Definition zu Beginn des Roboterprogramms notwendig. Daher wird diese Funktion anhand einer Skript-Funktion an den Anfang des Roboterprogramms gestellt.

Das erstellte Roboterprogramm und die Datei mit dem Ablauf der Aufgaben werden anschließend an die Simulationsgenerierung, die mit Hilfe von Matlab durchgeführt wird, weitergegeben. Wie beschrieben, dient dies zur Prüfung auf Kollisionen, zur Prüfung der Greifvorgänge, aber auch zur Vorbereitung und Prüfung des Ablaufs durch die nutzende Person. Die Simulation des Ablaufs zwischen Mensch und Roboter erfolgt mit Hilfe des Programms v-rep (vgl. Abbildung 56). Dieses Programm bietet den Vorteil, dass sowohl Mensch als auch Roboter mit ihren Bewegungsabläufen für die jeweiligen Aufgaben simuliert werden können. Obwohl die Bewegungsabläufe des integrierten Menschen eher einfach modelliert sind, kann mit der Simulation das Ziel erreicht werden, den prinzipiellen Ablauf zwischen Mensch und Roboter darzustellen.

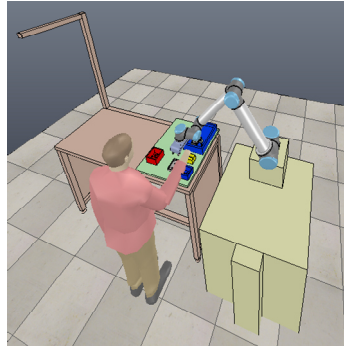


Abbildung 56: Aufbau der Simulation in v-rep

### 8.2.3.2 Konfiguration des Modells zur Tätigkeitserkennung

Neben der Generierung des Roboterprogramms wird im Programmiermodul auch das Modell zur Tätigkeitserkennung konfiguriert. Dies geschieht anhand des in Abschnitt 6.2.2 dargestellten Ablaufs, wobei die Ermittlung der Transitionsmatrizen des oberen und unteren Layers, der Emissionsmatrix des oberen Layers und für die Emissionsmatrix des unteren Layers die Trajektorien- und Sollwinkelberechnung durchgeführt wird. Die Erstellung der Transitionsmatrizen erfolgt nach den Berechnungsvorschriften, wie sie in Abschnitt 6.2.2 und im Anhang A4 erläutert sind.

Zur Bestimmung der Trajektorien und Sollwinkel ist die Kenntnis der Positionen der Speichereinrichtungen und der Montageposition notwendig. Diese werden als Information in der XML-Datei an das Programmiermodul übergeben. Basierend darauf kann die Berechnung der Trajektorien und der Dichtefunktion erfolgen. Die Trajektorien für die Grundbewegungen der Aufgaben im Anwendungsbeispiel sind in Abbildung 57 dargestellt. Für diese Trajektorien wird, wie beschrieben, jeweils der Mittelpunkt berechnet. Dieser wird erhöht und die Trajektorie darin unterteilt, um eine natürliche Handbewegung zu modellieren. Anhand der Trajektorien werden die Dichtefunktionen berechnet. Das Ergebnis dieser Modellierung ist ebenfalls in Abbildung 57 dargestellt.

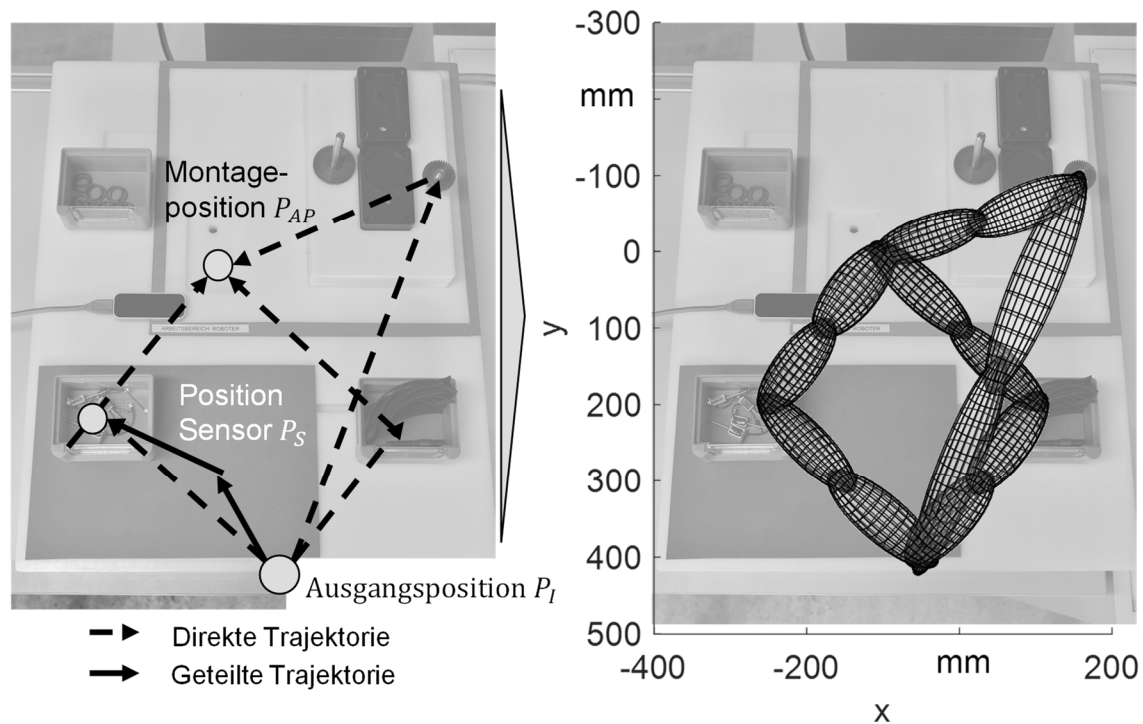


Abbildung 57: Trajektorien zum Antransport von Dichtung und Sensor (links) und Trajektorien mit Gaußscher Verteilung (rechts)

Auf der linken Seite der Abbildung 57 ist der Arbeitsplatz mit den Positionen der Teilebereitstellung und der Montageposition zu erkennen. Um ein Bauteil zu greifen, wird eine Trajektorie zwischen der Ausgangsposition der Hände und bspw. der Position des Behälters der Sensoren abgebildet. Dabei ist der Unterschied zwischen der direkten Verbindung der beiden Punkte und der Unterteilung in zwei Teiltrajektorien aufgezeigt. Auf der rechten Seite sind die gleichen Trajektorien mit den jeweiligen Gaußschen-Verteilungen abgebildet. In der Abbildung 57 (rechts) wird deutlich, weshalb die zusätzliche Information der Bewegungsrichtung relevant für eine robuste Erkennung der Aufgaben ist. Die Wahrscheinlichkeitsräume überlappen sich an einigen Stellen, insbesondere an der Ausgangsposition und der Montageposition.

### 8.2.3.3 Schnittstellen zwischen Robotersteuerung und Modell zur Tätigkeitserkennung

Zur Weitergabe einer erkannten Tätigkeit an die Robotersteuerung muss eine entsprechende Variable auf der Robotersteuerung im Roboterprogramm vorhanden sein und durch die Tätigkeitserkennung umgeschaltet werden. Zur Definition dieser



Variable wird in der XML-Datei eine Aufgabe mit einer entsprechenden Information versehen. Zusätzlich zu dieser XML-Datei mit dem Ablauf wird eine Verwaltung der Registereingänge auf der Robotersteuerung generiert. Diese enthält die Informationen zu den Variablen und den zugehörigen Registereingängen, die bei der Umschaltung einer Variable angesprochen werden müssen. Innerhalb der Tätigkeitserkennung wird zu jeder Aufgabe, die erkannt werden soll, der entsprechende Registereingang im System gespeichert. Mit Hilfe des generierten Roboterprogramms und des konfigurierten Modells zur Tätigkeitserkennung kann das System prinzipiell gestartet werden.

### **8.2.4 Betriebsmodul**

Während des Betriebs arbeiten Mensch und Roboter gemeinsam an der Montage des Getriebes, wobei das Roboterprogramm auf der Robotersteuerung abläuft, die Tätigkeiten des Menschen über Kamerasysteme detektiert und ebenfalls mit Hilfe der Kamerasysteme Kollisionen vermieden werden.

#### **8.2.4.1 Umsetzung der Tätigkeitserkennung**

Mit dem im Programmiersystem konfigurierten Modell zur Tätigkeitserkennung werden die Tätigkeiten über die Positionen der Hände des Menschen detektiert. Dazu werden Kamerasysteme verwendet. Eine Kamera detektiert den gesamten Arbeitsraum, eine zweite Kamera ist für eine genauere Detektion der Hände zuständig. Zur Detektion des gesamten Körpers wird eine Kinect® verwendet und zur Detektion der Hände ein Leap Motion Controller. Befinden sich die Hände im Arbeitsraum des Leap Motion Controllers, so werden diese Daten für eine Detektion verwendet, da diese Kamera genauere Positionsdaten liefern kann als die Kinect®. Diese Positionsdaten werden kontinuierlich in das System eingespeist, wo sie weiterverarbeitet werden. Im Programm werden Positionsdaten der Hände für beide Hände extrahiert sowie die Bewegungsrichtung der Hände identifiziert. Diese Daten sind Basis für die Klassifizierung der Tätigkeit über das Hidden Markov Modell.

Wie erläutert, erfolgt die Tätigkeitserkennung über das zweischichtige Hidden Markov Modell, wobei in der unteren Schicht ein Modell für die Positionsdaten und eines für die Bewegungsrichtung zur Verfügung steht. Über die Bewegung der Hand nahe der Trajektorie in einer bestimmten Richtung kann dann die Tätigkeit detektiert werden. Die Trajektorien werden, wie beschrieben, über die Grundbewegungen bzw. deren Anpassung generiert. Für den Anwendungsfall ergeben sich für die

Bewegungen des Menschen die in nachfolgender Tabelle 5 angegebenen Zeiten basierend auf MTM.

Tabelle 5: Zeiten der Grundbewegungen (1 TMU entspricht 0,036 Sekunden)

Aufgabe	Hin- langen (TMU)	Greifen (TMU)	Bringen (TMU)	Fügen (TMU)	Loslassen (TMU)	Gesamt (TMU)	Gesamt (s)
Antransport Sensor	11,9	7,3	12,4			31,6	1,14
Fügen Sensor				10,4	2,0	12,4	0,45
Antransport Zahnrad	13	2,0	11,1			26,1	0,94
Fügen Zahnrad				16,2	2,0	18,2	0,66
Antransport Dichtung	13,6	7,3	14,4			35,6	1,28
Fügen Dichtung				9,1	2,0	11,1	0,4

Die angegebenen Zeiten sind für die Abschätzung der Dauer der Vorgänge des Fügens und Loslassens relevant, um die Variable in der Robotersteuerung zum korrekten Zeitpunkt umzuschalten. Die zu erkennenden Aufgaben werden im System mit Zuständen verknüpft, wobei jeder Aufgabe ein Zustand zugeordnet ist. Zustand 1 entspricht dem Grundzustand. Der Bewegung *Hinlangen* des Antransports des Sensors beispielsweise wird der Zustand 2 zugeordnet; dem *Bringen*, *Fügen* und *Loslassen* der Zustand 3. Wird eine dieser Bewegungen erkannt, so wechselt das System in den entsprechenden Zustand. Die Erkennung der in Tabelle 5 beschriebenen Aufgaben stellt nachfolgende Abbildung 58 dar.

Es ist zu erkennen, dass sich das System zunächst im Grundzustand (Zustand 1) befindet, bevor das *Hinlangen* beim Antransport des Sensors erkannt wird (Zustand 2). Dies ist dann der Fall, wenn sich die Hand in der modellierten Ellipse um die Trajektorie bewegt.

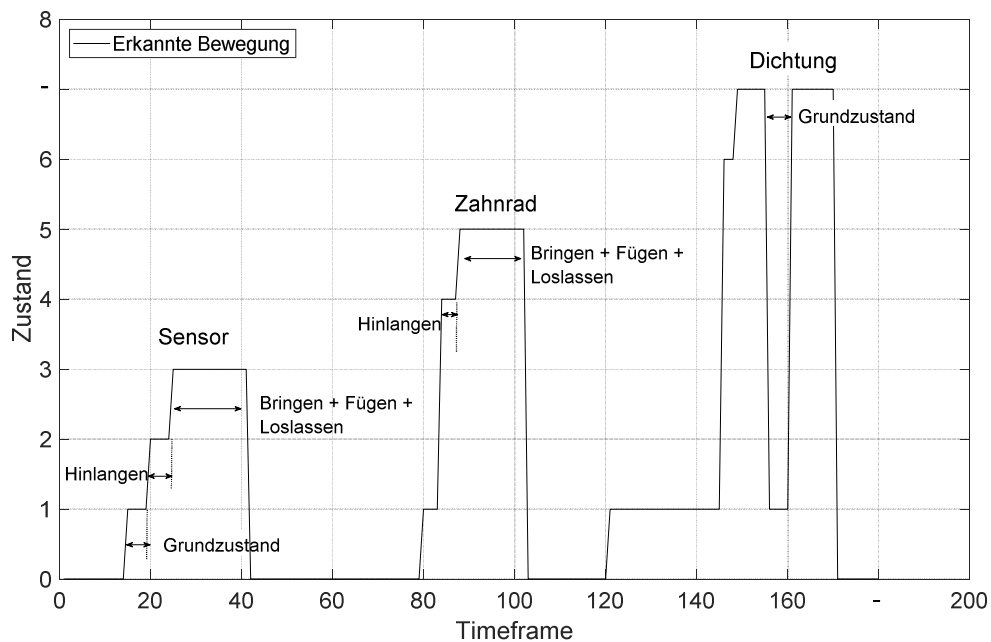


Abbildung 58: Zustandsänderung bei Erkennung der Tätigkeiten

Darauf folgt der Wechsel in den Zustand 3, der das *Bringen*, *Fügen* und *Loslassen* umfasst. Das Greifen, das zwischen beiden Zuständen 2 und 3 erfolgt, kann in der Darstellung entweder im Vorgang *Hinlangen* oder *Bringen* enthalten sein und stellt keinen gesonderten Vorgang dar. Analog zur Erkennung der Aufgaben für den Sensor erfolgt die Erkennung für das Zahnrad und die Dichtung. Im Verlauf der Erkennung der Aufgaben zum Fügen der Dichtung tritt der Grundzustand ein. Dies kann daran liegen, dass eine Aufgabe nicht vollständig erkannt wurde, da die Bewegung von der Bewegungsbahn in der Ellipse abgewichen ist.

Gemäß dem Modell wird bei Erkennung der Bewegung *Hinlangen* die Aufgabe Antransport des Bauteils erkannt. Auf Basis der Erkennung dieser Aufgabe wird die der Aufgabe entsprechende Variable aktiviert und ein Signal an den Roboter gesendet, dass diese Aufgabe erledigt ist. Eine Evaluation zur Überprüfung der Funktion der Tätigkeitserkennung ist in Abschnitt 8.3.2 beschrieben.

#### 8.2.4.2 Umsetzung der Kollisionsvermeidung

Wie konzeptionell erläutert, werden für die Umsetzung der Kollisionsvermeidung Sicherheitsräume um die Gliedmaßen des Menschen und Gefahrenräume um den Endeffektor des Roboters modelliert. Die Größen der Räume können dabei je nach Anwendungsfall und Werkstück eingestellt werden. In diesem Fall wurde für den

Radius des Sicherheitsraums um die Hand ein Durchmesser von 90 mm und für den Quader um den Arm eine Dimension von 220x100x100 mm gewählt. Bei Überschneidung eines Sicherheitsraums mit einem Gefahrenraum pausiert der Roboter bei seiner Bewegung, bis keine Überschneidung mehr vorhanden ist.

Zur Modellierung des Gefahrenraums des Roboters werden über eine Verbindung zum Robotersystem aktuelle Daten zu seinem Status, insbesondere die aktuelle Position des Tool Center Points, ausgelesen. Diese Daten werden verwendet, um den Gefahrenbereich abzubilden.

Zur Modellierung der Sicherheitsbereiche werden, wie bei der Tätigkeitserkennung auch, die Positionsdaten der Kinect® bzw. - wenn sich die Hände im Arbeitsraum der Leap Motion befinden - die Daten der Leap Motion verwendet. Wie in Kapitel 6.3.2 beschrieben, erfolgt mit den modellierten Sicherheits- und Gefahrenräumen die Kollisionsprüfung. Abbildung 59 zeigt die Modellierung der Räume.

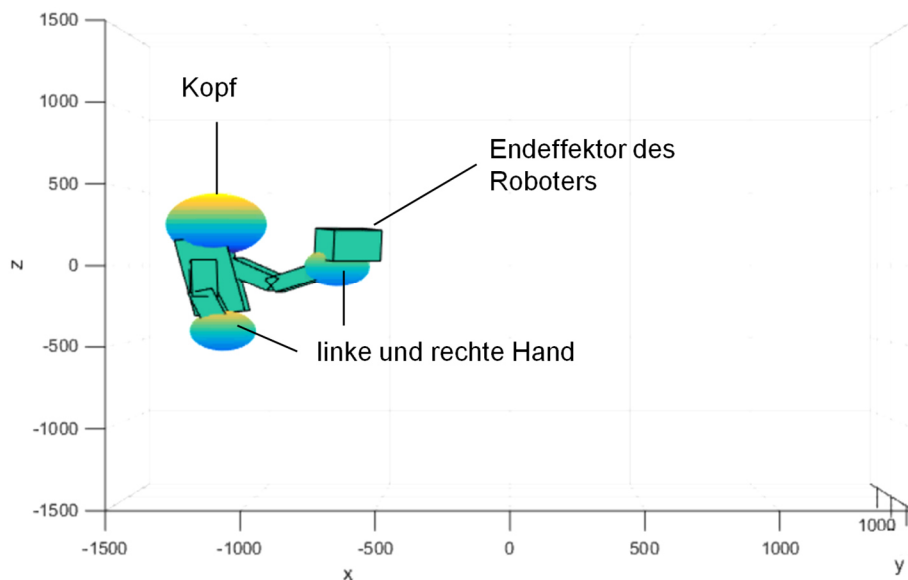


Abbildung 59: Visualisierung der kamerabasierten Kollisionsvermeidung

Die Funktion der Kollisionsprüfung ist ebenso wie die Tätigkeitserkennung mit Matlab umgesetzt. Die Kollisionsvermeidung läuft parallel zur Tätigkeitserkennung und pausiert die Bewegung des Roboters bei Auftritt einer Kollision. Zum Pausieren des Roboterprogramms wird ein Befehl an die Robotersteuerung gesendet. Die Übertragung des Signals erfolgt dabei über eine einfache TCP/IP Verbindung.

Aus der Beschreibung der Kollisionsvermeidung mit Verwendung einer Kinect® bzw. einer Leap Motion wird ersichtlich, dass es sich hierbei nicht um eine

Sicherheitsfunktion nach Normen und Richtlinien handelt. Die Einbindung der Kollisionsvermeidung zeigt vor allem, dass zusätzliche Funktionen zur besseren Zusammenarbeit zwischen Mensch und Roboter integriert werden können bzw. sollten, wenn bereits Kamerasysteme eingesetzt werden.

### **8.3 Erprobung und Evaluation**

Das System wird hinsichtlich zweier Aspekte evaluiert. Dabei ist der erste Aspekt, inwiefern das Programmiersystem über die Benutzerschnittstelle den Planungs- und Programmierprozess unterstützt und vereinfacht. Der zweite Aspekt ist die Zusammenarbeit zwischen Mensch und Roboter mit Unterstützung durch die Tätigkeitserkennung.

Zur Evaluation der Unterstützung beim Programmierprozess wird ein Vergleichsszenario betrachtet, bei dem der Prozess ohne das Programmiersystem durchgeführt wird. Die Schritte hierbei sind:

- 1) Planung des Ablaufs
- 2) Programmierung des Roboters
- 3) Integration von Schnittstellen für die gemeinsame Arbeit
- 4) CE-Kennzeichnung

Der erste Schritt umfasst eine Beschreibung der Montageschritte und die Bewertung, welche der Aufgaben automatisiert werden können und ob dann alle zu automatisierenden Aufgaben durch den Roboter erledigt werden sollen. Für diesen Schritt wird bspw. die Bewertung des wirtschaftlichen Automatisierungsgrades nach ROSS (2002) herangezogen. Für den betrachteten Prozess des Getriebes ergibt sich dadurch, dass alle Operationen außer der Montage des Sensors, der Abdichtung und dem Fügen des zweiten Zahnrades automatisierbar sind. Darauf basierend kann die Aufgabenzuteilung erfolgen. Hierfür sind in der Literatur einige Ansätze zu finden, wie sie in 3.2 beschrieben sind. Je nach Erfahrung und Anzahl betrachteter Kriterien kann sich der Aufwand dabei auf 4-6 Stunden belaufen.

Der Aufwand zur Programmierung des Roboters hängt von der Art des Roboters ab. Die betrachteten Leichtbauroboter für den Einsatz in einer Mensch-Roboter-Kooperation besitzen in vielen Fällen eine grafische Benutzerschnittstelle, bei der die Bewegungen über Apps oder Aufgabenbausteine eingegeben werden können. Für den Fall eines Universal Robot muss für die Bewegung zum Bauteil hin, zum Greifen und für die Bewegung zur Fügeposition jeweils ein Aufgabenbaustein eingefügt und

parametriert werden. Die Eingabe der Positionen kann dabei über das Teach-In-Verfahren oder durch Eingabe der Koordinaten auf der Oberfläche erfolgen. Für die Programmierung der betrachteten Montage des Getriebes kann der Zeitaufwand für die reine Erstellung der Bewegungen mit 2-4 Stunden berechnet werden. In diesem Schritt wurden jedoch nur die Aufgaben des Roboters betrachtet.

Als weiterer Schritt muss anschließend noch die Integration von Schnittstellen für die gemeinsame Arbeit erfolgen. Hierbei ist die Verwendung eines Tasters oder ein physischer Kontakt zum Roboter möglich. Die Integration eines Buttons wird dabei als komplexer angesehen, da hierbei eine Schnittstelle zur Robotersteuerung erstellt werden muss. Eine Option zu physischem Kontakt steht bei einigen Modellen bereits zur Verfügung. Dieser Baustein muss dann an der entsprechenden Stelle des Roboterprogramms integriert und ebenfalls parametriert werden. Für diesen Schritt können je nach Roboter und Option weitere 60 Minuten vorgesehen werden. Nach der Planung und dem Aufbau der Applikation erfolgen die Risikoanalyse und die Ableitung von Maßnahmen zur Verminderung von Gefährdungen. Das Vorgehen hierzu, das in Abschnitt 6.4 beschrieben wurde, kann einige Zeit in Anspruch nehmen. Als Wert werden hierfür 3-5 Stunden angenommen. Insgesamt beläuft sich der gesamte Prozess für die Schritte 1) – 4) daher auf ca. 7-12 Stunden.

### **8.3.1 Evaluation der Benutzerschnittstelle**

Zur Überprüfung der Anwendbarkeit der Benutzerschnittstelle wurde ein Evaluationsbogen erstellt, mit dem Probanden befragt wurden. Der Evaluationsbogen (siehe Anhang A8) eignet sich als Methode, da damit sowohl eine quantitative Messung durchgeführt werden kann als auch spezifische Fragen zum Systemdesign gestellt werden können. Der Evaluationsbogen orientiert sich an der ISO-Norm 9241 10 sowie an Fragebögen des Kompetenzzentrums Usability (KOMPETENZZENTRUM USABILITY MITTELSTAND 2017). Die Fragen der Evaluation können in fünf Bereiche eingeteilt werden. Diese Bereiche werden nachfolgend kurz beschrieben:

- **Selbstbeschreibung / Steuerbarkeit:** Dieses Kriterium befasst sich damit, inwiefern die Inhalte der Oberfläche zu verstehen sind, ohne dass weitere Beschreibungen vorliegen.
- **Fehlertoleranz:** Hier wird bewertet, wie das System mit fehlerhaften Eingaben des Nutzers umgeht.

- Erwartungskonformität bzw. Lernförderlichkeit: Mit diesem Kriterium wird beurteilt, ob das System der Erwartung entspricht und ob der Umgang mit dem System leicht zu erlernen ist.
- Eignung für Arbeitsaufgabe: Damit wird abgefragt, ob die Auswahl des Designs für die Art der durchzuführenden Aufgabe angemessen ist.
- Benutzerfreundlichkeit bzgl. Werteänderung von Aufgaben: In diesem Bereich wurde überprüft, wie die Nutzer den Umgang mit Änderungen von Werten in den Aufgaben bewerten.

Das Ergebnis der Evaluation zeigt die nachfolgende Abbildung 60 auf. Dabei wurden die Angaben in den einzelnen Bereichen auf einer Skala von eins bis fünf angegeben, wobei eins den schlechtesten und fünf den besten Wert bedeutet.

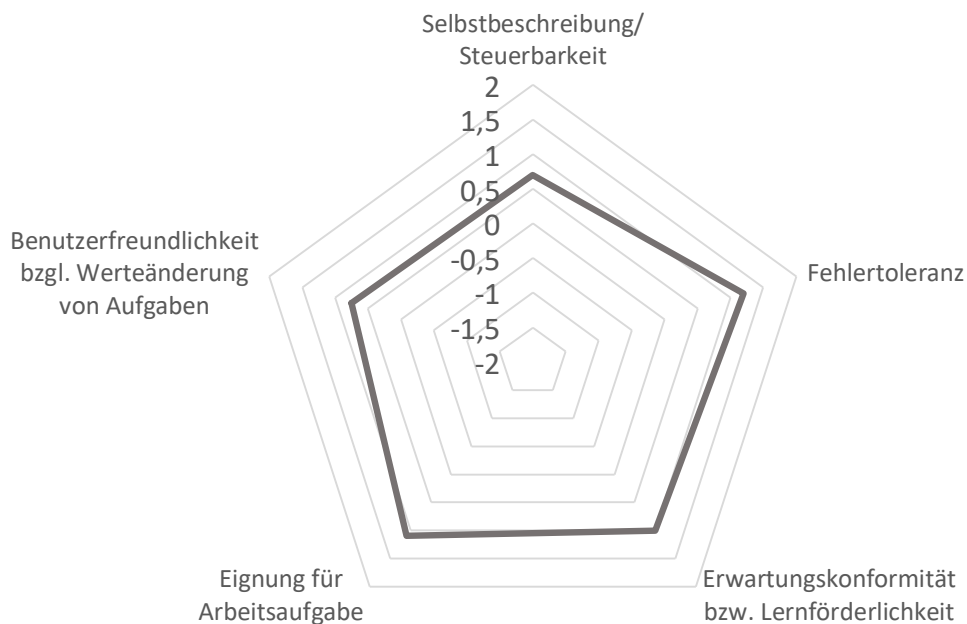


Abbildung 60: Ergebnis der Evaluation der Benutzerschnittstelle

Es ist festzustellen, dass die Benutzerschnittstelle als angemessen bewertet wurde, jedoch auch noch Verbesserungsbedarfe in den einzelnen Bereichen vorhanden sind. Diese sind beispielsweise die Integration von Hilfestellungen, die Verbesserung der Scroll-Eigenschaften im Aufgabenbereich und das Umgestalten der Funktion zum Hinzufügen von Aufgaben.

### 8.3.2 Evaluation der Tätigkeitserkennung

Die Tätigkeitserkennung kann auf ihre Funktion hin untersucht werden, inwieweit Tätigkeiten erkannt werden und wie robust die Rückspiegelung zum Roboter

funktioniert. Zur Evaluation der Tätigkeitserkennung bezüglich ihrer Funktion zur Erkennung der Tätigkeiten wurde die Getriebemontage mit mehreren Personen unterschiedlicher Statur insgesamt 40 Mal durchgeführt. Die Ergebnisse der Durchführung sind in nachfolgender Tabelle 6 enthalten.

*Tabelle 6: Ergebnis der Versuche zur Tätigkeitserkennung*

<b>Aufgabe</b>	<b>Anzahl Ausführungen</b>	<b>Anzahl Fehler</b>	<b>Erkennungsquote</b>
Antransport Sensor	40	3	92,5 %
Antransport Zahnrad	40	4	90 %
Antransport Dichtung	40	6	85 %
<b>Gesamt</b>	<b>120</b>	<b>13</b>	<b>89 %</b>

Als Fehler wurde gewertet, wenn eine Aufgabe nicht oder falsch erkannt wurde oder wenn die Erkennung nicht ausgegeben wurde. Die Tabelle zeigt, dass die Erkennungsquote der einzelnen Aufgaben bei insgesamt 89 % liegt. Die Erkennungsquote vollständiger Durchgänge, das heißt, die Erkennung der drei Tätigkeiten in einem Durchlauf liegt bei 78 %. Dieser Wert erhöht sich um 6 %, wenn die Durchgänge, bei denen Grundbewegungen zwar erkannt wurden, jedoch nicht ausgegeben wurden, als erkannt gewertet werden.

In ihrer Arbeit zur Erkennung von Tätigkeiten an einem PC-Arbeitsplatz erreichen PERDIKIS ET AL. (2008) eine Erkennungsquote von 93 %. Daher ist festzustellen, dass der Wert von 89 % in einem ähnlichen Bereich liegt. Allerdings muss angemerkt werden, dass die Modelldaten im Fall von PERDIKIS ET AL. (2008) über Trainingsdaten bestimmt wurden. Auf diese Weise liegt die Erkennung zwar geringfügig höher, jedoch ist der Aufwand zur Konfiguration und zum Training des Modells erheblich höher. Obwohl die Tätigkeiten erkannt werden konnten, muss der Prozentsatz zur Erkennung noch erhöht werden, um das System industriell einsetzen zu können.



## 9 Wirtschaftliche und Technische Bewertung

### 9.1 Technische Bewertung

Nach Darlegung des Konzepts und der Erprobung des konzipierten Programmiersystems für die Mensch-Roboter-Kooperation soll dieses nachfolgend diskutiert werden.

Wie in der Zielstellung beschrieben, ermöglicht das aufgabenorientierte Programmiersystem eine vereinfachte Programmierung eines Roboters ohne Kenntnisse der jeweiligen Roboterprogrammiersprache, da als Angaben lediglich die Aufgaben zu beschreiben sind. Die Herausforderung der Zuteilung der Aufgaben zu den Ressourcen Mensch und Roboter wurde dahingehend für den Nutzer vereinfacht, dass dieser einen Vorschlag für eine Zuteilung erhält. Um die nutzende Person bzw. auch die Person, die mit dem Roboter gemeinsam arbeitet, entsprechend in den Vorgang zur Aufgabenzuteilung einzubeziehen, besteht die Möglichkeit, die vorgeschlagene Aufgabenzuteilung anzupassen. Nach Erstellung des Roboterprogramms ermöglicht die integrierte Tätigkeitserkennung im Betrieb, dass die Tätigkeiten der arbeitenden Person detektiert werden und an den Roboter weitergegeben werden, so dass kein Signal durch die Person selbst an den Roboter gegeben werden muss. Hierbei liegt ein besonderes Augenmerk darauf, dass die Tätigkeitserkennung während des Programmierprozesses parametrierbar wird, so dass das Training für dieses Modell entfallen und die Tätigkeitserkennung unmittelbar genutzt werden kann. Im Bezug zu einem Einsatz für die Mensch-Roboter-Kooperation bietet das System Vorteile durch die Betrachtung der Sicherheit während des Programmierprozesses und der unterstützten Risikobeurteilung. Da diese Aspekte derzeit die größten Hemmnisse zum Einsatz einer MRK darstellen, kann damit eine Unterstützung erreicht werden. Dies wiederum hat Einfluss auf die flexiblere Einsetzbarkeit der Mensch-Roboter-Kooperation, deren Potenzial bisher noch nicht ausgenutzt werden kann.

Aus technischer Sicht bietet das System die Möglichkeit, für verschiedene Robotertypen verwendet zu werden. Die Programmerstellung des Roboters erfolgt basierend auf den Aufgaben. Für die Nutzung weiterer Roboter müssten die Daten der Verfahrbefehle in dem Wissensmodell hinterlegt werden. Das Wissensmodell ist so aufgebaut, dass es durch einen Prozessexperten erweiterbar ist.

Die Beschreibung der Aufgaben auf der gewählten Ebene bietet den Vorteil, dass sie abstrakt ist und mehrere Grundbewegungen zusammenfasst, um möglichst wenig

Aufgabenblöcke auf der Oberfläche definieren zu müssen. Auf diese Weise ist sie für die meisten Montageaufgaben einsetzbar, bei denen Fügebauteile an ein Basisbauteil montiert werden, wobei sich die Fügebauteile bereits im nahen Umfeld des Arbeitsbereichs befinden. Daher eignet sich diese Beschreibung und Art der Programmierung für Montagearbeitsplätze, an denen ein Produkt oder eine Baugruppe montiert wird. Bei den Fügevorgängen können einfache Fügevorgänge abgebildet werden. Zur Anwendung des Programmiersystems ist es von Vorteil, wenn ein Montagearbeitsplatz mit einem Roboter, der für unterschiedliche Montageaufgaben verwendet wird, besteht. Dies ist vor allem relevant bei Betrachtung der Simulation, für die ein entsprechendes Umweltmodell vorhanden sein muss. Da das Programmiersystem noch keine Funktion zur Generierung eines Umweltmodells bietet, muss dieses erstellt und dem System zur Verfügung gestellt werden. Bei Änderung der Umgebung erfordert dieser Vorgang zusätzlichen Aufwand. Die Ableitung der Montagereihenfolge und der damit verbundenen Beschreibung der Aufgaben erfolgt in dem umgesetzten System mit Hilfe von Catia. Zur breiteren Verwendung des Programmiersystems sollte eine Erweiterung für andere CAD-Programme angestrebt werden.

Im Betriebsmodul können über eine Tätigkeitserkennung die Bewegungen des Menschen erkannt und ein Signal an den Roboter gegeben werden. Dies bietet den Vorteil, dass kein Taster integriert werden muss und die Werkkraft so auch keine zusätzlichen Aufgaben durchführen muss. Diese Erkennung wird mit Hilfe von Kamerasystemen durchgeführt. Mit diesen wird für den Betrieb auch die Kollisionsvermeidung umgesetzt, wodurch der Roboter bei Eintritt einer Werkkraft in den Gefahrenbereich seine Bewegung pausiert. Hierbei ist festzustellen, dass beide Funktionen mit den verwendeten Komponenten umsetzbar sind. Jedoch sind diese Kamerasysteme nicht industriell einsetzbar und müssten daher in der Industrie durch andere ersetzt werden.

Wie in der Evaluation beschrieben, wurde überprüft, wie einfach das Programmiersystem zu bedienen ist und wie häufig die Tätigkeitserkennung auch tatsächlich die menschlichen Tätigkeiten erkennt. Bezüglich der Tätigkeitserkennung war festzustellen, dass 89 % der Tätigkeiten korrekt erkannt werden konnten. Außerdem ermöglicht die automatische Konfiguration des Tätigkeitserkennungsmodells einen flexibleren Einsatz und kann im Rahmen des Programmierprozesses in einem anderen Anwendungsfall eingesetzt werden. Für einen industriellen Einsatz sollte das System noch robuster gestaltet werden, so dass die Tätigkeiten zuverlässig erkannt werden.

## 9.2 Wirtschaftliche Bewertung

Ziel der hier beschriebenen wirtschaftlichen Betrachtung ist es, beispielhaft darzustellen, inwiefern der Einsatz des konzipierten aufgabenorientierten Programmiersystems Auswirkungen auf die Kosten der Programmierung hat. Dabei können gewisse Aspekte in die Bewertung einbezogen werden, jedoch kann nicht alles monetär betrachtet werden. So soll bei der Bewertung der Fokus auf den Aufwand zur Programmierung und weniger auf den monetären Aspekt des Montageablaufs, wie z. B. der Unterschied des zeitlichen Ablaufs durch die Szenarien oder die Tätigkeitserkennung, gelegt werden.

Bei der Wirtschaftlichkeitsberechnung wird daher der Gewinn betrachtet, der durch die Anwendung des aufgabenorientierten Programmiersystems erzielt werden kann. Die Wirtschaftlichkeit wird beeinflusst durch die Zeiteinsparung bei Nutzung des Programmiersystems gegenüber einem Vorgehen zur Planung und Programmierung einer Applikation ohne Unterstützung. Da mit dem Einsatz des Programmiersystems für die MRK auch eine unterstützte Abnahme der Applikation angewandt werden sollte, fließt auch diese Zeitersparnis in die Berechnung ein. Ziel der Wirtschaftlichkeitsbetrachtung ist es zu prüfen, wann der Einsatz des Programmiersystems für Unternehmen rentabel ist. Dies bezieht sich vor allem auf die Anzahl der Umprogrammierungsvorgänge, da diese den Aufwand für die Programmierung und Applikationsplanung darstellen. Wie in Abschnitt 8.3 beschrieben, setzt sich der gesamte Prozess zur Applikationserstellung aus den vier genannten Punkten zusammen. Da diese abhängig von der Anzahl der Aufgaben des betrachteten Szenarios sind, wird bei der Wirtschaftlichkeitsbetrachtung zwischen Applikationen mit wenigen und vielen Aufgaben unterschieden. Aufgaben stellen bspw. der Antransport eines Bauteils oder ein Fügevorgang dar. Für beide Fälle wird für die einzelnen Phasen eine Zeitdauer angenommen. Diese sind in folgender Tabelle 7 zusammengefasst. Zur Schaffung einer Vergleichsbasis wurde das Beispiel der Getriebemontage betrachtet, wobei jeweils eine Version mit weniger und mehr Bauteilen verwendet wurde. In der verringerten Version wurden lediglich der Prozess des Fügens des zweiten Zahnrads, die Montage des Gehäusedeckels und das Verschrauben analysiert. Die Planung des Ablaufs umfasst die Analyse der Montageaufgabe, die Beschreibung der Aufgaben und die Zuteilung zu den Ressourcen. Mit der Integration von Schnittstellen wird die Kommunikation von vollendeten Aufgaben seitens des Menschen hin zur Robotersteuerung ermöglicht. Für diese Schnittstellen ist im Roboterprogramm ein Bedingungssteil einzufügen. Bei der CE-Kennzeichnung wird die Risikoanalyse durchgeführt und mögliche

Gefährdungen werden gemindert. Dazu müssen die einzelnen Aufgaben und der Arbeitsplatz betrachtet werden.

*Tabelle 7: Aufwand der einzelnen Phasen abhängig von der Anzahl der Aufgaben*

<b>Aufwand der Phasen in h</b>		
<b>Phase</b>	<b>Anzahl der Aufgaben</b>	<b>Aufwand [h]</b>
Planung des Ablaufs	5 Aufgaben	4
	15 Aufgaben	6
Programmierung des Roboters	5 Aufgaben	2
	15 Aufgaben	4
Integration von Schnittstellen	2 Schnittstellen	1
	5 Schnittstellen	2
CE-Kennzeichnung	5 Aufgaben	3
	15 Aufgaben	5
<b>Summe Aufwand</b>	5 Aufgaben	10
	15 Aufgaben	17

Zur Berechnung der Wirtschaftlichkeit wurde die Kapitalwertmethode herangezogen, deren Vorgehensweise in WÖHE & DÖRING (2010) beschrieben ist. Zur Ermittlung des Kapitalwerts sind Investitionskosten notwendig. Für diese wurden Werte geschätzt. Bei den Anpassungen des Programmiersystems handelt es sich um Anpassungen des Wissensmodells, wie beispielsweise die Aufnahme weiterer Ressourcen. Je nach Bedarf zur Erweiterung können hier höhere Kosten anfallen. Bei den Aufwänden zur Integration werden Kosten berechnet, die notwendig sind, um das System auf der Hardware zu integrieren. Des Weiteren fallen Kosten für die Qualifizierung des Personals an. Diese Kosten fasst Tabelle 8 zusammen.

Neben der unterschiedlichen Anzahl der betrachteten Aufgaben wird unterschieden zwischen der Anzahl an notwendigen Anpassungen der Applikation, z. B. aufgrund wechselnder Produkte oder Varianten. Die Berechnung wird für folgende drei Fälle durchgeführt:

- 10 Änderungen im Jahr
- 100 Änderungen im Jahr
- 200 Änderungen im Jahr

Die Anzahl von 200 Änderungen pro Jahr stellt dar, dass der Roboter als Werkzeug von einer Werkkraft eingesetzt wird und damit täglich eine neue Applikation erstellt werden muss.

Tabelle 8: Investition für das Programmiersystem

Investitionen		
Kosten des Programmiersystems	20.000	€
Anpassungen des Programmiersystems	5.000	€
Aufwände zur Integration	10.000	€
Aufwände zur Qualifizierung	10.000	€
<b>Summe Invest</b>	<b>45.000</b>	<b>€</b>

Bei der Kapitalwertmethode werden die jährlichen Einnahmen den Ausgaben gegenübergestellt. Die jährlichen Ausgaben setzen sich aus der Wartung und Erweiterung des Systems, bspw. bei Integration eines neuen Greifers, der Qualifizierung weiterer Personen und dem zusätzlichen Aufwand im Engineering zusammen. Damit ergeben sich für das Beispiel von 10 Änderungen im Jahr bei 5 Aufgaben folgende, in Tabelle 9 dargestellten, Werte.

Tabelle 9: Jährliche Kosten des Programmiersystems

Jährliche Kosten			
Wartung	$W_j = W \cdot I$	6.750,00 €	€ / Jahr
Erweiterung des Systems	$E$	100	h / Jahr
Qualifizierung weiterer Personen	$Q_j$	3.000,00 €	€ / Jahr
Zusätzlicher Aufwand im Engineering	$A_{eng}$	2.000,00 €	€ / Jahr
Jährliche Kosten	$K_j = W_j + Q_j + A_{eng} + E$	11.750,00 €	€ / Jahr
$W_j$ =jährliche Wartungskosten, $W$ =Instandhaltungssatz, $I$ = Invest, $E$ =Aufwand zur Erweiterung des Systems, $Q_j$ =Qualifizierungskosten, $A_{eng}$ =Aufwand im Engineering, $K_j$ =jährliche Kosten			

Zur Berechnung der jährlichen Wartungskosten  $W_j$  wird ein Instandhaltungssatz  $W$  von 15 % angenommen, wie beispielsweise bei WÖHE & DÖRING (2010) beschrieben. Zur Erweiterung  $E$  des Systems, beispielsweise zur Integration neuer Komponenten, wird eine Summe von 100 Stunden angenommen. Diese ist abhängig von der Anzahl der Produktvarianten im Unternehmen. Des Weiteren müssten jährlich weitere Mitarbeitende qualifiziert werden ( $Q_j$ ). Im Engineering kann zur Nutzung des Programmiersystems zusätzlicher Aufwand entstehen, z. B. zur Verbesserung der CAD-Konstruktion von Produkten. Die Summe der beschriebenen Kosten ergibt die jährlich anfallenden Kosten  $K_j$ .

Diesen Kosten gegenüber stehen die Einsparungen, die sich aus der Differenz der Kosten bei klassischem Vorgehen und bei Anwendung des Programmiersystems ergeben, wie sie in Tabelle 10 enthalten sind.

Tabelle 10: Jährliche Einsparungen bei Nutzung des Programmiersystems

<b>Jährliche Einsparungen</b>			
Aufwand zur Applikationserstellung mit klassischem Vorgehen	$t_A$	100	h / Jahr
Einsparung mit Programmiersystem	$d$	65	%
Stundensatz Werkskraft	$S_{MA}$	50	€
Jährliche Einsparungen	$E_j = (t_A \cdot S_{MA}) \cdot d$	3.250	€ / Jahr

Die jährlichen Einsparungen ergeben sich durch die Reduzierung der zeitlichen Aufwände zur Planung, Programmierung und Abnahme der Applikation. Da das Programmiersystem sowohl in dem Bereich der Planung als auch der Abnahme der Applikation unterstützt, wird die zeitliche Einsparung bei 65 % angesetzt.

Die Anzahl der Stunden zur Erstellung von MRK-Applikationen hängt von der Anzahl der Änderungen der Applikation ab. Dazu wird der in Tabelle 7 dargestellte Aufwand mit der Anzahl der Änderungen multipliziert. Für 10 Änderungen bei 5 Aufgaben ergibt sich daher der Wert von 100 Stunden. Im Falle von 100 Änderungen bei 5 Aufgaben bedeutet dies einen Aufwand von 1000 Stunden.

Mit der Gegenüberstellung der jährlichen Kosten und der jährlichen Einsparungen ergeben sich für die unterschiedlichen Fälle die in nachfolgendem Diagramm dargestellten Verläufe des Kapitalwerts.

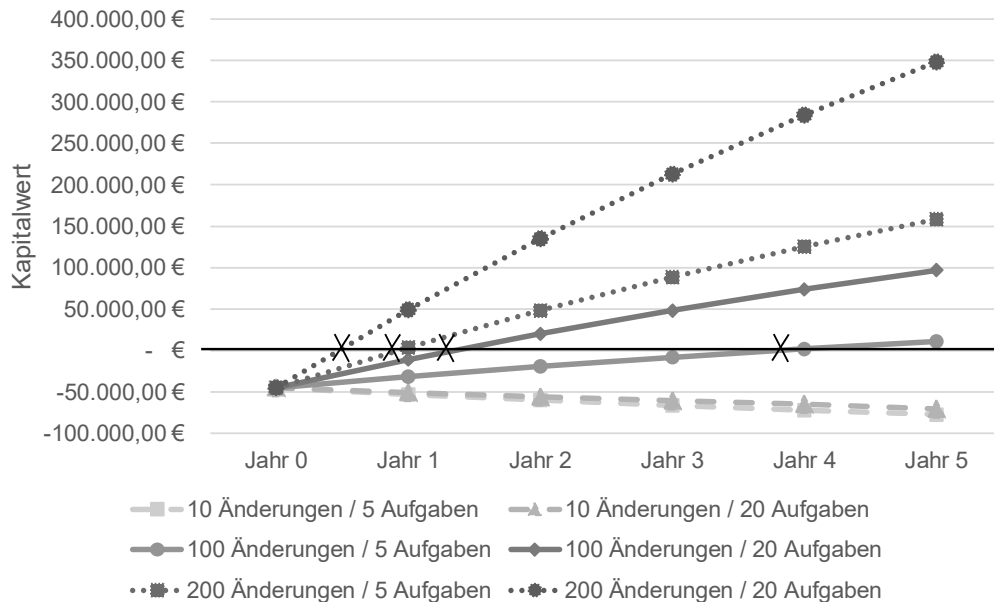


Abbildung 61: Vergleich des Kapitalwerts bei unterschiedlicher Anzahl an Änderungen

Daraus wird ersichtlich, dass der Einsatz des Programmiersystems bei einer kleinen Anzahl von Applikationsänderungen bzw. -anpassungen nicht rentabel ist. Bei einer größeren Anzahl jedoch, wie bei dem Fall der 100 Änderungen in einem Jahr, liegt der Kapitalwert etwa ab dem zweiten Jahr und bei 200 Änderungen bereits nach kurzer Zeit im positiven Bereich. Damit kann aufgezeigt werden, dass das Programmiersystem insbesondere für Firmen geeignet ist, die die Applikation aufgrund von Produktänderungen oft anpassen müssen. Eine Amortisation für die Fälle von 200 Änderungen sowie für 100 Änderungen bei 20 Aufgaben kann bis spätestens 2 Jahre nach Implementierung erwartet werden.

Zur Betrachtung der Wirtschaftlichkeit wurden Vergleichswerte zur Abschätzung des Aufwands in den einzelnen Phasen sowie Annahmen zur Höhe des Invests bestimmt. Da der Aufwand zur Planung und Implementierung einer Applikation jeweils vom Anwendungsfall und dem Robotersystem abhängig ist, dient diese Rechnung als Orientierung. Die angegebenen Annahmen für die Programmierung basieren auf der benötigten Zeit zur Implementierung; die für die sonstigen Positionen auf Expertengesprächen. Für den Anwendungsfall sind dies für Experten konservative und für Nicht-Experten optimistische Angaben. Da jedoch von einem Lerneffekt

auszugehen ist, wurde ein Mittelwert gewählt. Des Weiteren können die Aufwände, die während des Betriebs des Systems anfallen, wie die Erweiterung des Programmiersystems, der Aufwand im Engineering, aber auch die Qualifizierung des Personals, nur abgeschätzt werden. Dadurch könnten sich im realen Einsatz des Systems Verschiebungen der Wirtschaftlichkeit ergeben.



## 10 Zusammenfassung und Ausblick

### 10.1 Zusammenfassung

Aufgrund des Bedarfs nach flexiblen Automatisierungslösungen und den Entwicklungen in der Robotik, die es durch integrierte Sensorik erlauben, dass Roboter schutzzaunlos in der Nähe des Menschen arbeiten können, erhält die Mensch-Roboter-Kooperation vermehrt Beachtung. Dies gilt insbesondere für den Einsatz in der Montage. Dabei wird die Montageaufgabe in kleinere Aufgaben zerlegt und basierend auf den Stärken und Schwächen von Mensch und Roboter auf die beiden Ressourcen aufgeteilt. Um die Applikation letztendlich flexibel einsetzen zu können, muss das System programmiert werden. Da in kleinen und mittelständischen Unternehmen weiterhin nur selten Experten zur Roboterprogrammierung zur Verfügung stehen, besteht ein Bedarf an Systemen zur vereinfachten Programmierung. Im Hinblick auf die Zusammenarbeit zwischen Mensch und Roboter ergeben sich für die Programmierung neue Herausforderungen aufgrund der Abhängigkeiten zwischen den Ressourcen. Bei Beendigung einer Aufgabe durch den Menschen muss dem Roboter dies gemeldet werden. Daher sollte bereits bei der Programmierung in Betracht gezogen werden, wie diese Abhängigkeiten im Programm und im Betrieb abgebildet werden können.

Mit dieser Fragestellung befasst sich die vorgestellte Arbeit, in der ein aufgabenorientiertes Programmiersystem für die Mensch-Roboter-Kooperation beschrieben ist. Das Programmiersystem umfasst neben einem Programmiermodul ein Planungs- sowie ein Betriebsmodul. Im Planungsmodul wird basierend auf der CAD-Abbildung des Produkts ein Szenario erarbeitet, das beschreibt, wie die Aufgaben zeitlich zu den Ressourcen zugeordnet werden können. Dazu wird zunächst der Montageplan durch eine Disassembly-Methode generiert, wobei die einzelnen Aufgaben und deren zugehörige Bauteile beschrieben werden. Diese Aufgaben werden anschließend einer Fähigkeitsbewertung unterzogen, wobei geprüft wird, welche Ressource besser für die Aufgabe geeignet ist. Die bewerteten Aufgaben sind dann die Basis für einen genetischen Algorithmus, mit dem ein Szenario entwickelt wird, welche Ressource eine Aufgabe übernehmen könnte.

Dieser Vorschlag wird einer nutzenden Person auf der Benutzerschnittstelle dargestellt und kann von der Person entsprechend ihrer Anforderungen angepasst und parametrisiert werden. Die Aufgaben werden jeweils als Block auf der Benutzerschnittstelle angezeigt und können über Drag & Drop verschoben werden.

Basierend auf dieser Zuteilung der Aufgaben wird der Ablauf für die Mensch-Roboter-Kooperation erstellt und an das Programmiersystem weitergegeben. Zwischen einzelnen Aufgaben bestehen Abhängigkeiten, da nach dem Montageplan teilweise Zwangsbedingungen zwischen den Aufgaben auftreten. Liegt eine solche Abhängigkeit zwischen zwei Aufgaben vor, wird dies bereits durch einen Einschub in dem Ablauf für die Mensch-Roboter-Kooperation integriert und im Programmiersystem verarbeitet.

Im Programmiermodul wird mit Hilfe dieser Informationen das Roboterprogramm erstellt, wobei auf das Wissensmodell zugegriffen wird. Für die einzelnen Aufgaben wird der Robotercode entsprechend aufgenommen und mit den Parametern versehen, die in dem Ablauf angegeben sind. Die Abhängigkeiten werden als Bedingung vor eine Aufgabe eingefügt und mit einem Input-Signal versehen, das beschreibt, dass die vorherige Aufgabe bearbeitet wurde. Um dieses Signal zu erhalten, gibt es auf Seite des Menschen eine Tätigkeitserkennung. Kamerabasiert werden die menschlichen Bewegungen beobachtet und daraus mit Hilfe von Hidden Markov Modellen wahrscheinlichkeitsbasiert die Tätigkeiten erkannt. Das Hidden Markov Modell besteht aus zwei Schichten, wobei in der unteren Schicht die Grundbewegungen Hinlangen und Bringen und in der oberen Schicht die Aufgaben abgebildet werden. Zur Erkennung der Grundbewegungen werden zwischen Ausgangsposition und Zielposition Trajektorien generiert, die mit einer Gaußschen-Verteilung versehen werden. Bewegt sich die Hand entlang einer Trajektorie, so besteht eine hohe Wahrscheinlichkeit, dass diese Grundbewegung durchgeführt wird. Dazu werden die Positionsänderungen der Hände über zwei Kamerasysteme (Kinect® und Leap Motion Controller) aufgenommen. Durch dieses kamerabasierte System können Druckknöpfe zur Signalweitergabe durch den Menschen entfallen. Das System wurde anhand eines beispielhaften Anwendungsfalls, der Montage eines kleinen Getriebes, umgesetzt, wozu ein Aufbau mit einem Universal Robot 10 gewählt wurde.

In den Anforderungen wurde beschrieben, dass das System zur Programmierung die Mitarbeitenden bei der Planung unterstützen soll, jedoch gleichzeitig die Werkskraft noch mitentscheiden kann. Außerdem sollten Schnittstellen für den Betrieb bereits bei der Programmierung integriert werden und im Betrieb eine Antizipation des Roboters für eine flüssigere Zusammenarbeit erreicht werden. Mit Hilfe des Planungsmoduls kann ein Szenario zur Zusammenarbeit abgeleitet werden, das die Werkkraft entweder so nutzen oder anpassen kann. Damit wurde der Anforderung genügt, die Werkkraft zu unterstützen, aber auch die Mitentscheidung zu fördern. Die Schnittstellen für den Betrieb werden aus den Abhängigkeiten der Aufgaben

abgeleitet und als Bedingung in das Roboterprogramm eingefügt. Dadurch müssen diese nicht manuell integriert werden. Zur Identifikation der Schnittstelle wurde für den Betrieb ein Modell zur Tätigkeitserkennung basierend auf Hidden Markov Modellen angepasst und verwendet, um die durchgeführten Tätigkeiten des Menschen zu erkennen und ohne Taster das Signal an den Roboter übergeben zu können. Durch die Integration von Sicherheitselementen in die verschiedenen Module des Programmiersystems und die unterstützte Risikobeurteilung kann der Prozess zur Erlangung der CE-Konformität verkürzt werden.

Insgesamt ist festzustellen, dass das Programmiersystem für den Anwendungsfall konzeptioniert und umgesetzt wurde. Die Ableitung eines Szenarios kann durchgeführt und auf der Benutzerschnittstelle angezeigt und angepasst werden. Auch die Programmgenerierung konnte umgesetzt werden, so dass ein Roboterprogramm generiert wird. Die Tätigkeitserkennung im Betrieb konnte mit einer Erkennungsquote von 89 % implementiert werden. In allen Bereichen können jedoch noch Verbesserungsbedarfe abgeleitet werden, so dass das System für weitere Anwendungsfälle, weitere Ressourcen und andere Layouts verwendet werden kann. Im nachfolgenden Abschnitt wird auf mögliche Erweiterungen eingegangen.

## **10.2 Ausblick**

Das erarbeitete Programmiersystem kann in unterschiedlicher Hinsicht in allen drei Modulen und in dem Bereich der Sicherheit erweitert werden.

Im Planungsmodul ist zunächst eine Erweiterung der Ableitung der Montagereihenfolge möglich, so dass ein Montagevorranggraph entsteht. Durch die Nutzung dieser Informationen des Montagevorranggraphen stehen dem Roboter in den weiteren Modulen, wie der Programmierung oder im Betrieb, eine Auswahlmöglichkeit an Aufgaben zur Verfügung. Durch die Integration einer Layoutplanung in den Fähigkeitenabgleich können unterschiedliche Layouts überprüft und eine optimale Arbeitsplatzkonfiguration in Zusammenhang mit der Aufgabenzuteilung ermittelt werden. Mit Hilfe einer Analyse der Gewohnheiten verschiedener Nutzer bei Benutzung der Schnittstelle oder während des Betriebs ist eine Integration dieser Informationen in der nächsten Planung möglich, sodass die optimale Aufteilung von Aufgaben für einen jeweiligen Nutzer entsteht. Ein weiterer Aspekt ist die Eingabe von Aufgaben ohne die Durchführung der Montageableitung. Hierbei ist zu ermitteln, inwiefern es zielführend sein kann, zusätzliche Aufgaben eingeben zu können.

Im Bereich der Programmierung des Roboters könnten statt einer definierten Bewegung unterschiedliche Trajektorien hinterlegt werden, die im Betrieb entsprechend der aktuellen Tätigkeit des Menschen oder dessen Position zur Laufzeit ausgewählt werden. Ähnlich dazu könnte eine Erweiterung die Abbildung unterschiedlicher Montagereihenfolgen sein. Dies erhöht die Komplexität der Abbildung der Schnittstellen zwischen den Aufgaben, gibt jedoch dem System die Möglichkeit, auf eine ungeplante Reihenfolgeänderung durch den Menschen zu reagieren. Damit einhergehend ist die Erweiterung der Tätigkeitserkennung. Zu den jeweils bekannten Aufgaben, die über die Oberfläche beschrieben werden, könnte eine Integration von allgemeineren Tätigkeiten, die im normalen Arbeitsablauf nicht direkt abgebildet sind, erfolgen. Derzeit bildet der Grundzustand diese ab. Eine genauere Erkennung dieser Tätigkeit führt jedoch zu erweiterten Informationen über den aktuellen Montageablauf. Basierend auf der Evaluation der Benutzerschnittstelle kann diese entsprechend verbessert werden. Eine erneute Durchführung der Evaluation der Schnittstelle nach deren Anpassung deckt weitere Verbesserungspotenziale auf.

Insgesamt ist es denkbar, das System für eine Mensch-Roboter-Kollaboration zu erweitern. In diesem Fall arbeiten Mensch und Roboter gemeinsam an einer Aufgabe und am gleichen Produkt. Ein relevanter Fall wäre das Anreichen eines schweren Gegenstands durch den Roboter und ein anschließendes gemeinsames Fügen durch Mensch und Roboter. Um diesen Aspekt abzubilden, sind Erweiterungen im Planungsmodul notwendig, da die gemeinsame Aufgabe entsprechend eingeplant werden muss. Dies muss ebenso auf der Benutzerschnittstelle abgebildet werden und bewirkt Änderungen in der Programmierung.

Im Bereich der Sicherheitsbetrachtung bewirkt eine genauere Simulation die Detektion sicherheitskritischer Abläufe. Außerdem könnte in die Simulation die Berechnung der Druck- und Kraftwerte integriert werden, so dass eine Messung dieser bei der Risikoanalyse nicht notwendig ist. Die Definition des Vorgehens zur Risikobeurteilung unterliegt den Gremien zur Standardisierung. Es ist insbesondere ein Vorgehen dafür zu schaffen, wie mit Änderungen an der Applikation umgegangen werden muss, so dass ein flexiblerer Einsatz ermöglicht wird.

## 11 Literaturverzeichnis

ABB 2018

ABB Robotics (Hrsg.): Robot Studio.

<<https://new.abb.com/products/robotics/robotstudio>> 31.08.2018.

ABELE & REINHART 2011

Abele, Eberhard & Reinhart, Gunther: Zukunft der Produktion. München: Carl Hanser Fachbuchverlag, 2011. ISBN: 9783446428058.

AKELLA ET AL. 1999

Akella, P.; Peshkin, M.; Colgate, E.; Wannasuphoprasit, W.; Nagesh, N.; Wells, J.; Holland, S.; Pearson, T.; Peacock, B.: Cobots for the automobile assembly line in Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C). Detroit, MI, USA. 10-15 May 1999, 1999. S.728–733

ALBU-SCHÄFFER ET AL. 2007

Albu-Schäffer, Alin; Haddadin, Sami; Ott, Christian; Stemmer, Andreas; Wimböck, Thomas; Hirzinger, Gerd: The DLR lightweight robot in Industrial Robot: An International Journal. Vol. 34 5, 2007. S.376–385

ALZAHRANI & KAMMOUN 2016

Alzahrani, Mona & Kammoun, Salma: Human-activity-recognition-challenges-and-process-stages- in International Journal of Innovative Research in Computer and Communication Engineering. Vol. 4 5, 2016. S.1111–1118

ARAI ET AL. 2000

Arai, T.; Sugi, M., Aiyama, Y.; Ota, J.: Holonic Robot System for Assembly in Intelligent autonomous systems 6. Amsterdam: IOS Press, 2000. S.371–378

ARTIMINDS ROBOTICS GMBH 2019

ArtiMinds Robotics GmbH (Hrsg.): Robot Programming Suite.

<<https://www.artiminds.com/de/artiminds-rps/>> 07.09.2019.

AWAIS & HENRICH 2010

Awais, Muhammad & Henrich, Dominik: Human-robot collaboration by intention recognition using probabilistic state machines in 2010 IEEE 19<sup>th</sup> International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010). Balatonf red , Hungary, 2010. S.75–80

BACKHAUS 2016

Backhaus, Julian: Adaptierbares aufgabenorientiertes Programmiersystem für Montagesysteme. München: Herbert Utz Verlag, 2016. ISBN: 978-3-8316-4570-1.

BASHIR ET AL. 2007

Bashir, F. I.; Khokhar, A. A.; Schonfeld, D.: Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models in IEEE Transactions on Image Processing. Vol. 16 7, 2007. S.1912–1919

BEHRENS & ELKMANN 2014

Behrens, R. & Elkmann, N.: Study on meaningful and verified Thresholds for minimizing the consequences of human-robot collisions in 2014 IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, China. 31.05.2014 - 07.06.2014, 2014. S.3378–3383

BERG & REINHART 2017a

Berg, Julia & Reinhart, Gunther: An Integrated Planning and Programming System for Human-Robot-Cooperation in Procedia CIRP. Vol. 63, 2017. S.95–100

BERG ET AL. 2017b

Berg, Julia; Richter, Christoph; Reinhart, Gunther: Aufgabenorientierte Programmierung für die Mensch-Roboter-Kooperation in VDI Automation 2017, 2017.

BERG ET AL. 2018a

Berg, Julia; Reckordt, Tim; Richter, Christoph; Reinhart, Gunther: Action Recognition in Assembly for Human-Robot-Cooperation using Hidden Markov Models in Procedia CIRP. Tianjin, China. 10. - 12. Mai, 2018. S.205–210

BERG ET AL. 2018b

Berg, Julia; Richter, Christoph; Reinhart, Gunther: Integration of Safety elements into task-oriented programming system for human-robot-collaboration in Proceedings of ISR 2018: 50<sup>th</sup> International Symposium on Robotics. München. 20.– 21. Juni, 2018. S.281–284

BERUFGENOSSENSCHAFT HOLZ UND METALL 2011

Berufsgenossenschaft Holz und Metall: Obere Gliedmaßen - Sicherheitsabstände, 2011 <[https://www.bghm.de/uploads/tx\\_ttproducts/datasheet/BG\\_10.4.2.3\\_04.pdf](https://www.bghm.de/uploads/tx_ttproducts/datasheet/BG_10.4.2.3_04.pdf)> 07.06.2019.

BEUMELBURG 2005

Beumelburg, Katharina: Fähigkeitsorientierte Montageablaufplanung in der direkten Mensch-Roboter-Kooperation. Heimsheim: Jost-Jetter, 2005.  
ISBN: 3- 936947 - 52- X.

BLANKEMEYER ET AL. 2018

Blankemeyer, Sebastian; Recker, Tobias; Stuke, Tobias; Brokmann, Jens; Geese,

Markus; Reiniger, Michael; Pischke, Dennis; Oubari, Assem; Raatz, Annika: A Method to Distinguish Potential Workplaces for Human-Robot Collaboration in Procedia CIRP. Tianjin, China. 10. - 12. Mai, 2018. S.171–176

BORTOT ET AL. 2010

Bortot, Dino; Ding, Hao; Günzkofer, Fabian: Effizienzsteigerung durch die Bewegungsanalyse und -modellierung der Mensch-Roboter-Kooperationen in zfa 64, 2010. S.65–75

BUBB ET AL. 2006

Bubb, Heiner; Engstler, Florian; Fritzsche, Florian; Mergl, Christian; Sabbah, Olaf; Schaefer, Peter; Zacher, Iris: The development of RAMSIS in past and future as an example for the cooperation between industry and university in International Journal of Human Factors Modelling and Simulation. Vol. 1 1, 2006. S.44–52

BULLING ET AL. 2014

Bulling, Andreas; Blanke, Ulf; Schiele, Bernt: A tutorial on human activity recognition using body-worn inertial sensors in ACM Computing Surveys. Vol. 46 3, 2014. S.1–33

BULLINGER & AMMER 1986

Bullinger, Hans-Jörg & Ammer, Dieter: Systematische Montageplanung. München: Hanser, 1986. ISBN: 978-3446146068.

BUNDESMINISTERIUM FÜR ARBEIT UND SOZIALES 2015

Bundesministerium für Arbeit und Soziales: Interpretationspapier zum Thema „Wesentliche Veränderung von Maschinen“ in Gemeinsames Ministerialblatt, 2015. S.183–186

BUTZ & KRÜGER 2014

Butz, Andreas & Krüger, Antonio: Mensch-Maschine-Interaktion. München: De Gruyter Oldenbourg, 2014. ISBN: 978-3-486-71967-3.

CHOI & IP 1999

Choi, C. K. & Ip, W. H.: A comparison of MTM and RTM in Work Study. Vol. 48 2, 1999. S.57–61

COPPELIA ROBOTICS 2019

Coppelia Robotics <<http://www.coppeliarobotics.com/index.html>> 22.08.2019.

CRAIG 2005

Craig, John J.: Introduction to robotics. 3<sup>rd</sup> ed. Upper Saddle River, N.J: Pearson/Prentice Hall, 2005. ISBN: 0201543613.

DEHAIS ET AL. 2011

Dehais, Frédéric; Sisbot, Emrah Akin; Alami, Rachid; Causse, Mickaël: Physiological and subjective evaluation of a human-robot object hand-over task in *Applied ergonomics*. Vol. 42 6, 2011. S.785–791

DELANG ET AL. 2017

Delang, Kathleen; Bdiwi, Mohamad; Harsch, Ann-Kathrin: Evaluation and selection of workstations for an application of Human-Robot-Interaction (HRI) in manufacturing in *International Conference on Intelligent Robots and Systems (IROS)*. Vancouver. 24.–28. September, 2017.

DEUSE ET AL. 2014

Deuse, Jochen; Roßmann, Jürgen; Kuhlenkötter, Bernd; Hengstebeck, André; Stern, Oliver; Klöckner, Maik: A Methodology for the Planning and Implementation of Service Robotics in Industrial Work Processes in *Procedia CIRP*. Dresden. 13. - 14. November, 2014. S.41–46

DEUTSCHLÄNDER 1989

Deutschländer, Arthur: *Integrierte rechnerunterstützte Montageplanung*. München: Hanser, 1989. ISBN: 3446157794.

DGUV 2015

DGUV: Industrie 4.0 - kollaborierende Roboter in *Fachzeitschrift für Prävention, Rehabilitation und Entschädigung* 3, 2015. S.10–13

DIN ISO 10218-1

DIN ISO 10218-1: Industrieroboter - Sicherheitsanforderungen - Teil 1: Roboter 2012

DIN ISO 10218-2

DIN ISO 10218-2: Industrieroboter - Sicherheitsanforderungen - Teil 2: Robotersysteme und Integration 2012

DIN SPEC 91329:2016-01

DIN SPEC 91329:2016-01: Erweiterung des EPCIS-Ereignismodells um aggregierte Produktionsereignisse zur Verwendung in betrieblichen Informationssystemen.

DRAG AND BOT GMBH

drag and bot GmbH (Hrsg.): drag&bot. <<https://www.dragandbot.com/product/>> 13.08.2019.



DRAGAN ET AL. 2015

Dragan, Anca D.; Bauman, Shira; Forlizzi, Jodi; Srinivasa, Siddhartha S.: Effects of Robot Motion on Human-Robot Collaboration in Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction - HRI '15. Portland, Oregon, USA. 02.03.2015 - 05.03.2015, 2015. S.51–58

DRAKE 1967

Drake, Alvin W.: Fundamentals of applied probability theory. New York: McGraw-Hill, 1967. ISBN: 978-0070178151.

DRÖDER ET AL. 2018

Dröder, Klaus; Bobka, Paul; Germann, Tomas; Gabriel, Felix; Dietrich, Franz: A Machine Learning-Enhanced Digital Twin Approach for Human-Robot-Collaboration in Procedia CIRP. Tianjin, China. 10. - 12. Mai, 2018. S.187–192

ECKSTEIN 1999

Eckstein, Jens: Echtzeitfähige Kollisionserkennung für Virtual Reality Anwendungen. München: Utz, 1999.

EHRMANN 2007

Ehrmann, Markus: Beitrag zur Effizienzsteigerung bei der Programmierung flexibler, roboterbasierter Montagezellen. Kaiserslautern: Techn. Univ., Pak, 2007. ISBN: 9783939432388.

ENG ET AL. 1999

Eng, Tiam-Hock; Ling, Zhi-Kui; Olson, Walter; McLean, Chuck: Feature-based assembly modeling and sequence generation in Computers & Industrial Engineering. Vol. 36 1, 1999. S.17–33

ENGESER 2014

Engeser, Anselm: Verwendung von geometrischen Hüllkörpern in einem Präprozessor Framework für numerische Simulationen, 2014  
<[http://www.cie.bgu.tum.de/publications/bachelorthesis/2014\\_Engeser.pdf](http://www.cie.bgu.tum.de/publications/bachelorthesis/2014_Engeser.pdf)>  
21.08.2019.

FABER ET AL. 2015

Faber, Marco; Bützler, Jennifer; Schlick, Christopher M.: Human-robot Cooperation in Future Production Systems in 6<sup>th</sup> International Conference on Applied Human Factors and Ergonomics (AHFE 2015). Las Vegas, Nevada, USA. 26. - 30. Juli, 2015. S.510–517

FACHBEREICH HOLZ UND METALL DER DGUV 2019

Fachbereich Holz und Metall der DGUV <<https://www.dguv.de/medien/fb->

- holzundmetall/publikationen-dokumente/infoblaetter/infobl\_deutsch/080\_roboter.pdf>. 06.01.2020
- FANUC DEUTSCHLAND GMBH 2019  
FANUC Deutschland GmbH (Hrsg.): Kollaborierende Roboter CR-7iA and CR-7iA/L. <<https://www.fanuc.eu/de/de/roboter/roboterfilter-seite/kollaborierende-roboter/collaborative-cr7ial>> 10.10.2019
- FELDMANN 1992  
Feldmann, Klaus: Montageplanung in CIM. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. ISBN: 978-3-540-53240-8.
- FISCHER-STABEL & GOLLMER 2016  
Fischer-Stabel, Peter & Gollmer, Klaus-Uwe: Informatik für Ingenieure. Konstanz und München: UVK Verlagsgesellschaft mbH, 2016. ISBN: 3825246450.
- FRANKA EMIKA GMBH  
Franka Emika GmbH <<https://www.franka.de/>> 10.08.2018.
- FRAUNHOFER IAO 2016  
Fraunhofer IAO: Leichtbauroboter in der manuellen Montage. Stuttgart, 2016.
- FRUTIG 2013  
Frutig, Markus: (Leichtbau-)Roboter sind im kommen in technica. Vol. 2013 6/7, 2013. S.44–49
- GANSCHAR & GERLACH 2013  
Ganschar, Oliver & Gerlach, Stefan: Produktionsarbeit der Zukunft - Industrie 4.0. Stuttgart: Fraunhofer-Verl., 2013. ISBN: 978-3-8396-0570-7.
- GEISBERGER & BROY 2012  
Geisberger, Eva & Broy, Manfred: agendaCPS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-29098-5.
- GERKE 2015  
Gerke, Wolfgang: Technische Assistenzsysteme. Berlin: De Gruyter Oldenbourg, 2015. ISBN: 978-3-11-034371-7.
- GOTTSCHALK 2000  
Gottschalk, Stefan: Collision Queries Using Oriented Bounding Boxes. Chapel Hill: The University of North Carolina, 2000. ISBN: 0-493-01573-6.
- GROßBERNDT 1988  
Großberndt, Hermann: Die automatisierte Montage mit Schrauben. Ehningen bei Böblingen: expert-Verl., 1988. ISBN: 978-3816903185.

GU & YAN 1995

GU, P. & YAN, X.: CAD-directed automatic assembly sequence planning in International Journal of Production Research. Vol. 33 11, 1995. S.3069–3100

HAAG 2015

Haag, Michael: Kollaboratives Arbeiten mit Robotern – Vision und realistische Perspektive in Zukunft der Arbeit in Industrie 4.0: Springer-Vieweg, 2015. S.59–64

HAMMERSTINGL & REINHART 2017

Hammerstingl, Veit & Reinhart, Gunther: Fähigkeiten in der Montage. München, 2017 <<http://mediatum.ub.tum.de/?id=1370174>> 21.08.2019.

HANDLING 2019

handling.de (Hrsg.): Mit Cobots gegen den Fachkräftemangel.

<<https://www.handling.de/robotertechnik/mrk-in-der-flaschenverpackung---mit-cobots-gegen-den-fachkraeftemangel.htm>> 21.08.2019.

HAUN 2013

Haun, Matthias: Handbuch Robotik. 2. Auflage. Berlin, Heidelberg: Springer Vieweg, 2013. ISBN: 978-3-642-39858-2.

HEINZE ET AL. 2016

Heinze, Frank; Klöckner, Maike; Wantia, Nils; Rossmann, Jürgen; Kuhlenkötter, Bernd; Deuse, Jochen: Combining Planning and Simulation to Create Human Robot Cooperative Processes with Industrial Service Robots in Applied Mechanics and Materials. Vol. 840, 2016. S.91–98

HEINZMANN & ZELINSKY 2016

Heinzmann, Jochen & Zelinsky, Alexander: Quantitative Safety Guarantees for Physical Human-Robot Interaction in The International Journal of Robotics Research. Vol. 22 7-8, 2016. S.479–504

HOFFMAN & BREAZEAL 2007

Hoffman, Guy & Breazeal, Cynthia: Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team in Proceeding of the ACM/IEEE international conference on Human-robot interaction - HRI '07. Arlington, Virginia, USA. 10. - 12. März, 2007. S.1–8

HOFFMANN ET AL. 2016

Hoffmann, Alwin; Poeppl, Alexander; Schierl, Andreas; Reif, Wolfgang: Environment-aware proximity detection with capacitive sensors for human-robot-interaction in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Daejeon, South Korea. 09.10.2016 - 14.10.2016, 2016. S.145–150

HUBER ET AL. 2010

Huber, Markus; Knoll, Alois; Brandt, Thomas; Glasauer, Stefan: When to assist? - Modelling human behaviour for hybrid assembly systems in Robotics (ISR), 2010 41<sup>st</sup> International Symposium on and 2010 6<sup>th</sup> German Conference on Robotics (ROBOTIK). Berlin: VDE Verlag, 2010. S.1–6

HUBER ET AL. 2013

Huber, Markus; Lenz, Claus; Wendt, Cornelia; Farber, Berthold; Knoll, Alois; Glasauer, Stefan: Increasing efficiency in robot-supported assemblies through predictive mechanisms in 2013 IEEE RO-MAN. Gyeongju. 26.08.2013 - 29.08.2013, 2013. S.503–508

HUMBURGER 1998

Humburger, Rainer: Konzeption eines Systems zur aufgabenorientierten Roboterprogrammierung. Aachen: Shaker, 1998. ISBN: 3826534271.

INSTITUT DER DEUTSCHEN WIRTSCHAFT KÖLN 2017

Institut der deutschen Wirtschaft Köln: Die deutsche Konjunktur am Limit? Fachkräftemangel als Wachstumsbremse in IW-Trends 44, 2017.

INTERNATIONAL FEDERATION OF ROBOTICS 2018

International Federation of Robotics: World Robotics - Industrial Robots 2018. Frankfurt am Main: VDMA Verlag, 2018. ISBN: 978-3-8163-0725-9.

ISO TS 15066

ISO TS 15066: 15066:2016 Robots and robotic devices—Collaborative robots 2016

JUSCHKAT 2018

Juschkat, Katharina: Mit Sensorhaut aus Industrierobotern Cobots machen. elektrotechnik, 2018 <<https://www.elektrotechnik.vogel.de/mit-sensorhaut-aus-industrierobotern-cobots-machen-a-727956/>> 21.08.2019.

KIM ET AL. 2010

Kim, Eunju; Helal, Sumi; Cook, Diane: Human Activity Recognition and Pattern Discovery in IEEE pervasive computing. Vol. 9 1, 2010. S.48–53

KINKEL 2005

Kinkel, Steffen: Anforderungen an die Fertigungstechnik von morgen in Mitteilungen aus der Produktionsinnovationserhebung 37, 2005. S.1–12

KOMPETENZZENTRUM USABILITY MITTELSTAND 2019

Kompetenzzentrum Usability Mittelstand (Hrsg.): Nutzerbefragungen des Kompetenzzentrums.

<<https://www.usabilityzentrum.de/dokumente/crm.pdf>> 22.02.2019.

KOMPI.ORG 2019

Kompi: KoMPI: Verrichtungsbasierte, digitale Planung kollaborativer Montagesysteme und Integration in variable Produktionsszenarien <<https://kompi.org/projekt/>> 21.12.2019.

KONOLD & REGER 2003

Konold, Peter & Reger, Herbert: Praxis der Montagetechnik. Wiesbaden: Vieweg+Teubner Verlag, 2003. ISBN: 978-3-663-01610-6.

KOPPENBORG ET AL. 2013

Koppenborg, Markus; Lungfiel, Andy; Naber, Birgit; Nickel, Peter: Auswirkung von Autonomie und Geschwindigkeit in der virtuellen Mensch-Roboter-Kollaboration in Chancen durch Arbeits-, Produkt- und Systemgestaltung. Dortmund: GfA-Press, 2013. S.417–420

KOŠTURIK & GREGOR 1999

Košturiak, Ján & Gregor, Milan: Simulation in production system life cycle in Computers in Industry. Vol. 38 2, 1999. S.159–172

KREISZEITUNG.DE 2019

kreiszeitung.de (Hrsg.): Kollege Roboter. <<https://www.kreiszeitung.de/lokales/bremen/kollege-roboter-12296745.html>> 07.09.2019.

KRUG 2013

Krug, Stefan: Automatische Konfiguration von Robotersystemen (Plug and Produce). München: Utz, 2013. ISBN: 3831642435.

KUGELMANN 1999

Kugelman, Doris: Aufgabenorientierte Offline-Programmierung von Industrierobotern. München: Utz, 1999. ISBN: 9783896756152.

KUKA AG 2019

KUKA AG (Hrsg.): KUKA.Sim. <[https://www.kuka.com/de-de/produkte-leistungen/robotersysteme/software/planung-projektierung-service-sicherheit/kuka\\_sim](https://www.kuka.com/de-de/produkte-leistungen/robotersysteme/software/planung-projektierung-service-sicherheit/kuka_sim)> 07.09.2019.

KUKA AG 2019

KUKA AG (Hrsg.): LBR iiwa. <<https://www.kuka.com/de-de/produkte-leistungen/robotersysteme/industrieroboter/lbr-iiwa>> 21.08.2019

LASOTA ET AL. 2014

Lasota, Przemyslaw A.; Fong, Terrence; Shah, Julie A.: A Survey of Methods for

Safe Human-Robot Interaction in Foundations and Trends in Robotics. Vol. 5 3, 2014. S.261–349

LEMMERZ ET AL. 2018

Lemmerz, K.; Glogowski, P.; Hypki, A.; Kuhlenkötter, B.: Functional Integration of a Robotics Software Framework into a Human Simulation System in Proceedings of ISR 2018: 50<sup>th</sup> International Symposium on Robotics. München. 20. – 21. Juni, 2018. S.255–262

LENZ ET AL. 2008

Lenz, Claus; Nair, Suraj; Rickert, Markus; Knoll, Alois; Rosel, Wolfgang; Gast, Jurgen; Bannat, Alexander; Wallhoff, Frank: Joint-action for humans and industrial robots for assembly tasks in The 17<sup>th</sup> IEEE International Symposium on Robot and Human Interactive Communication. Munich, Germany. 1. - 3. August, 2008. S.130– 135

LINSINGER ET AL. 2018

Linsinger, Matthias; Sudhoff, Martin; Lemmerz, Kai; Glogowski, Paul; Kuhlenkötter, Bernd: Task-based Potential Analysis for Human-Robot Collaboration within Assembly Systems in Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. S.1–12

LOTTER & WIENDAHL 2006

Lotter, Bruno & Wiendahl, Hans-Peter: Montage in der industriellen Produktion. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg, 2006. ISBN: 9783540366690.

LUCA ET AL. 2006

Luca, Alessandro; Albu-Schaffer, Alin; Haddadin, Sami; Hirzinger, Gerd: Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing, China. 09. - 15. Oktober, 2006. S.1623–1630

MATTHIAS ET AL. 2013

Matthias, Björn; Ding, Hao; Miegel, Volker: Die Zukunft der Mensch-Roboter Kollaboration in der industriellen Montage in Internationales Forum Mechatronics. Winterthur. 30. - 31. Oktober, 2013.

MAURTUA ET AL. 2017

Maurtua, Iñaki; Ibarguren, Aitor; Kildal, Johan; Susperregi, Loreto; Sierra, Basilio:

Human–robot collaboration in industrial applications in *International Journal of Advanced Robotic Systems*. Vol. 14 4, 2017. S.1-10

MEZIANE ET AL. 2017

Meziane, Ramy; Otis, Martin J.-D.; Ezzaidi, Hassan: Human-robot collaboration while sharing production activities in dynamic environment in *Robotics and Computer-Integrated Manufacturing*. Vol. 48, 2017. S.243–253

MICHALOS ET AL. 2014

Michalos, George; Makris, Sotiris; Spiliotopoulos, Jason; Misios, Ioannis; Tsarouchi, Panagiota; Chryssolouris, George: ROBO-Partner: Seamless Human-Robot Cooperation for Intelligent, Flexible and Safe Operations in the Assembly Factories of the Future in *Procedia CIRP*. Dresden. 13. - 14. November, 2014. S.71– 76

MICHALOS ET AL. 2015

Michalos, George; Makris, Sotiris; Tsarouchi, Panagiota; Guasch, Toni; Kontovrakis, Dimitris; Chryssolouris, George: Design Considerations for Safe Human-robot Collaborative Workplaces in *CIRPe 2015 - Understanding the life cycle implications of manufacturing*. Vol. 37, 2015. S.248–253

MICHNIEWICZ ET AL. 2016

Michniewicz, Joachim; Reinhart, Gunther; Boschert, Stefan: CAD-Based Automated Assembly Planning for Variable Products in Modular Production Systems in 6<sup>th</sup> CIRP Conference on Assembly Technologies and Systems (CATS). Göteborg. 16. - 18. Mai, 2016. S.44–49

MICHNIEWICZ 2019

Michniewicz, Joachim: Automatische simulationsgestützte Arbeitsplanung in der Montage. München: Utz, 2019.

MICHNIEWICZ & REINHART 2016

Michniewicz, Joachim & Reinhart, Gunther: Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks in *Mechatronics*. Vol. 34, 2016. S.170–180

MIKKEL T. JENSEN 2001

Mikkkel T. Jensen: Generating robust and flexible scheduling with evolutionary computation. Aarhus, 2001

<<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.5896>> 21.08.2019.

MÜHLSTEDT 2016

Mühlstedt, Jens: Digitale Menschmodelle in Homo Sapiens Digitalis - Virtuelle

Ergonomie und digitale Menschmodelle. Wiesbaden: Springer Vieweg, 2016. S.73– 182

MÜHLSTEDT & SPANNER-ULMER 2010

Mühlstedt, Jens & Spanner-Ulmer, Birgit: Homo Sapiens Digitalis: über den Praxiseinsatz digitaler Menschmodelle in 8. Berliner Werkstatt Mensch-Maschine-Systeme; ZMMS Spektrum Band 22; Fortschr.-Ber. VDI Reihe 22 Nr. 29, 2010.

MÜLLER ET AL. 2015

Müller, Rainer; Eilers, Jan; Hörauf, Leenhard; Hermanns, Lukas: Modell zur fähigkeitsbasierten Montageplanung in ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb. Vol. 110 9, 2015. S.553–557

MÜLLER ET AL. 2016

Müller, Rainer; Vette, Matthias; Mailahn, Ortwin: Process-oriented Task Assignment for Assembly Processes with Human-robot Interaction in 6<sup>th</sup> CIRP Conference on Assembly Technologies and Systems (CATS). Göteborg. 16. - 18. Mai, 2016. S.210–215

MÜLLER ET AL. 2019

Müller, Rainer; Franke, Jörg; Henrich, Dominik; Kuhlenkötter, Bernd; Raatz, Annika; Verl, Alexander: Handbuch Mensch-Roboter-Kollaboration. München: Hanser, 2019. ISBN: 3446450165.

NAUMANN ET AL. 2010

Naumann, Martin; Bengel, Matthias; Verl, Alexander: Automatic Generation of Robot Applications Using a Knowledge Integration Framework in Robotics (ISR), 2010 41<sup>st</sup> International Symposium on and 2010 6<sup>th</sup> German Conference on Robotics (ROBOTIK), 2010. S.631–638

NEBDAL 2013

Nebdal, Dietmar: Entwicklung eines Vorgehensmodells für die partizipative Einführung betrieblicher Integrationslösungen. Österreich, 2013.

NETO ET AL. 2010a

Neto, Pedro; Pires, J. Norberto; Moreira, A. Paulo: CAD-based off-line robot programming in 2010 IEEE Conference on Robotics, Automation and Mechatronics. Singapore. 28.- 30. Juni, 2010. S.516–521

NETO ET AL. 2010b

Neto, Pedro; Norberto Pires, J.; Paulo Moreira, A.: High-level programming and control for industrial robotics in Industrial Robot: An International Journal. Vol. 37 2, 2010. S.137–147



NORBERTO PIRES ET AL. 2009

Norberto Pires, J.; Veiga, Germano; Araújo, Ricardo: Programming-by-demonstration in the coworker scenario for SMEs in Industrial Robot: An International Journal. Vol. 36 1, 2009. S.73–83

OBERER-TREITZ ET AL. 2013

Oberer-Treitz, Susanne; Dietz, Thomas; Verl, Alexander: Safety in industrial applications in IEEE ISR 2013. Seoul, Korea (South). 24. - 26. Oktober, 2013. S.1– 4

ONNASCH ET AL. 2016

Onnasch, Linda; Maier, Xenia; Jürgensohn, Thomas: Mensch-Roboter-Interaktion - Eine Taxonomie für alle Anwendungsfälle, 2016

<[https://www.baua.de/DE/Angebote/Publikationen/Fokus/Mensch-Roboter-Interaktion.pdf?\\_\\_blob=publicationFile&v=6](https://www.baua.de/DE/Angebote/Publikationen/Fokus/Mensch-Roboter-Interaktion.pdf?__blob=publicationFile&v=6)> 21.08.2019.

ORENDT ET AL. 2016

Orendt, Eric M.; Fichtner, Myriel; Henrich, Dominik: Robot programming by non-experts: Intuitiveness and robustness of One-Shot robot programming in 2016 25<sup>th</sup> IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). New York, NY, USA, 2016. S.192–199

OU & XU 2013

Ou, Li-Ming & Xu, Xun: Relationship matrix based automatic assembly sequence generation from a CAD model in Computer-Aided Design. Vol. 45 7, 2013. S.1053–1067

PELLEGRINELLI ET AL. 2017

Pellegrinelli, Stefania; Orlandini, Andrea; Pedrocchi, Nicola; Umbrico, Alessandro; Tolio, Tullio: Motion planning and scheduling for human and industrial-robot collaboration in CIRP Annals - Manufacturing Technology. Vol. 66 1, 2017. S.1–4

PERDIKIS ET AL. 2008

Perdikis, Serafeim; Tzovaras, Dimitrios; Strintzis, Michael Gerasimos: Recognition of Human Actions using Layered Hidden Markov Models in Proceedings 2008 IAPR, 2008.

PERZYLO ET AL. 2015a

Perzylo, Alexander; Somani, Nikhil; Profanter, Stefan; Rickert, Markus; Knoll, Alois: Multimodal binding of parameters for task-based robot programming based on semantic descriptions of modalities and parameter types in IEEE/RSJ

International Conference on Intelligent Robots and Systems (IROS). Hamburg. 28. September - 02. Oktober, 2015.

PERZYLO ET AL. 2015b

Perzylo, Alexander; Somani, Nikhil; Profanter, Stefan; Rickert, Markus; Knoll, Alois: Toward Efficient Robot Teach-In and Semantic Process Descriptions for Small Lot Sizes, 2015.

PLATTFORM INDUSTRIE 4.0

Plattform Industrie 4.0 (Hrsg.): Termin: Kollege Roboter - wie künstliche Intelligenz den Arbeitsalltag verändert. <<https://www.plattform-i40.de/PI40/Redaktion/DE/Newsletter/2019/Ausgabe17/2019-17-Kollege-Roboter.html>> 07.09.2019.

POLLICK & ISHIMURA 1996

Pollick, F. E. & Ishimura, G.: The Three-Dimensional Curvature of Straight-Ahead Movements in Journal of motor behavior. Vol. 28 3, 1996. S.271–279

PÖTZSCH & RÖßGER 2015

Pöttsch, Olga & Rößger, Felix: Bevölkerung Deutschlands bis 2060. Statistisches Bundesamt, Wiesbaden, 2015  
<[https://www.destatis.de/GPStatistik/servlets/MCRFileNodeServlet/DEMonografie\\_derivate\\_00001444/Bevoelkerung2060\(13\).pdf;jsessionid=A5121B19338FF6AFCB614AAE221BD948](https://www.destatis.de/GPStatistik/servlets/MCRFileNodeServlet/DEMonografie_derivate_00001444/Bevoelkerung2060(13).pdf;jsessionid=A5121B19338FF6AFCB614AAE221BD948)> 09.01.2020

QUIGLEY ET AL. 2009

Quigley, Morgan; Conley, Ken; Gerkey, Brian P.; Faust, Josh; Foote, Tully; Leibs, Jeremy; Wheeler, Rob; Ng, Andrew Y.: ROS: an open-source Robot Operating System in ICRA Workshop on Open Source Software, 2009.

RABINER 1989

Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition in Proceedings of the IEEE. Vol. 77 2, 1989. S.257–286

RANZ ET AL. 2017

Ranz, Fabian; Hummel, Vera; Sihn, Wilfried: Capability-based Task Allocation in Human-robot Collaboration in 7<sup>th</sup> Conference on Learning Factories, CLF 2017. Darmstadt. 04. - 05. April, 2017. S.182–189

RAO ET AL. 2002

Rao, Cen; Yilmaz, Alper; Shah, Mubarak: View-Invariant Representation and Recognition of Actions in International Journal of Computer Vision. Vol. 50 2, 2002. S.203–226

RAWAT ET AL. 2016

Rawat, Seema; Vats, Somya; Kumar, Praveen: Evaluating and exploring the MYO ARMBAND in Proceedings of the 5<sup>th</sup> International Conference System Modeling & Advancement in Research Trends (25<sup>th</sup>-27<sup>th</sup> November 2016). Moradabad, India. 25. - 27. November, 2016. S.115–120

REFA BUNDESVERBAND E.V. 2002

REFA Bundesverband e.V.: Ausgewählte Methoden zur prozessorientierten Arbeitsorganisation. 1. Hemsbach: Beltz Druckpartner, 2002.

REINHART 2017

Reinhart, Gunther: Handbuch Industrie 4.0. München: Hanser, 2017. ISBN: 3446446427.

RETHINK ROBOTICS 2018

Rethink Robotics (Hrsg.): Sawyer. <<https://www.rethinkrobotics.com/de/sawyer/>> 14.08.2018.

RICHTER ET AL. 2015

Richter, Alexander; Heinrich, Peter; Stocker, Alexander; Unzeitig, Wolfgang: Der Mensch im Mittelpunkt der Fabrik von morgen in HMD Praxis der Wirtschaftsinformatik. Vol. 52 5, 2015. S.690–712

RIEDL ET AL. 2016

Riedl, Michael; Baumgartl, Johannes; Henrich, Dominik: Editing and synchronizing multi-robot playback programs in Proceedings of ISR 2016: 47st International Symposium on Robotics. Piscataway, NJ: IEEE, 2016.

ROBERT BOSCH MANUFACTURING SOLUTIONS GMBH

Robert Bosch Manufacturing Solutions GmbH <<https://www.bosch-apas.com/produkte-und-services/apas-assistant-mobile/>> 05.09.2018.

ROBLA-GOMEZ ET AL. 2017

Robla-Gomez, S.; Becerra, Victor M.; Llata, J. R.; Gonzalez-Sarabia, E.; Torre-Ferrero, C.; Perez-Oria, J.: Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments in IEEE Access. Vol. 5, 2017. S.26754–26773

ROBODK 2018

RoboDK (Hrsg.): Simulate Robot Applications. <<https://robodk.com/>> 24.08.2018.

ROBOTIK UND PRODUKTION 2018

Robotik und Produktion: MRK-Roboter in Robotik und Produktion. Vol. 3 5, 2018. S.41–44

ROITBERG ET AL. 2014

Roitberg, Alina; Perzylo, Alexander; Somani, Nikhil; Giuliani, Manuel; Rickert, Markus; Knoll, Alois: Human activity recognition in the context of industrial human-robot interaction in 2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). Chiang Mai, Thailand, 2014. S.1–10

ROSS 2002

Ross, Paul: Bestimmung des wirtschaftlichen Automatisierungsgrades von Montageprozessen in der frühen Phase der Montageplanung. München: Utz, 2002. ISBN: 3831601917.

ROSSANO ET AL. 2013

Rossano, Gregory F.; Martinez, Carlos; Hedelind, Mikael; Murphy, Steve; Fuhlbrigge, Thomas A.: Easy robot programming concepts - An industrial perspective in 2013 IEEE International Conference on Automation Science and Engineering (CASE). Madison, WI, USA. 17. - 20. August, 2013. S.1119–1126

ROTTACH & GROß 2002

Rottach, Thilo & Groß, Sascha: XML kompakt. Heidelberg: Spektrum Akad. Verl., 2002. ISBN: 3827413397.

RÜCKERT ET AL. 2018

Rückert, Patrick; Adam, Jannes; Papenberg, Björn; Paulus, Henning; Tracht, Kirsten: Calibration of a modular assembly system for personalized and adaptive human robot collaboration in Procedia CIRP. Tianjin, China. 10. - 12. Mai, 2018. S.199–204

SAND & HENRICH 2014

Sand, Maximilian & Henrich, Dominik: Intuitive Erzeugung von 3D-Modellen mit handgehaltenen Sensoren in Mensch und Computer 2014 - Tagungsband. Berlin: De Gruyter Oldenbourg, 2014. S.85–94

SCHENCK & WILSON 1994

Schenck, Douglas & Wilson, Peter: Information modeling. New York: Oxford University Press, 1994. ISBN: 0195087143.

SCHIEMANN ET AL. 2018

Schiemann, Markus; Hodapp, Jan; Berger, Ulrich: Collaboration-Gap: A bus-modular architecture for human-robot-collaboration systems in production environments in Proceedings of ISR 2018: 50<sup>th</sup> International Symposium on Robotics. München. 20. – 21. Juni, 2018. S.450–454

SCHLICK ET AL. 2014

Schlick, Christopher M.; Faber, Marco; Kuz, Sinem; Bützler, Jennifer: Erweiterung einer kognitiven Architektur zur Unterstützung der Mensch-Roboter-Kooperation in der Montage, 2014.

SCHMIDT & WANG 2013

Schmidt, Bernard & Wang, Lihui: Contact-less and Programming-less Human-Robot Collaboration in 46<sup>th</sup> CIRP Conference on Manufacturing Systems 2013. Sesimbra. 29. - 31. Mai, 2013. S.545–550

SCHRAFT & MEYER 2006

Schraft, R. D. & Meyer, Christian: The need for an intuitive teaching method for small and medium enterprises in Robotik 2006, 2006. S.95–105

SCHRÖTER 2018

Schröter, Daniel: Entwicklung einer Methodik zur Planung von Arbeitssystemen in Mensch-Roboter-Kooperation. Stuttgart: Fraunhofer Verlag, 2018. ISBN: 2195-2892.

SEIDL 1994

Seidl, Andreas: Das Menschmodell RAMSIS - Analyse, Synthese und Simulation dreidimensionaler Körperhaltungen des Menschen. München, 1994.

SEITER 2017

Seiter, Mischa: Verbundforschungsprojekt „KoKoMo“ zur Mensch-Roboter-Kollaboration gestartet, 2017.

SELIG 2011

Selig, Andreas: Informationsmodell zur funktionalen Typisierung von Automatisierungsgeräten. Heimsheim: Jost-Jetter, 2011. ISBN: 978-3-939890-74-4.

SHEN 2015

Shen, Yi: System für die Mensch-Roboter-Koexistenz in der Fließmontage. München: Herbert Utz Verlag, 2015. ISBN: 978-3-8316-7167-0.

SHIFFMAN 2012

Shiffman, Daniel: The nature of code. Version 1.0, generated December 6, 2012. s.l.: Selbstverl., 2012. ISBN: 0985930802.

SIEMENS INDUSTRY SOFTWARE 2020

Siemens Industry Software (Hrsg.): Human-centered design, simulation and ergonomics.

<<https://www.plm.automation.siemens.com/global/en/products/tecnomatix/human-modeling-simulation.html>> 06.01.2020

SIHN ET AL. 2016

Sihn, Wilfried; Ranz, Fabian; Edtmayr, Thomas; Hold, Philipp; Reisinger, Gerhard; Markis, Alexandra; Montenegro, Harald; Neuhold, Michael; Oberweger, Andreas; Schlosser, Christian; Schwald, Christoph: Sicherheit in der Mensch-Roboter-Kollaboration. Erste Ausgabe. Wien, 2016.

SIHN ET AL. 2017

Sihn, Wilfried; Ranz, Fabian; Edtmayr, Thomas; Hold, Philipp; Reisinger, Gerhard; Markis, Alexandra; Montenegro, Harald; Neuhold, Michael; Oberweger, Andreas; Schwald, Christoph: Sicherheit in der Mensch-Roboter-Kollaboration. Wien, 2017.

SPILLNER 2015

Spillner, Rüdiger: Einsatz und Planung von Roboterassistenz zur Berücksichtigung von Leistungswandlungen in der Produktion. München: Herbert Utz Verlag, 2015. ISBN: ISBN 978-3-8316-7114-4.

STOLZENBERG & HEBERLE 2009

Stolzenberg, Kerstin & Heberle, Krischan: Change Management. 2. Aufl. s.l.: Springer-Verlag, 2009. ISBN: 3540788549.

THIEMERMANN 2005

Thiemermann, Stefan: Direkte Mensch-Roboter-Kooperation in der Kleinteilemontage mit einem SCARA-Roboter. Heimsheim: Jost-Jetter, 2005. ISBN: 3-936947-50-3.

THOMAS 2008

Thomas, Ulrike: Automatisierte Programmierung von Robotern für Montageaufgaben. Aachen: Shaker, 2008. ISBN: 9783832271015.

THOMAS ET AL. 2016

Thomas, C.; Stankiewicz, L.; Grötsch, A.; Wischniewski, S.; Deuse, J.; Kuhlentötter, B.: Intuitive Work Assistance by Reciprocal Human-robot Interaction in the Subject Area of Direct Human-robot Collaboration in 6<sup>th</sup> CIRP Conference on Assembly Technologies and Systems (CATS). Göteborg. 16. - 18. Mai, 2016. S.275–280

THOMAS 2017

Thomas, Carsten: Entwicklung einer Bewertungssystematik für die Mensch-Roboter-Kollaboration. 1. Auflage, 2017. ISBN: 978-3-8440-5416-3.

TSAROUCI ET AL. 2015

Tsarouchi, Panagiota; Makris, Sotiris; Michalos, George; Matthaiakis, Alexandros-Stereos; Chatzigeorgiou, Xenofon; Athanasatos, Athanasios; Stefos, Michael;

Aivaliotis, Panagiotis; Chryssolouris, George: ROS Based Coordination of Human Robot Cooperative Assembly Tasks-An Industrial Case Study in *Procedia CIRP*. Vol. 37, 2015. S.254–259

TSAROUCHI ET AL. 2016a

Tsarouchi, Panagiota; Makris, Sotiris; Chryssolouris, George: On a Human and Dual-arm Robot Task Planning Method in *Procedia CIRP*. Vol. 57, 2016. S.551– 555

TSAROUCHI ET AL. 2016b

Tsarouchi, Panagiota; Matthaiakis, Alexandros-Stereos; Makris, Sotiris; Chryssolouris, George: On a human-robot collaboration in an assembly cell in *International Journal of Computer Integrated Manufacturing*. Vol. 30 6, 2016. S.580–589

TSAROUCHI ET AL. 2017

Tsarouchi, Panagiota; Michalos, George; Makris, Sotiris; Athanasatos, Thanasis; Dimoulas, Konstantinos; Chryssolouris, George: On a human–robot workplace design and task allocation system in *International Journal of Computer Integrated Manufacturing*, 2017. S.1–8

ULBRICH ET AL. 2015

Ulbrich, Simon; Menzel, Till; Reschka, Andreas; Schuldt, Fabian; Maurer, Markus: Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving in 2015 IEEE 18<sup>th</sup> International Conference on Intelligent Transportation Systems. Gran Canaria, Spain. 15.09.2015 - 18.09.2015, 2015. S.982–988

ULRICH & HILI 1976

Ulrich, Peter & Hili, Wilhelm: *Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil I)* in *WiST Zeitschrift für Ausbildung und Hochschulkontakt*. Heft 7, 1976.

UNIVERSAL ROBOTS A/S 2020

Universal Robots A/S (Hrsg.): *Universal Robots e-Series*. <<https://www.universal-robots.com/de/e-series/>> 06.01.2020

VDI RICHTLINIE 3633

VDI RICHTLINIE 3633: *Simulation von Logistik-, Materialfluss- und Produktionssystemen*. 2014

VDMA ROBOTICS + AUTOMATION 2016

VDMA Robotics + Automation: *Safety in Human-Robot Collaboration in*

Wissenschaftliche Gesellschaft für Montage, Handhabung und Industrierobotik - MHI e.V., 2016.

VISUAL COMPONENTS GMBH 2019

Visual Components GmbH (Hrsg.): Visual Components 4.2.

<<https://www.visualcomponents.com/de/produkte/visual-components/>> 16.08.2019.

VOSS & NENTWIG 1998

Voss, Josef & Nentwig, Dietmar: Entwicklung von graphischen

Benutzungsschnittstellen. München: Hanser, 1998. ISBN: 9783446190894.

WANTIA ET AL. 2016

Wantia, Nils; Esen, Menno; Hengstebeck, Andre; Heinze, Frank; Rossmann, Juergen; Deuse, Jochen; Kuhlenkoetter, Bernd: Task planning for human robot interactive processes in 2016 IEEE 21<sup>st</sup> International Conference on Emerging Technologies and Factory Automation (ETFA). Berlin, Germany. 6/9/2016 - 9/9/2016, 2016. S.1–8

WARD ET AL. 2006

Ward, Jamie A.; Lukowicz, Paul; Tröster, Gerhard; Starner, Thad E.: Activity recognition of assembly tasks using body-worn microphones and accelerometers in IEEE transactions on pattern analysis and machine intelligence. Vol. 28 10, 2006. S.1553–1567

WARNECKE 1995

Warnecke, Hans-Jürgen: Die Montage im flexiblen Produktionsbetrieb. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. ISBN: 978-3-642-79964-8.

WEBER 2009

Weber, Wolfgang: Industrieroboter. 2., neu bearb. Aufl. München: Hanser, 2009. ISBN: 978-3446410312.

WECK & BRECHER 2006

Weck, Manfred & Brecher, Christian: Werkzeugmaschinen 4. 6., neu bearbeitete Auflage. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006. ISBN: 9783540225072.

WEICHERT ET AL. 2013

Weichert, Frank; Bachmann, Daniel; Rudak, Bartholomäus; Fisseler, Denis: Analysis of the Accuracy and Robustness of the Leap Motion Controller in Sensors (Basel, Switzerland). Vol. 13 5, 2013. S.6380–6393



WIELAND 1995

Wieland, Eugen: Anwendungsorientierte Programmierung für die robotergestützte Montage. Berlin, Heidelberg: Springer, 1995. ISBN: 9783540590255.

WÖHE & DÖRING 2010

Wöhe, Günter & Döring, Ulrich: Einführung in die allgemeine Betriebswirtschaftslehre. 24., überarb. und aktualisierte Aufl. München: Vahlen, 2010. ISBN: 9783800637959.



# Anhang

## A1 Wissensmodell: Fügeoperationen

Ablauf	Schematischer Ablauf	Parameter	Sicherheitsrelevante Parameter
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Bewegen zu Montageposition</div> <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Aufsetzen durchführen</div> <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Greifer öffnen</div> <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 2px;">Vertikale Bewegung nach oben</div>		<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Montageposition (x/y/z)</div> <div style="border: 1px solid black; padding: 2px;">Greifkraft (N)* Greifweg (mm)*</div>	<div style="border: 1px solid black; padding: 2px;">Definierte Bewegung; reduzierte Geschwindigkeit zwischen A und B</div>

\*abhängig von Greifer

Abbildung 62: Ablauf und Parameter zum Auflegen, Aufsetzen und Schichten

Ablauf	Schematischer Ablauf	Parameter	Sicherheitsrelevante Parameter
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Bewegen zu Montageposition</div> <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Fügevorgang durchführen</div> <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Greifer öffnen</div> <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 2px;">Vertikale Bewegung nach oben</div>		<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Montageposition (x/y/z)</div> <div style="border: 1px solid black; padding: 2px;">Greifkraft (N)* Greifweg (mm)*</div>	<div style="border: 1px solid black; padding: 2px;">Definierte Bewegung; reduzierte Geschwindigkeit zwischen A und B</div>

\*abhängig von Greifer

Abbildung 63: Ablauf und Parameter zum Einlegen und Einsetzen

Ablauf	Schematischer Ablauf	Parameter	Sicherheitsrelevante Parameter
<p>Bewegen zu Montageposition</p> <p>↓</p> <p>Ineinanderschieben durchführen</p> <p>↓</p> <p>Greifer öffnen</p> <p>↓</p> <p>Vertikale Bewegung nach oben</p>		<p>Montageposition (x/y/z)</p> <p>Greifkraft (N)*</p> <p>Greifweg (mm)*</p>	<p>Definierte Bewegung; reduzierte Geschwindigkeit zwischen A und B</p>

\*abhängig von Greifer

Abbildung 64: Ablauf und Parameter zum Ineinanderschieben

Ablauf	Schematischer Ablauf	Parameter	Sicherheitsrelevante Parameter
<p>Bewegen zu Startposition</p> <p>↓</p> <p>Materialstrom einschalten</p> <p>↓</p> <p>Bahn abfahren</p> <p>↓</p> <p>Materialstrom abschalten</p> <p>↓</p> <p>Abfahren</p>		<p>Startposition (x/y/z)</p> <p>Bahnvektor</p> <p>Applikationsdruck</p> <p>Materialstrom</p>	<p>Definierte Bewegung; reduzierte Anfahrsgeschwindigkeit</p> <p>Überwachung der Temperatur</p>

Abbildung 65: Ablauf und Parameter beim Kleben

## A2 Wissensmodell: Beschreibung des Roboters

```

</Task>
<Task>
  <TaskName>Antransport Bauteil_1</TaskName>
  <MethodName>transportComponent</MethodName>
  <Attributes>
    <Attribute>
      <Name>StartPosition</Name>
      <Value>Please enter Value!</Value>
    </Attribute>
    <Attribute>
      <Name>EndPosition</Name>
      <Value>Please enter Value!</Value>
    </Attribute>
    <Attribute>
      <Name>Hoehendifferenz</Name>
      <Value>Please enter Value!</Value>
    </Attribute>
    <Attribute>
      <Name>Greifkraft</Name>
      <Value>Please enter Value!</Value>
    </Attribute>
    <Attribute>
      <Name>Greifdistanz</Name>
      <Value>Please enter Value!</Value>
    </Attribute>
  </Attributes>
  <Code>
    movej(&lt;attr&gt;StartPosition&lt;/attr&gt;);
    movel(&lt;attr&gt;StartPosition&lt;/attr&gt;);
    grip&lt;attr&gt;Greifkraft&lt;/attr&gt;(&lt;attr&gt;Greifdistanz&lt;/attr&gt;);
    movej(&lt;attr&gt;EndPosition&lt;/attr&gt;);
    movel(&lt;attr&gt;EndPosition&lt;/attr&gt;);
  </Code>
  <ExtraInformation>
    <StartingTime>Please enter Value!</StartingTime>
    <Duration>Please enter Value!</Duration>
    <Resource>Please enter Value!</Resource>
    <Attribute>
      <Name>StartPosition1</Name>
      <Value>Please enter Value!</Value>
    </Attribute>
  </Attribute>

```

Abbildung 66: Auszug aus Wissensmodell zur Beschreibung des Roboters UR10

### A3 Eignungsgrade von Mensch und Roboter

Tabelle 11: Darstellung der Eignungsgrade hinsichtlich der Kriterien

Kriterium	Ausprägungen											
	Beschr.	M	R	Beschr.	M	R	Beschr.	M	R	Beschr.	M	R
Abmaße	drei Ausdehnungen mittelgroß	4	4	eine Ausdehnung groß	3	3	zwei Ausdehnungen klein und/oder groß	3	2	drei Ausdehnungen groß	2	1
Masse	0 - 500g	4	4	0,5 kg - 3 kg	3	4	3 kg - 6 kg	2	4	6 kg - 10 kg	1	4
Empfindlichkeit	unempfindlich	4	4	kaum empfindlich	4	3	empfindlich	3	2	sehr empfindlich	3	1
Greifflächen	außen	4	4	innen	4	3	Magnetgreifer	4	2	mit Stoffschluss	4	1
Formstabilität	Formstabil	4	4	Reduziert formstabil	4	3	kaum formstabil	4	2	formlabil	3	1
Kräfte oder Momente	kein(e)	4	4	gering	3	3	mittel	2	3	hoch	1	2
Montagerichtung	von oben	4	4				seitlich	3	2			
Fügebewegung	linear	4	4	Rotation	4	3	linear-rotatorisch	4	2	Bahn-bewegung	3	1
Feinfühligkeit	notwendig	4	1				nicht notwendig	4	4			
Scharfkantigkeit	Kanten vorhanden	3	1				keine scharfen Kanten	4	4			

## A4 Berechnung der Matrizen des Hidden Markov Modells

### Transitions- und Emissionsmatrix des oberen Layers

Die Transitionsmatrix des oberen Layers wird bestimmt aus der Anzahl der Aufgaben des Menschen, die den Zuständen des Modells entsprechen. Hinzu kommt ein Grundzustand, in dem der Mensch keine Aufgabe erledigt und der von jedem anderen Zustand aus erreichbar sein muss. Daher wird aus dem übergebenen Ablauf mit den Aufgaben die Anzahl der Aufgaben  $t$  des Menschen bestimmt. Die Transitionsmatrix  $A_o$  des oberen Layers besitzt eine Größe von  $t \times t$ . Anschließend können die Werte für die einzelnen Einträge ermittelt werden. Die Diagonaleinträge erhalten den Wert 0,9. Alle weiteren Werte werden berechnet. Die Übergangswahrscheinlichkeiten vom Grundzustand in einen anderen Zustand werden über folgende Gleichung gegeben:

$$A_o(1, i) = \frac{(1 - A_o(1,1))}{(t-1)} \quad (\text{Gl. A4 1})$$

mit:

$$i: 1: 1:t$$

$t$ : Anzahl der Aufgaben

Der Übergang von einem Zustand in den nächsten, was folglich dem Übergang von einer Aufgabe in eine andere entspricht, wird wie nachfolgend dargestellt, berechnet:

$$A_o(i, j) = \frac{(1 - A_o(1,1))}{2} \quad \text{mit } i = 2: 1: (t - 1); j = 3: 1: t \quad (\text{Gl. A4 2})$$

Diese Formel ergibt sich dadurch, dass die Wahrscheinlichkeit in einen anderen Zustand zu wechseln gleich der Wahrscheinlichkeit ist, in den Grundzustand überzugehen. Abschließend können noch die Werte zum Übergang in den Grundzustand in der Matrix ergänzt werden. Dies erfolgt nach folgender Vorschrift:

$$A_o(i, 1) = 1 - (\text{sum}(A_o(i, :))) \quad \text{mit } i = 2: 1: t \quad (\text{Gl. A4 3})$$

Die Emissionsmatrix des oberen Layers beschreibt die Wahrscheinlichkeit, dass zu einem Zustand bestimmte Emissionen ausgesandt werden. Dies entspricht der Wahrscheinlichkeit, dass bei einer Aufgabe die Grundbewegungen ausgeführt werden. Umgekehrt bedeutet dies, dass bei Auftritt einer Hinlang-Bewegung zu einer Speichereinrichtung die Wahrscheinlichkeit besteht, dass diese zu einer Aufgabe gehört. Um diese Wahrscheinlichkeiten in der Emissionsmatrix zu beschreiben, werden zunächst die Anzahl der Grundbewegungen zu den Aufgaben bestimmt. Der

Grundzustand besitzt nur eine Unteraufgabe, weshalb sich für die Anzahl der Unteraufgaben  $s$  folgender Wert ergibt:

$$s = (2 \cdot t) - 1 \quad (\text{Gl. A4 4})$$

mit

$s$ : Anzahl der Elementarbewegungen

$t$ : Anzahl der Aufgaben

Die Emissionsmatrix des oberen Layers hat eine Größe von  $t \times s$ . Zur Parametrierung der Matrix wird ein Hilfsvektor  $H$  eingeführt, der für jede Aufgabe eine Spalte enthält. In diesem sind jeweils die Anzahl an Unteraufgaben enthalten. Damit können über folgende Formel die Positionen in der Emissionsmatrix  $B_o$  bestimmt werden, die mit Werten befüllt werden:

$$j > \text{sum}(\text{abs}(H(1:i-1,1))) \&\& j \leq \text{sum}(\text{abs}(H(1:i,1))) \quad (\text{Gl. A4 5})$$

mit  $i=1:1:t$

$j=1:1:s$

An den jeweiligen Positionen wird dann ein Wert eingefügt.

$$B_o(i,j) = \frac{1}{H(i,1)} \quad (\text{Gl. A4 6})$$

### **Transitions- und Emissionsmatrix des unteren Layers**

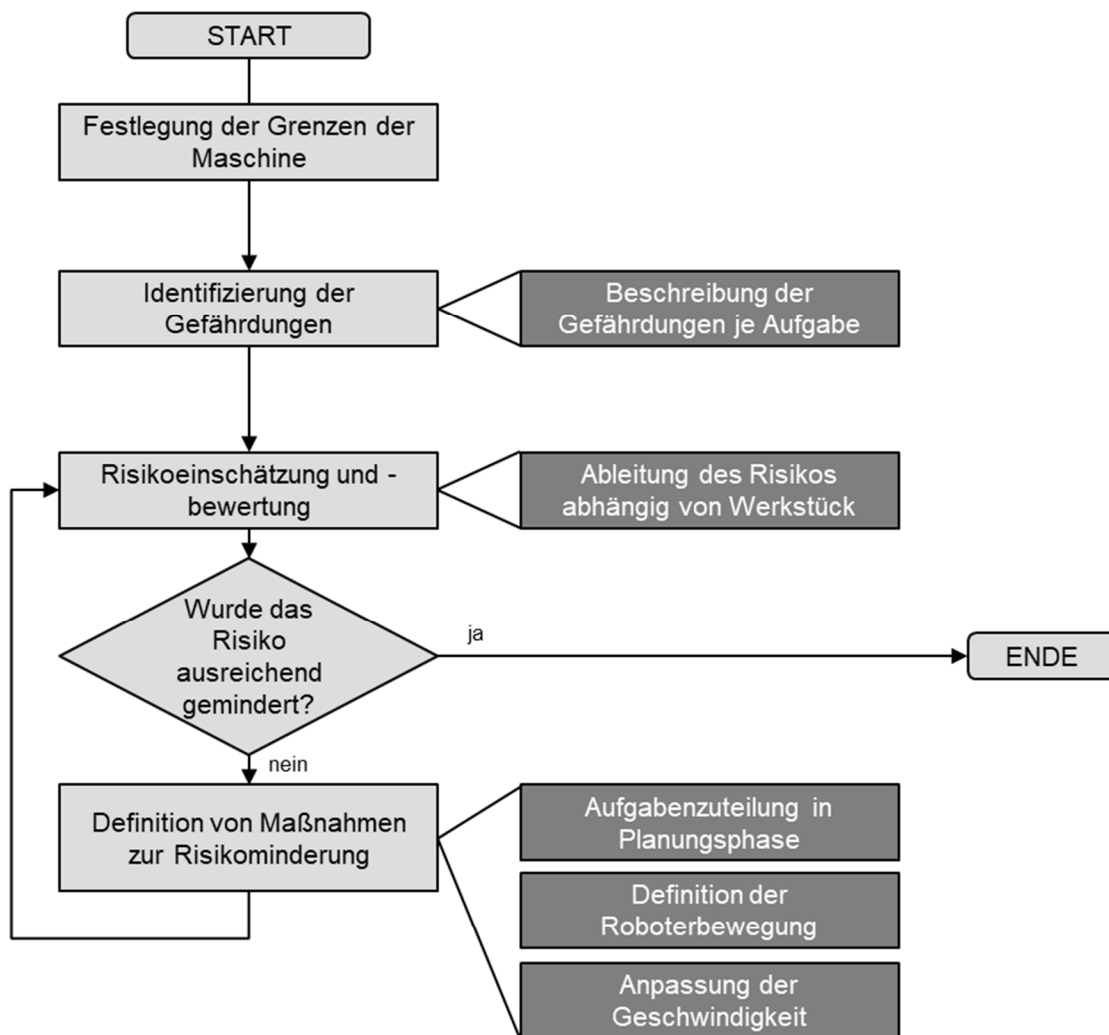
Die Transitionsmatrix des unteren Layers ist ähnlich zu der des oberen Layers aufgebaut und wird ebenso konfiguriert. Die Matrix besitzt eine Größe von  $s \times s$ , wobei  $s$  der Anzahl der Unteraufgaben beschreibt und nach (Gl. A4 4) berechnet wird.

Die Emissionsmatrix des unteren Layers wird über die Gaußschen Verteilungen für die Trajektorien und Sollwinkel beschrieben. Die Beschreibung ist in Abschnitt 6.2.2 beschrieben.



## A5 Unterstützte Risikoanalyse

### Ablauf der Risikoanalyse und Stellen der Unterstützung



Legende

Element der Risikobeurteilung

Unterstützung durch Programmiersystem

Abbildung 67: Elemente zur Unterstützung der Risikobeurteilung

### Fragen zur Prüfung des Arbeitsplatzes

1. Steht die Applikation nicht an einer Wand?
2. Steht der Roboter gegenüber dem Menschen?
3. Werden nur Komponenten, wie Endeffektoren und Roboter, verwendet, die für die Nutzung zugelassen sind?
4. Befinden sich im Arbeitsraum keine Objekte, die nicht für die Montageaufgabe relevant sind?

### Analyse der Gefährdungen der Aufgabenblöcke

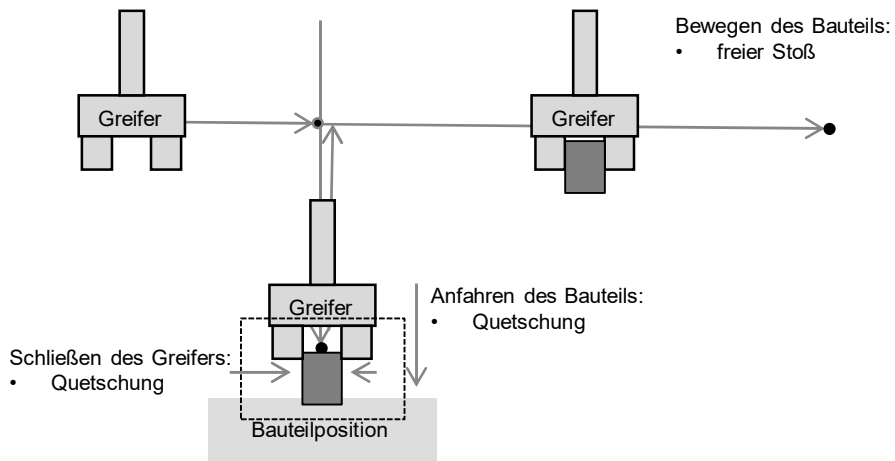


Abbildung 68: Analyse der Gefährdungen des Aufgabenblocks Antransport

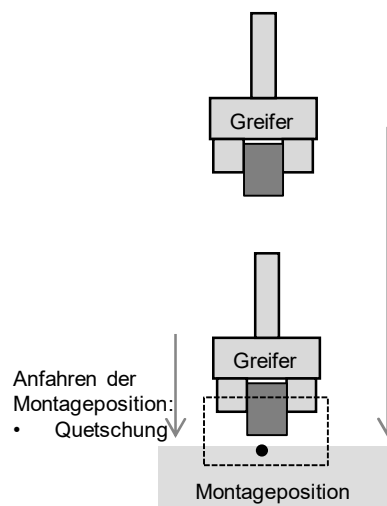


Abbildung 69: Analyse der Gefährdungen des Aufgabenblocks Fügen

## A6 Ergebnisse des genetischen Algorithmus

Tabelle 12: Zuordnung der Montageschritte zu den Nummern

Montage- schritt	Aufgabe	Bauteil	Zeit [s]	
			Mensch	Roboter
01	Antransport	Getriebehaelfte_2	0,7956	21
02	Fügen	Getriebehaelfte_2	0,7812	10
03	Antransport	Welle_lang	1,1880	15
04	Fügen	Welle_lang	1,6200	12
05	Antransport	Welle_kurz	1,1880	15
06	Fügen	Welle_kurz	1,6200	12
07	Antransport	Temperatursensor	1,4616	X
08	Fügen	Temperatursensor	2,1312	X
09	Antransport	Dichtung	1,2816	X
10	Fügen	Dichtung	0,7812	X
11	Antransport	Getriebehaelfte_1	0,7956	13
12	Fügen	Getriebehaelfte_1	1,7460	19
13	Antransport	Schraube	2,1348	15
14	Schrauben	Schraube	46,4371	8
15	Antransport	Schraube	2,1348	15
16	Schrauben	Schraube	46,4371	8
17	Antransport	Schraube	2,1348	15
18	Schrauben	Schraube	46,4371	8
19	Antransport	Schraube	2,1348	15
20	Schrauben	Schraube	46,4371	8

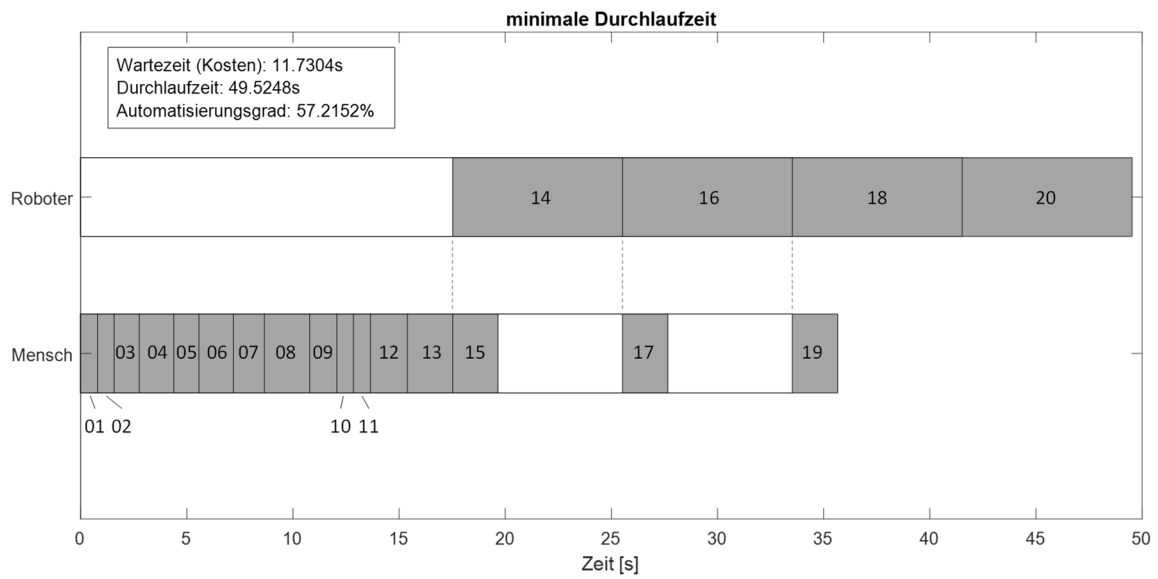


Abbildung 70: Ausgabe optimiert nach minimaler Durchlaufzeit

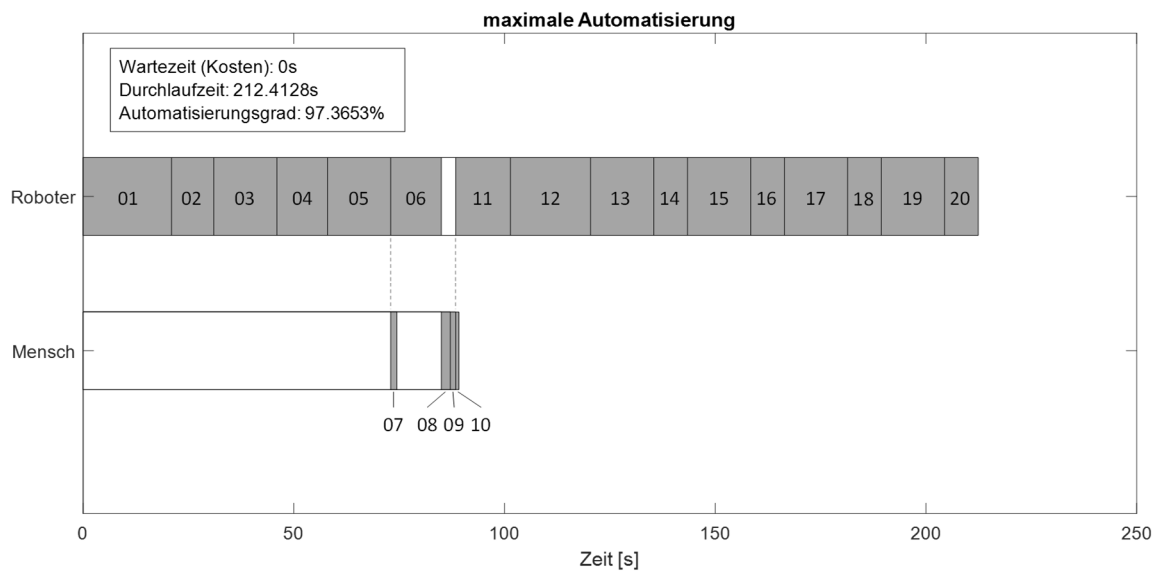


Abbildung 71: Ausgabe optimiert nach maximaler Automatisierung

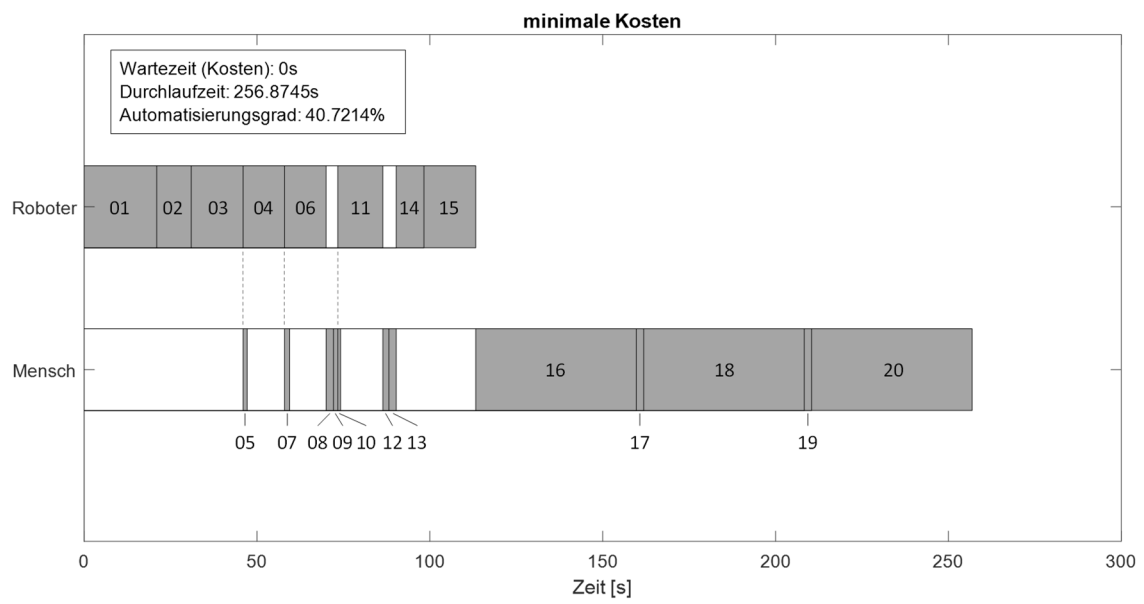


Abbildung 72: Ausgabe optimiert nach minimalen Kosten

## A7 Vorgehensmodell

*Tabelle 13: Fragen zum Reifegrad zum Einsatz einer Mensch-Roboter-Kooperation*

<b>Frage</b>	<b>Reifegrad</b>
Werden im Unternehmen bereits Leichtbauroboter eingesetzt?	3
Gibt es Experten zur Roboterprogrammierung?	2
Sind die Werkkräfte skeptisch gegenüber dem Einsatz von Robotern?	1
Werden die Aufgaben in der Montage überwiegend manuell gelöst?	3
Besteht eine hohe Variantenvielfalt?	3
Ist es notwendig, die Montageanlage oft umzurüsten?	3
Besteht ein schwankender personenseitiger Kapazitätsbedarf?	3

## A8 Evaluationsbogen

### Evaluation der Benutzeroberfläche des Planungssystems für die Roboterprogrammierung für die Mensch-Roboter-Kooperation (in Anlehnung an: ISO- Norm 9241 10 und Fragebögen des Kompetenzzentrums Usability)

Vielen Dank, dass Sie sich die Zeit für diese Evaluation nehmen. Bitte geben Sie zunächst generelle Informationen zu Ihrer Person an:

1. Ihr Geschlecht?  Weiblich  Männlich  Andere
2. Ihr Alter? \_\_\_\_\_
3. Haben Sie schon einmal einen Roboter programmiert?  Ja  Nein
4. Haben Sie schon einmal in der Montage gearbeitet?  Ja  Nein

Dieser Fragebogen dient der Evaluation der Benutzeroberfläche, wobei insbesondere betrachtet wird, inwiefern die benutzende Person intuitiv mit der Software umgehen kann. Infolgedessen werden Ihnen vorab folgende Informationen geliefert:

Der Montageprozess wird in Teilschritte, hier als Aufgaben bezeichnet, unterteilt. Relevant sind momentan der Antransport und das Fügen eines Bauteils. In Abbildung 73 sehen Sie einen möglichen Montageprozess sowie die dazugehörigen Aufgaben.

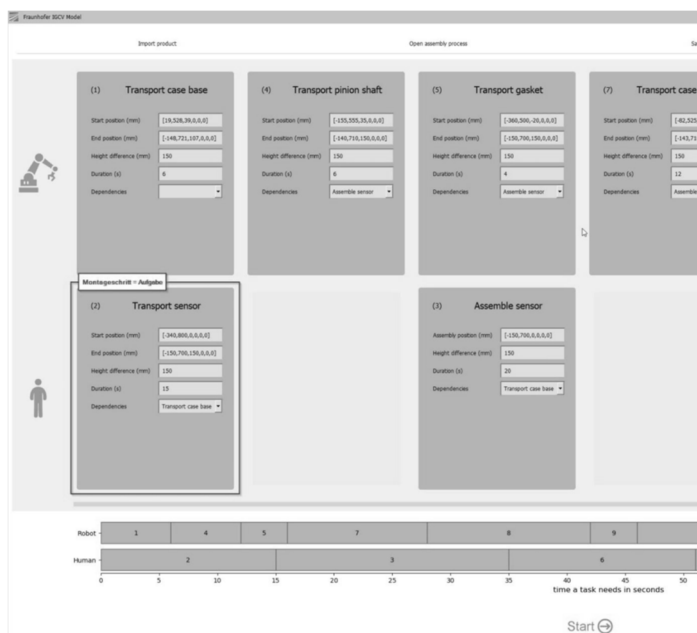


Abbildung 73: Darstellung eines Ausschnitts der Benutzerschnittstelle

Diese Aufgaben sind bereits Roboter und Menschen zugeteilt und richtig sortiert. Ihre Position in den jeweiligen Teilbereichen für Mensch und Roboter (angedeutet durch das entsprechende Symbol) entsprechen nur ungefähr der Realität. Eine korrekte Darstellung der Verhältnisse der Aufgaben zueinander ist in der Übersicht im unteren Bereich der Oberfläche zu sehen. Da das System als Ganzes noch nicht im gewünschten Maße implementiert ist, stellen Sie sich bitte folgende Situation vor:

Sie möchten ein kleines Getriebe (siehe Abbildung 2) mit Hilfe eines Roboters zusammenbauen. Das vorliegende System ermöglicht es Ihnen, den Roboter anhand von Aufgabenblöcken zu programmieren. Ihnen stehen eine GetriebeKlein.CATProduct-Datei für den Import und der darauffolgenden Erstellung eines empfohlenen Montageablaufs bereit. Außerdem ist ein bereits vorhandener Montageablauf unter GetriebeKlein.xml abgespeichert, der ebenfalls in der Oberfläche dargestellt werden kann. Nachdem Sie die Datei importiert bzw. geöffnet haben, wird Ihnen ein vorgeschlagener Montageablauf angezeigt, den Sie jetzt mit Hilfe des Programms nach Ihren Vorstellungen bearbeiten sollen.

Vorschläge:

- Sie brauchen länger für eine Aufgabe, als das System berechnet hat.
- Die Reihenfolge der Montageschritte gefällt Ihnen nicht.
- In der CAD-Datei wurde ein Bauteil bzw. Aufgabe zu wenig/viel eingefügt was sich in dem Ablauf widerspiegelt.
- Sie sind noch nicht fertig mit der Planung des Montageprozesses und möchten ihn morgen vervollständigen.



*Abbildung 74: Explosionszeichnung des betrachteten Produkts*



Beantworten Sie die nachfolgenden Fragen, nachdem Sie die Software (mindestens ein Mal) bedient haben.

5. Der Zweck der Teilbereiche der Oberfläche sind Ihnen klar. Wenn nicht, erklären Sie bitte, was Ihnen unklar ist.

Ja  Nein

<b>6. Die Software...</b>	Trifft gar nicht zu		Weder noch		Trifft voll und ganz zu
...liefert in zureichendem Maße Informationen darüber, welche Eingaben zulässig oder unnötig sind.					
...erleichtert die Orientierung durch eine einheitliche Gestaltung.					
...informiert in ausreichendem Maße über das, was sie gerade macht.					
...lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.					
Zusätzliche Kommentare:					

<b>7. Die Software...</b>	Trifft gar nicht zu		Weder noch		Trifft voll und ganz zu
...liefert gut verständliche Fehlermeldungen.					
...erfordert bei Fehlern im Großen und Ganzen einen geringen Korrekturaufwand.					
...gibt konkrete Hinweise zur Fehlerbehebung.					

Zusätzliche Kommentare:					
<b>8. Die Software...</b>	Trifft gar nicht zu		Weder noch		Trifft voll und ganz zu
...erfordert wenig Zeit zum Erlernen.					
...erfordert nicht, dass ich mir viele Details merken muss.					
...ist gut ohne fremde Hilfe oder Handbuch erlernbar.					
Zusätzliche Kommentare:					

**9.** Bei welchen der genannten Schritte stoßen Sie auf Probleme? Bitte schildern Sie kurz, sofern vorhanden, worin das Problem beim jeweiligen Arbeitsschritt besteht.

Eingabe in Felder mit Auswahllisten	
Ausfüllen von Pflichtfeldern	
Prüfen, ob Speichern erfolgreich war	
Reaktion auf Fehlermeldungen	

<b>10.</b>	Trifft gar nicht zu		Weder noch		Trifft voll und ganz zu
Die Anzahl an Informationen auf dem Bildschirm ist angemessen.					
Die Informationen sind logisch gruppiert.					
Die Informationen befinden sich dort, wo ich sie erwarte.					
Zusätzliche Kommentare:					

<b>11. Wenn Sie Werte von Aufgaben ändern wollen, ist Ihnen das Speichern der Änderungen lieber indem ...</b>	Trifft gar nicht zu		Weder noch		Trifft voll und ganz zu
... Enter gedrückt wird / in die nächste Zeile per Mausklick oder Tab gewechselt wird.					
... Sie auf eine Speichertaste für den jeweiligen Aufgabenblock drücken.					
... Sie auf die Updatetaste drücken, um alle gemachten Änderungen zu übernehmen.					
Zusätzliche Kommentare:					

<b>12. Wenn Sie Werte von Aufgaben ändern wollen...</b>	Trifft gar nicht zu		Weder noch		Trifft voll und ganz zu
... ist dies einfach auszuführen.					
... ist Ihnen klar, welche Schritte dafür nötig sind.					
... wird Ihnen eine Rückmeldung über den Erfolg gegeben.					
... ist Ihnen klar, in welcher Form die Informationen vorliegen müssen.					
Zusätzliche Kommentare:					

<b>13.</b>	Sehr selten				Sehr häufig
Wie häufig kommt es vor, dass Sie eine Eingabe vergessen?					
Zusätzliche Kommentare:					

Warum vergessen Sie manche Eingaben?

Fenster ist verdeckt	
Kein Hinweis des Systems, dass es ein Pflichtfeld ist	
Information/Angabe befindet sich an einem ungewöhnlich Ort	
Reihenfolge der Arbeitsschritte entspricht nicht meiner Vorgehensweise	
Zu viele Schritte zu merken	

## Verzeichnis betreuter Studienarbeiten

Der Autor der vorliegenden Dissertation hat in den Jahren von 2015 bis 2019 unter wesentlicher wissenschaftlicher, fachlicher und inhaltlicher Anleitung die im Folgenden aufgeführten studentischen Arbeiten betreut. Die entstandenen Ergebnisse sind teilweise in das vorliegende Dokument eingeflossen. Der Autor dankt allen Studenten für ihre Unterstützung bei dieser wissenschaftlichen Arbeit und wünscht ihnen alles Gute in ihrer beruflichen Laufbahn als Ingenieur.

*Zamponi, Franziska:* Entwicklung einer Kollisionsvermeidung und berührungslosen Not-Aus Strategie mittels Leap Motion Sensor, Hochschule Augsburg, 20.07.2016 – 20.10.2016

*Rohr, Matthias:* Identifikation und Umsetzung eines Montagevorgangs im Rahmen einer Mensch-Roboter-Kollaboration, Hochschule Augsburg, 20.04.2016 – 20.09.2016

*Reckordt, Tim:* Entwicklung eines Verfahrens zur Erkennung von menschlichen Tätigkeiten in einer Mensch-Roboter, Kooperation, TU München, 13.06.2016 – 13.12.2016

*Pichlmair, Monika:* Kollisionsvermeidung und Bewegungserkennung durch Verknüpfung einer Kinect mit einem LeapMotion-Sensor, Universität Augsburg, 01.10.2016 – 31.05.2017

*Naumann, Maria:* Konzeption und Umsetzung eines Modells zur Ableitung eines Montagevorranggraphen für die Anwendung in der Mensch-Roboter-Kooperation, TU München, 01.02.2017 – 31.07.2017

*Steingröver, Svenja:* Tätigkeitserkennung in der Mensch-Roboter-Kooperation mithilfe von Hidden Markov Modellen, Ostbayerische Technische Hochschule Amberg-Weiden, 01.05.2017 – 31.10.2017

*Weiß, Maximilian:* Konzeption und Umsetzung eines Planungssystems für die aufgabenorientierte Programmierung für die Mensch-Roboter-Kooperation, Hochschule Augsburg, 21.04.2017 – 21.09.2017

*Baranyai, Attila:* Entwicklung einer Benutzerschnittstelle zur aufgabenorientierten Programmierung eines Roboters für die Mensch-Roboter-Kooperation, Hochschule Augsburg, 20.07.2017 – 20.11.2017

<p><i>Hoss, Danilo</i>: Konzeption und Umsetzung eines Programmiermoduls für die aufgabenorientierte Programmierung für die Mensch-Roboter-Kooperation, Universität Augsburg, 07.08.2017 – 07.11.2017</p>
<p><i>Sánchez, Alejandro Ureña</i>: Fähigkeitsabgleich zur Aufgabenzuteilung in der Mensch-Roboter-Kooperation, TU München, 01.03.2017 – 31.08.2017</p>
<p><i>Barisic, Tea</i>: Konzept zur Integration arbeitswissenschaftlicher und sicherheitsrelevanter Kriterien für die Aufgabenzuteilung in der Mensch-Roboter-Kooperation, TU München, 16.02.2018 – 03.08.2018</p>
<p><i>Amereller, Xaver</i>: Anpassung und Umsetzung eines Modells zur Tätigkeitserkennung in der Mensch-Roboter-Kooperation, AKAD Hochschule Stuttgart, 16.05.2018 – 16.11.2018</p>
<p><i>Haas, Natalia</i>: Weiterentwicklung einer Benutzerschnittstelle zur einfachen Programmierung eines Roboters für die Mensch-Roboter-Kooperation, Universität Augsburg, 23.11.2018 – 23.01.2019</p>
<p><i>Ferber, Maximilian</i>: Konzeptionierung und Implementierung eines Systems zur automatisierten Erstellung der Prozessparameter einer anwendungsbezogenen Mensch-Roboter-Kooperation, Hochschule Augsburg, 22.05.2018 – 03.09.2018</p>
<p><i>Göbl, Fabian</i>: Simulation eines Szenarios der Mensch-Roboter-Kooperation im Rahmen der aufgabenorientierten Programmierung, TU München, 01.05.2018 – 02.11.2018</p>
<p><i>Klein, Corina</i>: Konzeptionierung und Implementierung eines durchgängigen Planungsmoduls für die Mensch-Roboter-Kooperation, TU München, 01.05.2018 – 01.11.2018</p>
<p><i>Gerner, Jeremias</i>: Erweiterung eines Modells zur Tätigkeitserkennung in der Mensch-Roboter-Kooperation, Technische Hochschule Georg Simon Ohm, 01.11.2018 – 01.02.2019</p>