

**Entwicklung eines intelligenten
Kognitiven Assistenzsystems
für
dynamische Produktionsumgebungen**

– Am Beispiel eines Assistenzsystems zur Unterstützung von Mitarbeitern in der Nacharbeit –

Dissertation zur Erlangung des Grades des Doktors der Ingenieurwissenschaften der Naturwissenschaftlich-Technischen Fakultät der Universität des Saarlandes

von

Attique Bashir

Saarbrücken 2023

Tag des Kolloquiums: 22. September 2023

Dekan: Prof. Dr. Ludger Santen

Berichterstatter: Prof. Dr.-Ing. Rainer Müller
Prof. Dr.-Ing. Robert Heinrich Schmitt

Akad. Mitglied: Dr. Michael Roland

Vorsitz: Prof. Dr.-Ing. Dirk Bähre

Hinweis

In dieser Arbeit wird aus Gründen der Lesbarkeit das generische Maskulinum verwendet. Der Autor weist darauf hin, dass ausdrücklich alle Geschlechtsidentitäten angesprochen werden.

Inhaltsverzeichnis

I.	Abbildungsverzeichnis	I
II.	Abkürzungsverzeichnis	IV
III.	Formelzeichen.....	V
IV.	Danksagung	VII
1	Zusammenfassung	1
2	Einleitung	2
2.1	Ausgangssituation und Motivation.....	2
2.2	Ziel der Arbeit.....	3
2.3	Aufbau der Arbeit	6
3	Grundlagen des Betrachtungsbereichs und bestehende Forschungsansätze	9
3.1	Montage und Montagesequenzen.....	9
3.1.1	Montagesequenzen	10
3.1.2	Verfahren zur Generierung von Verbaureihenfolgen	10
3.2	Design for Assembly und Design for Disassembly	16
3.3	Einordnung und Notwendigkeit der Nacharbeit	17
3.3.1	Abgrenzung und Einordnung der Nacharbeit.....	18
3.3.2	Notwendigkeit für die Nacharbeit	21
3.3.3	Diskussion	22
3.4	Digitale Assistenzsysteme.....	23
3.4.1	Domainenübergreifende Betrachtung von Assistenzsystemen (Unterstützungssystemen) nach Weidner.....	23
3.4.2	Beschreibung von Assistenzsystemen nach Wandke.....	25
3.4.3	Klassifikation von digitalen Assistenzsystemen nach Apt	28
3.4.4	Intelligente Tutorielle Systeme.....	29
3.4.5	Montageassistenzsysteme	30
3.4.6	Diskussion	31
3.5	Gestaltung Digitaler Assistenzsysteme unter Berücksichtigung menschlicher kognitiver Leistungen	32
3.5.1	Theorie zur Kognitiven Belastung – Cognitive Load Theory	33
3.5.2	Baddeley's Modell des Working Memory	34
3.5.3	Multiple - Resource Modell	35
4	Theoretische Grundlagen	37
4.1	Graphentheorie	37
4.1.1	Grundbegriffe der Graphentheorie.....	37

4.1.2 Arten von Graphen	39
4.1.3 Darstellungsformen und Datenstrukturen von Graphen.....	40
4.2 Metaheuristiken zur Lösung von Optimierungsproblemen	41
4.3 Wartesysteme und Warteschlangentheorie.....	47
4.3.1 Grundaufbau und Notation in einem Wartesystem	48
4.3.2 Modellierungsmethoden der Ankunfts- und Abfertigungsrate	49
5 Anforderungen an den kognitiv assistierenden Nacharbeitsplatz	53
5.1 Ablauf in der Nacharbeit.....	53
5.2 Anforderungen an das Assistenzsystem im Kontext der Nacharbeit..	55
6 Entwicklung des Digitalen Kognitiven Assistenzsystems für die Nacharbeit	56
6.1 Entwicklung eines herkömmlichen Assistenzsystems	56
6.1.1 Theoretische Umsetzung.....	56
6.1.2 Praktische Umsetzung der Assistenzfunktion	60
6.2 Weiterentwicklung des Kognitiven Assistenzsystems für dynamische Produktionsbereiche	75
6.2.1 Stand der Technik.....	75
6.2.2 Theoretische Umsetzung.....	81
6.2.3 Praktische Umsetzung der Assistenzfunktion	90
6.3 Weiterentwicklung des Kognitiven Assistenzsystems zur Neubewertung der Fehlerursache (Routing)	96
6.3.1 Theoretische Umsetzung.....	96
6.3.2 Praktische Umsetzung der Assistenzfunktion	98
6.3.3 Diskussion	98
6.4 Weiterentwicklung des Kognitiven Assistenzsystems zur Realisierung der Zustandssynchronisation	99
6.4.1 Stand der Technik.....	101
6.4.2 Theoretische Umsetzung.....	112
6.4.3 Praktische Umsetzung der Assistenzfunktion	117
6.5 Zusammenfassung der digitalen Modellierung eines Assistenzsystems zur Unterstützung in der Nacharbeit.....	117
7 Aufbau der Steuerung des Assistenzsystems	120
8 Optimierte Auftragszuweisung in der Nacharbeit.....	123
8.1 Betrachtung und Auswahl relevanter Lagersysteme	123
8.2 Modellierung der Zugriffsbeschränkung auf den Auftragsbestand ...	124
8.2.1 Modellierung der Zugriffs kombinationen.....	125
8.2.2 Modellierung der Zuweisungskombination.....	130
8.3 Zuweisung der Reparaturaufträge an verfügbare Mitarbeiter	133
8.3.1 Ermittlung von Näherungslösungen mit Hilfe des Genetischen Algorithmus.....	133

8.3.2 Ermittlung von Näherungslösungen mit Hilfe des Simulated Annealing.....	136
8.4 Prozessvertrautheit der Mitarbeiter	137
8.5 Fallspezifische Anwendung der Metaheuristiken	138
8.5.1 Kontinuierliche Auftragszuweisung	138
8.5.2 Optimierte Auftragszuweisungen zur Variantendurchschleusung..	138
8.6 Zusammenfassung der Methode zur optimierten Auftragszuweisung	139
9 Erprobung und Validierung der Methode in einer Pilot- und Simulationsumgebung	140
9.1 Betrachtung des Standardfalls der Nacharbeit.....	141
9.2 Berücksichtigung der Erfahrungsgrade des Mitarbeiters.....	143
9.3 Betrachtung Neuermittlung des Nacharbeitsplans	144
9.4 Betrachtung der Abweichung vom vorgesehenen Nacharbeitsplan .	146
9.4.1 Trainingsphase der vierten Entwicklungsstufe in einer Simulationsumgebung	146
9.4.2 Betriebsphase der vierten Entwicklungsstufe	151
10 Simulation und Validierung der Auftragszuweisung.....	156
10.1 Aufbau der Simulationsumgebung	156
10.1.1 Modellierung der Auftragsklasse	159
10.2 Ergebnisse mit Genetischem Algorithmus.....	162
10.3 Ergebnisse mit Simulated Annealing.....	163
10.4 Ergebnisse der Variantendurchschleusung.....	168
11 Zusammenfassung und Ausblick.....	170
11.1 Zusammenfassung.....	170
11.2 Ausblick.....	172
12 Literaturverzeichnis.....	173
13 Wissenschaftliche Arbeiten	183
13.1 Eigene Veröffentlichungen	183
13.2 Betreute Arbeiten	184

I. Abbildungsverzeichnis

Abbildung 1: Betrachtungsbereich der vorliegenden Arbeit	6
Abbildung 2: Entwicklungsstufen des Assistenzsystems	7
Abbildung 3: Beispielprodukt nach Bourjault	11
Abbildung 4: Relationales Modell als Graph	13
Abbildung 5: Verbindungsgraph	13
Abbildung 6: Machbare Schnittpunkte zur Generierung der Cut-Sets	14
Abbildung 7: Einordnung des Remanufacturing	20
Abbildung 8: Ausprägung der Produktwiederverwertung.....	21
Abbildung 9: Ganzheitliche Systematik von Unterstützungssystemen	24
Abbildung 10: Domainenübergreifende Systematik für Unterstützungssysteme	25
Abbildung 11: Stufen der Assistenz.....	26
Abbildung 12: Klassifizierungsansatz für (Digitale) Assistenzsysteme	28
Abbildung 13: Informationsschnittstellen in Mensch-Maschine-Systemen	33
Abbildung 14: Drei Belastungsbereiche des Gedächtnisses	34
Abbildung 15: Komponenten des Arbeitsgedächtnisses.....	35
Abbildung 16: Multiple-Resource Modell zur Beurteilung der Aufmerksamkeitslast .	36
Abbildung 17: Einfacher Graph.....	37
Abbildung 18: Graphenumwandlung:	38
Abbildung 19: Auswahl unterschiedlicher Graphenarten:	39
Abbildung 20: Darstellungsformen von Graphen	40
Abbildung 21: Phasen des Genetischen Optimierungsalgorithmus	42
Abbildung 22: Auswahl von Crossover Methoden für Listen bzw. Zeichenketten.....	44
Abbildung 23: Wahrscheinlichkeitsverlauf	46
Abbildung 24: Aufbau und Elemente eines einfachen Wartesystems.....	48
Abbildung 25: Poissonverteilung für eine durchschnittliche Ankunftsrate $\lambda = 5$	51
Abbildung 26: Erlangverteilung.....	52
Abbildung 27: Zusammenhang zwischen Serienmontage und Nacharbeitsbereich .	54
Abbildung 28: Schematische Darstellung eines Nacharbeitsplans	57
Abbildung 29: Schema des Nacharbeitsplanes als Prozesskette	59
Abbildung 30: Pilotarbeitsplatz als Assistenzsystem	61
Abbildung 31: Beispiel eines industriell genutzten Mitarbeiterführungssystems [119 S.184]	63
Abbildung 32: Grundansicht des entwickelten Mitarbeiterführungssystems	64
Abbildung 33: Nachrichtenaustausch zwischen Mitarbeiter und Assistenzsystem ...	65
Abbildung 34: Transformation von Tischkoordinaten auf Beamerkoordinaten	67
Abbildung 35: Waage zur Kontrolle der Bauteilentsorgung	70
Abbildung 36: Intelligente Kommissionier-Box.....	71
Abbildung 37: Paarweise Generierung der JSON-Datei.....	72
Abbildung 38: Mitarbeiterqualifikation in der Datenbank.....	74
Abbildung 39: Pathfinding mit dem Dijkstra Algorithmus	77
Abbildung 40: Floyd-Warshall-Algorithmus:.....	80

Abbildung 41: Zwischenknoten des Floyd-Warshall Algorithmus	81
Abbildung 42: Redundanter Produktzustandsgraph als Ergebnis des Algorithmus ..	83
Abbildung 43: Zusammenführung der Knoten	84
Abbildung 44: a) Zeigt den Produktzustandsgraph b) zugehöriger Kantengraph	85
Abbildung 45: Produktzustandsgraph des Vorderachsantriebs des Getriebes	92
Abbildung 46: Erstellung der Adjazenzmatrix im Tabellenkalkulationsprogramm	93
Abbildung 47: Halbautomatisiert generierter Produktzustandsgraph	93
Abbildung 48: Dateien als Bestandteil des Produktzustandsgraphen	94
Abbildung 49: Anlegen des Produktzustandsgraphen	95
Abbildung 50: Konstruktion des Nacharbeitsplans	98
Abbildung 51: Schematische Funktionsweise	100
Abbildung 52: Intuitive Betrachtung der drei Probleme	104
Abbildung 53: Beispiel eines Hidden-Markov-Modells	105
Abbildung 54: Beispiel einer Folge von Beobachtungen	106
Abbildung 55: Beispiel einer Folge von Hidden States (versteckten Zuständen)	106
Abbildung 56: Permutative Zusammenstellungsmöglichkeiten	109
Abbildung 57: Wahrscheinlichkeiten für Zustandswechsel	110
Abbildung 58: a) Produktzustandsgraph b) Erweiterter Produktzustandsgraph	115
Abbildung 59: Vorgehensweise zur Befähigung eines Kognitiven Assistenzsystems	119
Abbildung 60: Sequenzdiagramm der Standardkommunikation des Assistenzsystems in der zweiten Entwicklungsstufe	121
Abbildung 61: Sequenzdiagramm des Standardablaufs der Entwicklungsstufe vierter Stufe	122
Abbildung 62: Layout von Bodenlagersystemen	124
Abbildung 63: Schematische Darstellung des Materialflusses	125
Abbildung 64: Entnahmemöglichkeiten aus einer Bereitstellungsfläche	126
Abbildung 65: a) Bereitstellungsfläche b) Stapel c) Stars and Bars	127
Abbildung 66: Zugriffsmöglichkeiten bei der Vergabe von vier Aufträgen.	128
Abbildung 67: Stars and Bars auf fixen Stellen positioniert.	128
Abbildung 68: Ableitung der Anzahl zu entnehmender Aufträge	130
Abbildung 69: (Auftrags-) Zuweisungskombination an verfügbare Mitarbeiter	131
Abbildung 70: Graphenbasierte Darstellung der Bereitstellung	133
Abbildung 71: Die Entnahmekombination	134
Abbildung 72: Schematisches Beispiel zur Suche im Kombinationsraum	135
Abbildung 73: Beispiel für die Zugriffskombination und die Zuweisungskombination	136
Abbildung 74: Möglichkeiten zur Generierung von Nachbarlösungen	137
Abbildung 75: Möglichkeiten der Auftragsentnahme	139
Abbildung 76: Das betrachtete Produkt	140
Abbildung 77: Eingabemaske vor Beginn der Auftragsbearbeitung	141
Abbildung 78: Gegenstand der Betrachtung: Baugruppe Zwischengehäuse- Verteilergehäuse	142

Abbildung 79: Eingabe eines neuen Fehlercodes	144
Abbildung 80: Ausgabe des neuen Arbeitsplans	145
Abbildung 81: Erstellung des Fehlercodegraphen	148
Abbildung 82: a) Ideale Transitionsmatrix b) Ideale Beobachtungsmatrix	149
Abbildung 83: Abweichung der trainierten Matrizen und der idealen Matrizen	150
Abbildung 84: Beobachtungs- und Transitionswahrscheinlichkeiten vor und nach dem Training.....	151
Abbildung 85: Verteilung des Belief State	154
Abbildung 86: Die Verteilung des Belief State nach Mitarbeiter-Rückmeldung	155
Abbildung 87: Stochastische Betrachtungsweisen	157
Abbildung 88: Aufbau der Klasse des Nacharbeitsauftrags.....	160
Abbildung 89: Aufbau der Klasse der Bereitstellungsfläche	160
Abbildung 90: Matrizenmultiplikation zur Berechnung der Stati.....	162
Abbildung 91: Vergleich zwischen Fitnessverläufen.....	163
Abbildung 92: Lösung auf Basis von Simulated Annealing.....	164
Abbildung 93: Vergleich der Lagerbestände bei zufälliger Zuweisung	165
Abbildung 94: Lösung auf Basis von Simulated Annealing.....	166
Abbildung 95: Vergleich der Lagerbestände im priorisierten Fall.....	167
Abbildung 96: Vergleich der Lagerbestände	168
Abbildung 97: Beispielhafter Vergleich	169

II. Abkürzungsverzeichnis

Kurzform	Langform
AP	Alternating Position Crossover
AR	Augmented Reality
API	Application programming interface
CAD	Computer Aided Design
FIFO	First-in-First-out
GIF	Graphics Interchange Format
HMM	Hidden Markov Modell
i.O.	in Ordnung
JSON	JavaScript Object Notation
LIFO	Last-in-First-out
MTM	Method Time Measurement
n.i.O.	nicht in Ordnung
SIRO	Selection-in-Random-Order
SQL	Structured Query Language
TCP	Transmission Control Protocol
VR	Virtual Reality

III. Formelzeichen

Zeichen	Bedeutung	Zusammenhang
\cap	Schnittmenge	Graphen
\cup	Vereinigung	Graphen
\subseteq	Teilmenge	Graphen
V	Menge der Knoten (Vertex)	Graphen
v	Ein Knoten	Graphen
E	Menge der Kanten (Edges)	Graphen
e	Eine Kante	Graphen
α	Wahrscheinlichkeit einer Beobachtungsfolge bis Zustand S	Markov-Modell
β	Wahrscheinlichkeit einer Beobachtungsfolge ab Zustand S	Markov-Modell
π	Anfangswahrscheinlichkeit	Markov-Modell
ξ	Zustandsübergangswahrscheinlichkeit	Markov-Modell
γ	Aufenthaltswahrscheinlichkeit	Markov-Modell
t	Zeitschritt	Markov-Modell
i	Hidden State i	Markov-Modell
j	Hidden State j (meist folgend auf i)	Markov-Modell
N	Anzahl der Hidden States	Markov-Modell
A	Übergangsmatrix im HMM	Markov-Modell
T	Übergangsmatrix im POMDP	Markov-Modell
B	Beobachtungsmatrix im HMM	Markov-Modell
O	Beobachtungsmatrix im POMDP	Markov-Modell
γ	Discount-Faktor	Markov-Modell
λ	Ankunftsrate	Wartesystem
μ	Abfertigungsrate	Wartesystem
$P()$	Wahrscheinlichkeit	Allgemein
$K()$	Kostenfunktion	Wartesystem
$E()$	Energiefunktion / Kostenfunktion im Simulated Annealing	Wartesystem

M_v	Liste Verfügbarer Mitarbeiter	Wartesystem
m_i	i-te Mitarbeiter aus M_v	Wartesystem
p_j	Prozess aus Nacharbeitsplan mit dem Index j	Wartesystem
O	Auftrag/ Order	Wartesystem
o_i	i-te Auftrag aus O	Wartesystem

IV. Danksagung

Während meiner Tätigkeit am Zentrum für Mechatronik und Automatisierungstechnik (ZeMA) habe ich viele tolle Menschen kennengelernt, ohne die der Weg zur Dissertation sicherlich nicht in einer solch freundschaftlichen und produktiven Atmosphäre verlaufen wäre. Leider können nicht alle namentlich genannt werden, die auf eine ganz besondere Art und Weise zum Erfolg dieser Arbeit beigetragen haben, trotzdem sollen zumindest einige Personen Erwähnung finden.

Zuallererst möchte ich mich bei meinem Doktorvater Herrn Prof. Dr.-Ing. Rainer Müller für die Chance, diese Arbeit verfassen zu dürfen sowie das mir entgegengebrachte Vertrauen bedanken. Die intensive Betreuung der Arbeit, das offene Ohr für jedwede Ideen sowie die Anmerkungen habe ich als Privileg betrachtet und die wertvollen Gespräche stets als lehr- und aufschlussreich empfunden. Der hilfreiche Input hat mir sicherlich den ein oder anderen Umweg erspart.

Zudem möchte ich Herrn Prof. Dr.-Ing. Robert Schmitt vom Werkzeugmaschinenlabor WZL der RWTH Aachen für das Interesse an dieser Arbeit und die Übernahme des Zweitgutachtens herzlich danken.

Weiter möchte ich mich bei meinen Arbeitskollegen am ZeMA für die gute Zusammenarbeit bedanken, insbesondere bei Herrn Leenhard Hörauf und Christoph Speicher. Sie waren stets offen für neue Ideen und haben mich in meiner Forschungstätigkeit stets gefördert. Zudem Danke ich Herrn Dirk Burkhard, Ali Kanso und Martin Karkowski. Der fachliche Austausch war sehr konstruktiv und ideenreich. Die Diskussionen haben mir oft neue Perspektiven eröffnet und zu kreativen Lösungen beigetragen.

Ein ganz herzlicher Dank gilt den wissenschaftlichen Hilfskräften sowie den Abschlussarbeitern. Besonders hervorheben möchte ich Herrn Daniel Becker, der durch sein außerordentliches Interesse und seine besondere Hingabe zum Projekterfolg beigetragen hat.

Ich bedanke mich bei meiner Familie für den Rückhalt und speziell bei meinen Eltern, die mir alle Wege eröffnet haben.

Gewidmet meiner Frau Amy.

1 Zusammenfassung

Deutsch

Kognitive Assistenzsysteme in der Produktion unterstützen Mitarbeiter bei der Bewältigung manueller Tätigkeiten. Im laufenden (Montage-)Prozess versorgen sie den Mitarbeiter mit Informationen zur auszuführenden Tätigkeit. Fortschrittlichere Assistenzsysteme überprüfen zugleich den Prozess und melden etwaige Fehler zurück. Herkömmliche Kognitive Assistenzsysteme eignen sich für lineare Prozessabfolgen, wie sie bspw. in der Linienmontage vorzufinden sind. In Produktionsbereichen, wo Prozesse von unterschiedlichen Einflussfaktoren abhängig sind, eignen sich bisherige Assistenzsysteme kaum. In dieser Arbeit wird die Entwicklung eines Kognitiven Assistenzsystems für den Einsatz in dynamischen Produktionsbereichen am Beispiel der Nacharbeit beschrieben. Grundlage hierfür ist eine Graphenstruktur, die den Produktfortschritt abbildet. Diese wird mit unterschiedlichen Prozessdaten angereichert und erlaubt die Generierung einer Prozessliste mit variablem Ziel. Diese Liste steht in digitaler Form zur Verfügung und steuert das Kognitive Assistenzsystem an. Zudem werden am Assistenzsystem Daten aufgenommen. Diese ermöglichen den Rückschluss auf die Vertrautheit einer Person mit bestimmten Montageprozessen. Diese Information wird genutzt, um Aufträge den Mitarbeitern derart zuzuweisen, dass Auftragsbestände besser reduziert werden. Hierzu werden Metaheuristiken genutzt, damit auch eine Vielzahl von Kombinationsmöglichkeiten (Auftrag an Mitarbeiter) zu berücksichtigen ist.

English

Cognitive Assistance Systems used in production provide operators with valuable information about the assembly process to assist in decision-making. Some systems can even check processes and give feedback in case of any errors. Commonly available assistance systems work well in sequentially organized processes like those of an assembly line. However, they are barely able to operate in more dynamic environments such as rework areas, where the processes depend on numerous factors. This work aims to describe a Cognitive Assistance System suitable for use in these dynamic production environments. A graph structure poses as the backbone for this Cognitive Assistance System as it represents the production stream. This graph structure is then enriched with process-related data and is able to generate a digital process list that serves as the program for the Cognitive Assistance System. The data collected by the system is used to determine the operator's familiarity with the task to be performed. The information is then used to assign a suitable rework job to the operator in order to reduce the amount of pending rework jobs. Meta-heuristics help to handle the huge number of combinations (job-to-operator) that this system would generate.

2 Einleitung

Key Wörter: Kognitive Assistenzsysteme, Lernende Systeme, Optimierung

2.1 Ausgangssituation und Motivation

Produzierende Unternehmen stehen im fortwährenden Wettbewerb miteinander und müssen auch global mit anderen Unternehmen konkurrieren. Um den Anschluss an den Weltmarkt halten oder ausbauen zu können, ist es ein ständiges Ziel, Optimierungspotenziale in wertschöpfenden sowie angrenzenden Bereichen zu identifizieren und zu realisieren. Der herkömmliche Ansatz orientiert sich an den Lehren optimierter Abläufe in der Produktion wie z. B. dem Toyota Production System. Die gängige Praxis in der Optimierung ist die Minimierung von Verschwendung. Dabei werden nach Taiichi Ōno 7 Arten der Verschwendung (Muda) benannt: Überproduktion (Over-Production), Bestände (Inventory), Transport (Transportation), Fehler (Defects), Bewegungen (Motion), Wartezeiten (Waiting), Verarbeitung (Over-Processing). [Vgl. 1 S.280f., Vgl. 2 S. 319]

Zur heutigen unternehmerischen Realität gehört das Vorkommen von Verschwendung, selbst wenn Optimierungen erzielt werden. Gerade Produktfehler oder Qualitätsfehler tauchen permanent in der Produktion auf und drängen Unternehmen zum Handeln. Dies kann in Form von Verschrottung, Vertrieb zum geringeren Preis oder Nacharbeit mit anschließendem Verkauf geschehen. Auch in diesem Bereich können noch weitere Optimierungen erzielt werden. Teilweise geschieht dies mit Prinzipien, die bereits in der Serienmontage erfolgreich angewendet werden. Wird ein Produkt in der Endkontrolle als fehlerhaft erkannt, so stellt dies für die Serienmontage eine Form der Verschwendung dar. Denn zum einen war die Produktionszeit nicht wertschöpfend, zum anderen wurden Betriebsmittel und andere Ressourcen ineffektiv in Anspruch genommen. In der Montage begegnet man Fehlerquellen durch (punktuelle) Einleitung von Fehlerabstellmaßnahmen.

Global betrachtet wird der Produktfehler je nach Komplexität unabhängig vom Entstehungsort in einer Nacharbeitsstation oder im Nacharbeitsbereich behoben. Nacharbeitsprozesse sind häufig nicht im Fokus von Optimierungsmaßnahmen. Das liegt zum einen daran, dass Fehler möglichst am Entstehungsort behoben werden sollen und zum anderen an der Komplexität der Prozesse und Nacharbeitsumfänge. Die Komplexität der Prozesse wird der menschlichen Entscheidungsfähigkeit und Flexibilität überlassen. Da viele Prozesse in Mitarbeiterverantwortung liegen, sind die Prozesse im Vergleich zu automatisierten Prozessen oft fehleranfälliger. Einerseits herrscht im Nacharbeitsbereich oft nur unzureichende Transparenz, andererseits sind die Prozesse aufgrund des geringen Automatisierungsgrades im Vergleich zur Serienmontage tendenziell langsamer. Zudem ist für die Nacharbeit ein sehr gut geschultes Personal notwendig, dessen Abwesenheit nicht ohne Weiteres kompensiert werden kann. Oftmals führen ineffiziente Prozesse zu langsameren Bearbeitungszeiten, die sich in steigenden Beständen niederschlagen.

Punktuell bestehen Optionen, den Nacharbeitsbereich durch Methoden und Techniken aus der Serienmontage zu optimieren. Allerdings müssen die Randbedingungen der Nacharbeit berücksichtigt werden. Assistenzsysteme bieten die Möglichkeit, Mitarbeiter an Handarbeitsplätzen zu unterstützen. Die Systeme können zur physischen und psychischen Entlastung des Mitarbeiters beitragen. Allerdings sind gerade Assistenzsysteme zur kognitiven Entlastung des Mitarbeiters (sogenannte Kognitive Assistenzsysteme), die zur Entscheidungsfindung verhelfen, im Kern mit einer Automatisierung von Prozessen vergleichbar. Ähnlich wie Automaten müssen auch diese ereignisbezogen (re-)agieren. Die wenigsten kommerziell verfügbaren Assistenzsysteme sind in der Lage, diese Komplexität so abzubilden, dass das System auch gepflegt werden kann. Je mehr die Umwelt von äußeren Einflüssen abhängig ist, desto komplizierter können Automatisierungslösungen sein.

2.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist die Beantwortung der folgenden wissenschaftlichen Frage:

Wie können in dynamischen Montagebereichen Kognitive Assistenzsysteme den Mitarbeiter unterstützen und Aufträge unter Berücksichtigung der Mitarbeitererfahrung so zugewiesen werden, dass die Produktivität erhöht wird?

Diese kann in drei Teilfragen untergliedert werden:

1. *Wie kann ein Kognitives Assistenzsystem für den Einsatz in dynamischen Produktionsbereichen befähigt werden?*
2. *Wie kann das Kognitive Assistenzsystem auf die Bedürfnisse des Mitarbeiters in dynamischen Produktionsbereichen angepasst werden?*
3. *Wie kann ein Kognitives Assistenzsystem über die herkömmliche Mitarbeiterassistenz hinaus die Produktivität steigern?*

Ziel der Arbeit ist die Entwicklung und Umsetzung eines lernenden Kognitiven Assistenzsystems, das für den Einsatz in dynamischen Montagebereichen geeignet ist. Ein Kognitives Assistenzsystem bezeichnet vor diesem Hintergrund einen Arbeitsplatz, der den Mitarbeiter bei der Entscheidungsfindung während der Montage oder Nacharbeit unterstützt. Die Entscheidungsunterstützung umfasst den auszuführenden Montageprozess, kann aber auch Handlungen nahe am Montageprozess wie etwa das Reporting beinhalten. Die Funktion des Assistenzsystems erfordert die Beobachtung der Produkt- bzw. Prozesszustände. Die Daten können genutzt werden, um weitere Optimierungen durchzuführen. Daher ist ein weiteres Ziel der Arbeit die Verwertung der Daten des Assistenzsystems zur Steigerung der Produktivität am Beispiel der Auftragszuweisung und die damit erwirkte erhöhte Durchschleusung von Aufträgen.

Zu Beginn sollen einige Begriffe aus dem Titel der Arbeit näher erläutert und im Kontext dieser Arbeit erklärt werden.

Der Ursprung des Begriffs **Kognition** liegt im lateinischen Begriff *cognoscere* und kann mit „erkennen“, „erfahren“ oder „kennenlernen“ übersetzt werden. [3 S.1]

Kognitive Assistenzsysteme unterstützen den Mitarbeiter in der Fertigung, Montage und Logistik und führen somit zu Effizienzsteigerungen. Zudem können hiermit Einlernphasen verkürzt und Fehler vermieden werden. [4]. Oftmals handelt es sich dabei um (mobile) Arbeitsplatzsysteme, die dem Mitarbeiter Informationen situationsgerecht präsentieren. Jedoch wird dieser Begriff häufig unscharf genutzt. Da Kognitive Assistenzsysteme ein Schwerpunkt dieser Arbeit sind, wird dem Thema ein gesondertes Kapitel gewidmet.

Der Duden beschreibt das **Lernen** in unterschiedlicher Art und Weise [5]:

„**sich Wissen, Kenntnisse aneignen**“

„**sich, seinem Gedächtnis einprägen**“

„**im Laufe der Zeit [durch Erfahrungen, Einsichten] zu einer bestimmten Einstellung, einem bestimmten Verhalten gelangen**“

Da im Englischen der Begriff Learning mehrere Bedeutungen haben kann wie etwa „kennenlernen“ bzw. „erfahren“ werden lediglich die Definitionen betrachtet, die dem deutschen Verständnis des Lernens nahekommen.

Im Oxford Learners Dictionary wird das Lernen wie folgt beschrieben:

„**to gain knowledge or skill by studying, from experience, from being taught, etc.**“ [6]

Die Definitionen lassen eine bestimmte Tendenz erkennen. Das Lernen beschreibt einen Prozess, sich Fähigkeiten anzueignen, der eine bestimmte Zeit benötigt. Dieser Prozess erfordert allerdings (externen) Informationsinput.

Ein **Lernendes System** stellt ein System mit der Fähigkeit dar, Informationen zu verarbeiten und zukünftig innerhalb eines bestimmten Rahmens auf neue, aber vergleichbare Probleme anzuwenden. Die „Plattform Lernende Systeme“ definiert Lernende Systeme wie folgt:

„**Lernende Systeme sind Maschinen, Roboter und Softwaresysteme, die abstrakt beschriebene Aufgaben auf Basis von Daten, die ihnen als Lerngrundlage dienen, selbstständig erledigen, ohne dass jeder Schritt spezifisch vom Menschen programmiert wird [...] Lernende Systeme basieren auf Methoden der Künstlichen Intelligenz (KI), genauer: des maschinellen Lernens. Vor allem durch die Fortschritte im Deep Learning entwickelten sich Lernende Systeme in den letzten Jahren zum dynamischsten Bereich der KI-Forschung und -Anwendung**“ [7]

Untersucht man den Begriff „**Dynamik**“, stellt man schnell fest, dass die meisten Definitionen hauptsächlich auf die Interpretation hinsichtlich einer Subdisziplin der Physik beschränkt sind. Das zugehörige Adjektiv **dynamisch** hingegen besitzt sehr vielfältige

Definitionen. Zwar ist auch hier die Krafteinwirkung bedeutsam, allerdings wird diese oft auch als antreibende Kraft mit motivierendem Charakter verstanden.

Dynamische Produktionsumgebung

Ein dynamischer Produktions- bzw. Montagebereich bezeichnet einen Arbeitsbereich, in dem ein Produkt zusammengesetzt wird, die einzelnen Abläufe sich aber von den gegebenen Umständen beeinflussen lassen. Zum Teil kann eine neue Information den Prozessablauf beeinflussen. Somit ist nicht von Beginn an klar, welcher Prozessablauf letztendlich gewählt wird. Der Nacharbeitsbereich stellt bspw. einen dynamischen Arbeitsbereich dar. Zum einen sind die Montageprozessesequenzen abhängig vom vorliegenden Fall, zum anderen verfügt der Mitarbeiter innerhalb gewisser Grenzen über Freiheiten in den Prozessabläufen. Herkömmliche Assistenzsysteme setzen einen vordefinierten und gleichbleibenden Prozessablauf voraus. Das in der Arbeit beschriebene Assistenzsystem hingegen soll jedoch gewisse Abweichungen zulassen und diese auch erfassen. Daher muss das Assistenzsystem ein Lernendes System darstellen, da die Lernfähigkeit die Grundlage für die Einsatzfähigkeit in dynamischen Produktionsumgebungen darstellt.

In dieser Arbeit werden ebenfalls die Methoden beschrieben, die das System zum Einsatz in dynamischen Umgebungen befähigen. Beginnend beim Produkt und der Produktanalyse über die Erstellung der Verbaureihenfolgen sowie des Vorranggraphen werden Datenstrukturen geschaffen, die die software- und datentechnische Basis des Assistenzsystems bilden. Es werden des Weiteren die entsprechenden Betriebsmittel zur Assistenz des Mitarbeiters im Prozess betrachtet. Diese umfassen u. a. Betriebsmittel zur Information des Mitarbeiters, aber auch Betriebsmittel zur Prozessausführung und Prozessprüfung. Das Referenzprodukt für die Validierung ist ein Automatgetriebe für den PKW-Antrieb, wodurch die Methode nur auf eine Auswahl an Produkten (z. B. aus dem Maschinenbau) wirtschaftlich abzielt.

Zudem beschreibt die Arbeit eine Methode, mit der Nacharbeitsprodukte mithilfe der erfassten Daten aus dem Assistenzsystem effizient durch die Nacharbeit durchgeschleust werden können. Dies führt zur Reduzierung von Beständen. Hierzu werden Optimierungsverfahren in Form von Metaheuristiken entwickelt und angewendet, die die Restriktionen der Auftragsentnahme aus einer Bereitstellungsfläche sowie die Möglichkeiten der Auftragszuweisung berücksichtigen.

Arbeitswissenschaftliche, psychologische bzw. soziologische Betrachtungen der Assistenz sind ausdrücklich nicht Gegenstand der Arbeit. Die Arbeit beschränkt sich auf die rein technische Betrachtung des Assistenzsystems, obgleich der Mensch in der Entwicklung Berücksichtigung findet.

Abbildung 1 zeigt den exemplarischen Anwendungsbereich des zu entwickelnden Kognitiven Assistenzsystems. Dieser befindet sich im Nacharbeitsbereich und umfasst neben der Assistenz des Mitarbeiters auch die optimierte Zuweisung der nach dem Prüfvorgang eingelagerten Getriebe.

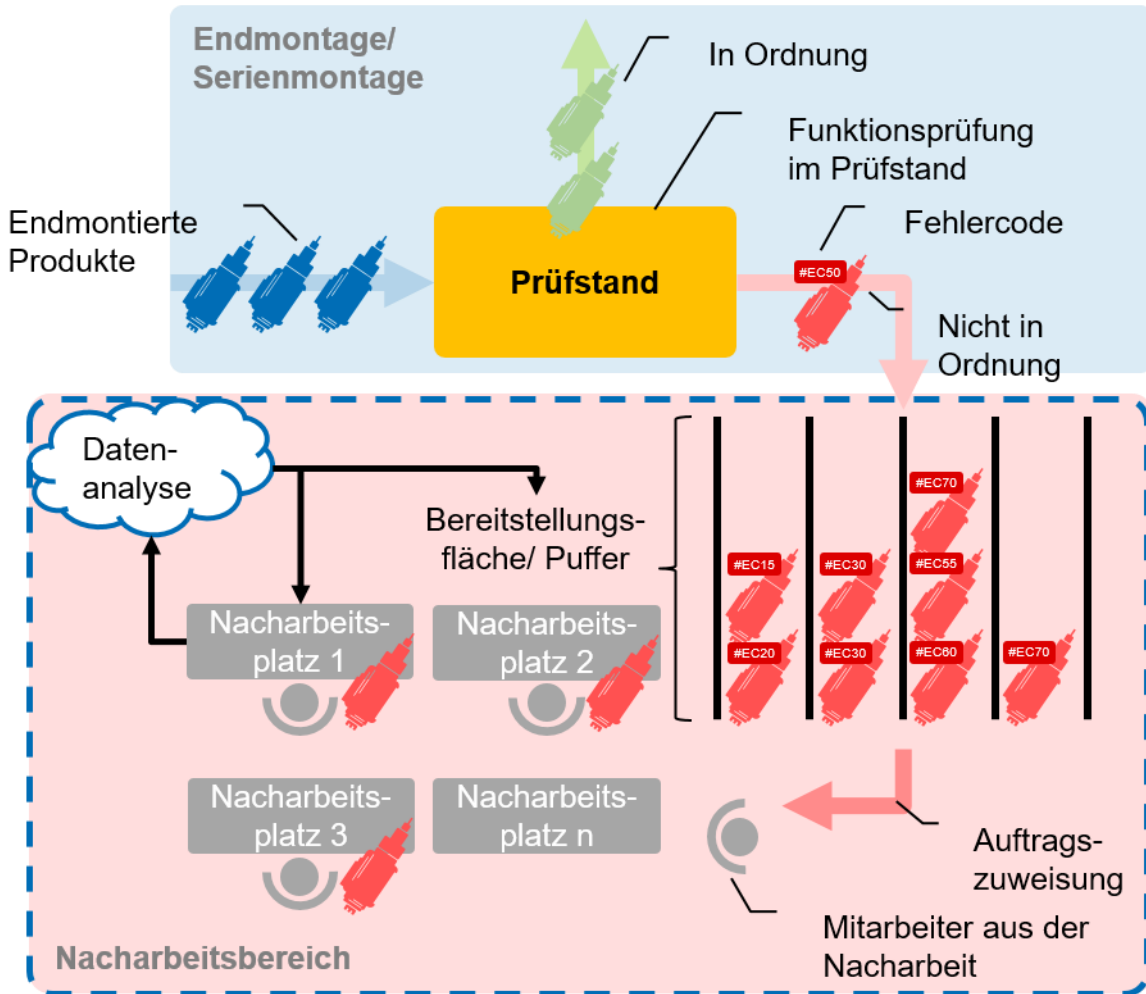


Abbildung 1: Betrachtungsbereich der vorliegenden Arbeit [Eigene Darstellung]

2.3 Aufbau der Arbeit

Die vorliegende Arbeit beschreibt die Entwicklung eines Digitalen Kognitiven Assistenzsystems für Montageaufgaben in dynamischen Produktionsbereichen. Als Anwendungsfall dient der Nacharbeitsbereich, der durch vielfältige Nacharbeitsaufträge sowie durch einen geringen Automatisierungsgrad gekennzeichnet ist. Die Entwicklung erfolgt in vier aufeinander aufbauenden Schritten, die die Fähigkeiten des Assistenzsystems erweitern, wie in Abbildung 2 dargestellt.

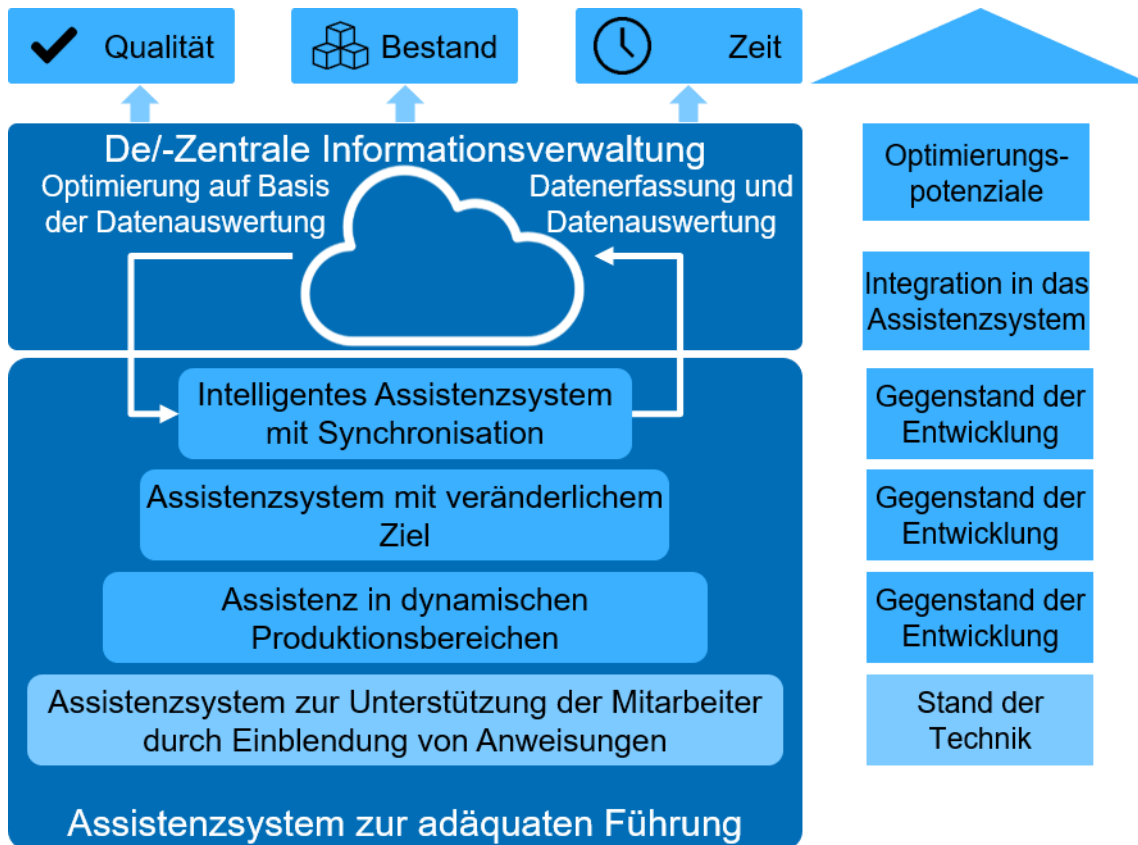


Abbildung 2: Entwicklungsstufen des Assistenzsystems [Eigene Darstellung]

Erste Entwicklungsstufe: Zu Beginn der Entwicklung wird ein Assistenzsystem aufgebaut, das dem bisherigen Stand der Technik entspricht. Die wesentliche Funktion besteht dabei in der Einblendung von Arbeitsanweisungen pro Prozessschritt. Die Datenstruktur im Hintergrund mag von bisherigen Systemen abweichen, allerdings bringt dies in der ersten Entwicklungsstufe noch keine Vorteile gegenüber herkömmlichen Assistenzsystemen.

Zweite Entwicklungsstufe: Die Vorteile der Datenstruktur werden erst in der zweiten Entwicklungsstufe deutlich, in der das Assistenzsystem im Nacharbeitsbereich einsetzbar ist. Dies sind Bereiche, in denen die Aufgaben und Aufträge stark variieren können. Das Assistenzsystem ist auf Montageaufgaben beschränkt.

Dritte Entwicklungsstufe: In dieser Entwicklungsstufe kann der Mitarbeiter bei Bedarf zusätzlich das Auftragsziel fortschrittsunabhängig ändern, sodass das Assistenzsystem ihn hinsichtlich eines neuen Ziels unterstützt und die Prozesse nachverfolgt sowie dokumentiert.

Vierte Entwicklungsstufe: In dieser Entwicklungsstufe besitzt das Assistenzsystem die Fähigkeit, den Produktzustand allein aufgrund der Beobachtungen zu ermitteln und dementsprechend Handlungsanweisungen auszusprechen.

Erweiterter Einsatz: Während des Einsatzes prüft und dokumentiert das Assistenzsystem nahezu kontinuierlich die Nacharbeitsprozesse. Dadurch können bestimmte

Informationen gewonnen werden, die später für unterschiedliche Zwecke genutzt werden können. In der vorliegenden Arbeit wird dies am Beispiel der optimierten Auftragszuweisung in der Nacharbeit gezeigt.

Die vorliegende Arbeit ist in 11 Hauptkapitel untergliedert.

Kapitel 1 gibt eine Zusammenfassung der vorliegenden Arbeit wieder.

Kapitel 2 leitet in das Thema ein, erläutert grundlegende Herausforderungen in der Produktion und greift prägnant das Thema Assistenzsysteme auf. Zudem wird das Ziel der Arbeit erläutert und der Aufbau dieser Arbeit beschrieben.

Kapitel 3 und 4 gehen auf den aktuellen Stand der Technik ein, wobei Kapitel 3 die theoretischen Werkzeuge abhandelt und noch keinen Bezug zur Produktion und Assistenzsystemen herstellt. Die vorgestellten theoretischen Grundlagen sind zum Verständnis der Methodik und des entwickelten Assistenzsystems erforderlich. Kapitel 4 nimmt Bezug zum Stand der Technik in der Produktion mit Hinblick auf Assistenzsysteme. Beide Kapitel gehen jedoch nur so weit, wie das Thema für die weitere Arbeit erforderlich ist. So werden bspw. zum Thema Graphentheorie der Graphenaufbau und die Graphtraversierung aufgegriffen. Für diese Arbeit weniger relevante Themen wie Graphenfärbung werden zugunsten der Überschaubarkeit und fehlender Notwendigkeit nicht behandelt.

In Kapitel 5 werden die Anforderungen für das zu entwickelnde Kognitive Assistenzsystem vor dem Hintergrund des Nacharbeitsbereichs definiert.

In Kapitel 6 wird die Entwicklung des Kognitiven Assistenzsystems in vier Entwicklungsstufen beschrieben. Dabei wird für jede Entwicklungsstufe die theoretische Umsetzung, aber auch die praktische Umsetzung ausgeführt.

In Kapitel 7 wird auf die Steuerung und Kommunikationsstruktur innerhalb des Kognitiven Assistenzsystems eingegangen.

In Kapitel 8 wird die optimierte Auftragszuweisung, die durch ein intelligentes Assistenzsystem ermöglicht wird, betrachtet. Hierbei wird das Thema Assistenz kurzzeitig außer Acht gelassen. Vielmehr wird die Modellierung des Lagerbereichs beschrieben.

In Kapitel 9 wird die Umsetzung und Validierung des Assistenzsystems erläutert. Hierzu wird ein Teilbereich eines Automatgetriebes als Fallbeispiel für die Nacharbeit herangezogen.

In Kapitel 10 wird die Anwendung der optimierten Zuweisung durch die ermittelten Mitarbeiterfähigkeiten beschrieben.

In Kapitel 11 werden die wesentlichen Punkte der Arbeit zusammengefasst. Zudem wird ein kurzer Ausblick über weitere Entwicklungen in diesem Bereich gegeben.

3 Grundlagen des Betrachtungsbereichs und bestehende Forschungsansätze

Dieses Kapitel legt den Fokus auf den Teil der Produktion, der mit der Montage eine hohe Ähnlichkeit aufweist. Dies sind Bereiche, in denen eine hohe Anzahl manueller und wertschöpfender Tätigkeiten anfallen. Dabei ist die Fertigung nicht Gegenstand der Betrachtung. Berücksichtigt werden allerdings die klassische Montage und funktional angrenzende Bereiche.

3.1 Montage und Montagesequenzen

Da diese Arbeit im industriellen Kontext dem Bereich der Montage zuzuordnen ist, soll zunächst der Begriff „Montage“ betrachtet werden. Im Folgenden werden daher einige Definitionen nach Normen, Verbänden oder Personen näher beleuchtet. Der Duden definiert das Substantiv „Montage“ als **„Zusammenbau einzelner vorgefertigter Teile zu einer funktionsfähigen Maschine“** [8]. Diese Definition entspricht tendenziell dem alltäglichen Gebrauch des Begriffes „Montage“.

Die International Academy for Production Engineering CIRP (College International pour la Recherche en Productique) definiert im Wörterbuch der Fertigungstechnik die Montage als **„Die Gesamtheit aller Vorgänge, die dem Zusammenbau von geometrisch bestimmten Körpern dienen“**. [9 S.1]

Lotter bzw. Warnecke definieren die Montage wie folgt: „Es ist Aufgabe der Montage, aus diesen Teilen ein Produkt höherer Komplexität mit vorgegebenen Funktionen in einer bestimmten Zeit zusammenzubauen.“ [Warnecke nach [10 S.1]]

Whitney beschreibt die Montage (Assembly) wie folgt: „Assembly is more than putting parts together. Assembly is the capstone process in manufacturing. It brings together all the upstream processes of design, engineering, manufacturing, and logistics to create an object that performs a function.“ [11 S.1]

Dabei unterscheidet Whitney zwischen Montage und Fertigung (Fabrication). [Vgl. 11 S.1]

Oft werden die Begriffe „Fertigung“ und „Montage“ synonym verwendet. Die Kanban-Fertigung wird bspw. beschrieben und anhand der Montage erläutert. [Vgl. 12 S.43] Teilweise werden die Teilefertigung und die Montage als Teil der Fertigung bezeichnet. [Vgl. 13 S.4] Allerdings werden in vielen Fällen „Fertigung“ und „Montage“ zwar als Teil der Produktion verstanden, aber strikt voneinander getrennt oder begrifflich getrennt genutzt. [14 S.30, Vgl. 15 S.4, Vgl. 16 S.242, Vgl. 17 S.448, Vgl. 18 S.25, Vgl. 19 S.6]

Müller beschreibt „die Montage [...] neben der Fertigung [als] [...] Teilsystem der Produktion innerhalb eines Unternehmens“ [20 S.702]. Durch das Durchführen von Mon-

tageprozessen wird ein Endprodukt aus Bauteilen, Baugruppen und Modulen zusammengesetzt. Die Inbetriebnahmen und somit die Funktionsprüfung liegen ebenfalls im Verantwortungsbereich der Montage. [20 S.702-704]

3.1.1 Montagesequenzen

Bei der Entwicklung eines Montagesystems eignet sich die Herangehensweise nach dem Prinzip: Produkt – Prozess – Betriebsmittel. [20 S.713] Dahinter steht der Gedanke, dass das Produkt idealerweise auf keine Merkmale und Funktionen verzichtet. Basierend auf der Produktstruktur werden Montageprozesse abgeleitet. Stehen die Montageprozesse fest, werden geeignete Betriebsmittel bestimmt. Die Frage lautet somit „Wie kann das Produkt erstellt werden?“ und nicht „Welches Produkt lässt sich mit den vorhandenen Betriebsmitteln erstellen?“. In der Realität bestehen aber bidirektionale Beziehungen, da die Berücksichtigung von Anforderungen und Fähigkeiten zu wirtschaftlichen Kompromissen führt.

Die Herangehensweise bei der Bestimmung der Prozesse und Betriebsmittel besteht im ersten Schritt aus der rein geometrischen Betrachtung des Produktes. Da ein Produkt meist eine Vielzahl von Möglichkeiten der Zusammensetzung zulässt, ergeben sich unterschiedliche Verbaureihenfolgen. Diese rein geometrischen Sequenzen werden anschließend mit Montageprozessen verknüpft, wodurch zusammen mit geeigneten Betriebsmitteln ein Plan zur Zusammensetzung des Produktes erstellt werden kann: der sogenannte Montagevorranggraph. [21 S.62] Montagesequenzen sind somit eine Abfolge von Montageprozessen (Arbeitsvorgängen), die auf einem Bauteil oder einer Baugruppe zur Montage eines Bauteils oder Baugruppe angewendet werden [22 S.10]. [Vgl. 23 S.337]

Zur Erstellung der Verbaureihenfolge existieren viele systematische Ansätze. Eine Auswahl bekannter Methoden wird in Kapitel 3.1.2 vorgestellt.

3.1.2 Verfahren zur Generierung von Verbaureihenfolgen

Bei der Generierung der Verbaureihenfolgen können bei unsystematischer Herangehensweise unmögliche Lösungen entstehen. Daher existieren viele Verfahren zur systematischen Generierung von Verbaureihenfolgen. Diese reichen von manuellen bis hin zu computergestützten Verfahren. Obwohl in dieser Arbeit der Demontage eine besondere Rolle zugesprochen wird, werden Verfahren zur Generierung von Verbaureihenfolgen vorgestellt. Diese Betrachtung steht allerdings nicht im Widerspruch zum Demontageproblem, da Montageprozesse oftmals aus der Betrachtung der Demontage eines Produktes abgeleitet werden. De Mello und Sanderson weisen darauf hin, dass das Montageproblem in ein Demontageproblem transferiert werden kann [Vgl. 24 S.11]. Auch Lambert verweist hier auf enge Verflechtungen zwischen Demontage und Montage [25 S.85]. Whitney stellt fest, dass unter bestimmten Umständen die Sequenzumkehrung einer Demontagereihenfolge zu einer zulässigen/durchführbaren Verbaureihenfolge führt. [11 S.184] Einige Verfahren zur Generierung von Verbaureihenfolgen behandeln die Demontage analytisch und kehren die Sequenz um.

Methode nach Bourjault zur Generierung möglicher Verbaureihenfolgen

Eine systematisierte Vorgehensweise zur Generierung von Verbaureihenfolgen ist die Methode nach Bourjault, die Whitney beschreibt. Dazu bedient er sich eines Basiskonstruktes, das für die systematische Generierung von Verbaureihenfolgen verwendet wird. Dieses bezeichnet er als Liaison-Graph (eng. Liaison Diagram). Abbildung 3a zeigt ein Beispielprodukt, wobei Abbildung 3b den zugehörigen Liaison-Graphen darstellt.

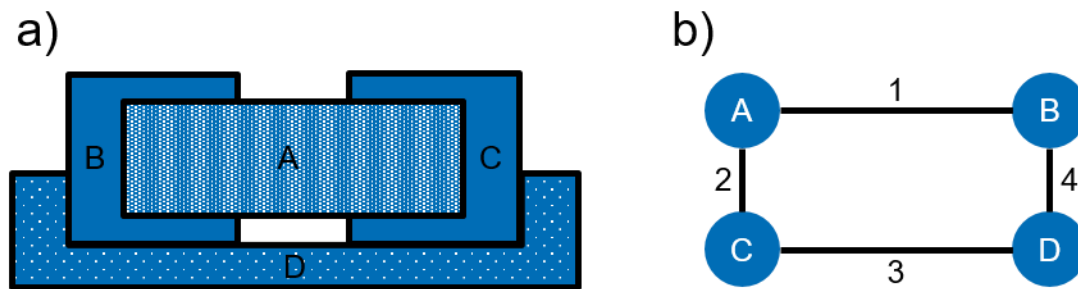


Abbildung 3: Beispielprodukt nach Bourjault (a) Darstellung eines fiktiven Produktes mit den Bauteilen A, B, C, D. b) Zu a) zugehöriger Liaison Graph (Liaison Diagram) [Eigene Darstellung nach [11 S.191]]

Whitney beschreibt den Liaison-Graphen in Form eines Diagramms, das „in den Knoten die Bauteile eines Produktes darstellt, wobei die Kanten für Verbindungen entsprechender Bauteile stehen“. [11 S. 34] Eine Verbindung ist gegeben, sobald ein weiteres Bauteil so berührt/kontaktiert wird, dass für beide Bauteile eine Bewegungseinschränkung entsteht. Whitney beschreibt die Methode nach Bourjault textuell und nicht mathematisch-algorithmisch. Sie basiert auf der Beantwortung bestimmter Fragen der Form $R(i; B)$. Hierbei steht i für die Verbindung zweier Bauteile/Baugruppen, die gerade betrachtet werden. B ist die Menge bereits getätigter Verbindungen. Eine Verbindung ist die Beziehung zweier Bauteile/Baugruppen, die sich direkt in der relativen räumlichen Freiheit (translatorische Freiheitsgrade) einschränken. In naiver Ausformulierung bedeutet $R(i; B)$: „Ist die Verbindung i möglich, wenn die Verbindungen der Menge B bereits eingegangen wurden (?)“. Ausgangspunkt der Betrachtung ist das komplettierte Produkt. Zu Beginn wird davon ausgegangen, dass alle Verbindungen bis auf die Letzte eingegangen wurden. Sind Verbindungen physikalisch nicht möglich, so wird diese Frage so lange in Subfragen zerlegt, bis eine Verbindung möglich ist. Die Subfragen werden generiert, indem aus der Menge bereits vorhandener Verbindungen B eine Verbindung gelöst wird. Hieraus ergeben sich die spätesten Verbaupunkte.

Das folgende Beispiel aus [11 S.190] dient der Verdeutlichung der Methode nach Bourjault. Als Beispielprodukt wird die Konstruktion nach Abbildung 3 gewählt. Die Montage findet in der Ebene statt. Zunächst wird Verbindung $i = 1$, also die Verbindung A-B und $B = 2,3,4$ überprüft: $R(1; 2,3,4)$. Betrachtet man das Beispielprodukt, wird klar, dass die Verbindung $R(1; 2,3,4)$ in der Ebene nicht möglich ist. Zudem ist die Prüfung

dieser Verbindung nicht notwendig, da die letzte Verbindung bestimmt ist, weil bei $N - 1$ Verbindungen die $n - te$ Verbindung gleich mitbestimmt ist. Diese Regel wird als loop closure rule bezeichnet [11 S.185]. Daher wird diese Frage in drei Subfragen untergliedert: $R(1; 3,4)$, $R(1; 2,4)$ und $R(1; 2,3)$. Da diese ebenfalls nicht möglich sind, werden die Fragen in weitere Subfragen $R(1; 3)$, $R(1; 4)$, $R(1; 2)$ unterteilt. Alle drei Subfragen können positiv beantwortet werden, da bspw. die Verbindung 1 möglich ist, wenn ausschließlich die Verbindungen 2 oder ausschließlich 3 oder ausschließlich 4 bereits bestehen. Dies führt zur Schlussfolgerung, dass die Verbindung 2 vor 3 und 4 eingegangen werden muss. Hieraus ergeben sich die folgenden Beziehungen für die Verbaureihenfolge: $1 \geq 3, 4$ und $1 \geq 2, 4$ (Verbindung 1 muss eingegangen werden, bevor 2 und 4 eingegangen wurden). Ausgehend von den Fragen $R(2; 1,3,4)$, $R(3; 1,2,4)$, $R(4; 1,2,3)$ kann der oben beschriebene Vorgang wiederholt werden, um somit weitere Verbaureihenfolgen und Bedingungen zu berücksichtigen.

Cut-Set-Methode

Eine weitere Methode zur Generierung von Verbaureihenfolgen ist die Cut-Set-Methode. In der Graphentheorie bezeichnet ein Schnitt (eng. Cut) eine Operation, die eine Menge von Knoten eines Graphen in zwei Mengen von Knoten unterteilt [Vgl. 26 S.3]. Nach der Operation ist dieser Graph nicht mehr zusammenhängend. Die Cut-Set-Methode nach de Mello untersucht ein Produkt und seine Bauteile auf ihre Verbindungsbeziehungen und bildet diese grafisch ab. Dabei stehen die Knoten für ein Bauteil und die Kanten für Verbindungen.[24]

Die Cut-Set-Methode wird in zwei Stufen durchgeführt:

1. Erstellung eines Beziehungsmodells (*Relational Model*)
2. Betrachtung der (De-) Montierbarkeit mittels Cut-Sets

Die Methode beginnt mit der Produktanalyse und der expliziten Betrachtung im zusammengebauten Zustand. Ergebnis ist ein Produktmodell, das mittels eines 5 Tupels $[P, C, A, R, a\text{-functions}]$ beschreiben wird. Wobei P die Menge der Bauteile darstellt, C für die Menge der Kontakte (planar, (Gewinde) Schaft und (Gewinde) Bohrung) steht und A für die Menge der Verbindungen (Schraubverbindung, Klebverbindung etc.) [24 S.6]. R repräsentiert die paarweise Verbindung zwischen P, C oder A und beinhaltet die Typen part-contact, target-attachement, agent-attachement, blocking-part-attachement. Der part-contact-Typ beschreibt zwei Kontaktierungen. Target-attachements sind gesicherte Verbindungen [27 S.2]. Kljajin führt hierfür ein Beispiel in Form einer Verbindung zwischen einer Mutter und einer Unterlegscheibe bzw. der Verbindung zwischen Unterlegscheibe und einer Schraube auf. Agent-attachements sind Verbindungen, die eine Verbindung verursachen [Vgl. 27 S.2]. Im zuvor genannten Beispiel wäre zwischen Mutter und Schraube ein agent-attachement vorhanden. Abbildung 4 stellt ein Relationales Modell in Graphenform dar. Die Rechtecke stehen für ein Element aus P. Die Kreise (C) stehen für eine Kontaktierung. Die Kreise (A) geben die Form der Verbindung wieder. Die Linien (R) stehen für die Verbindungen.

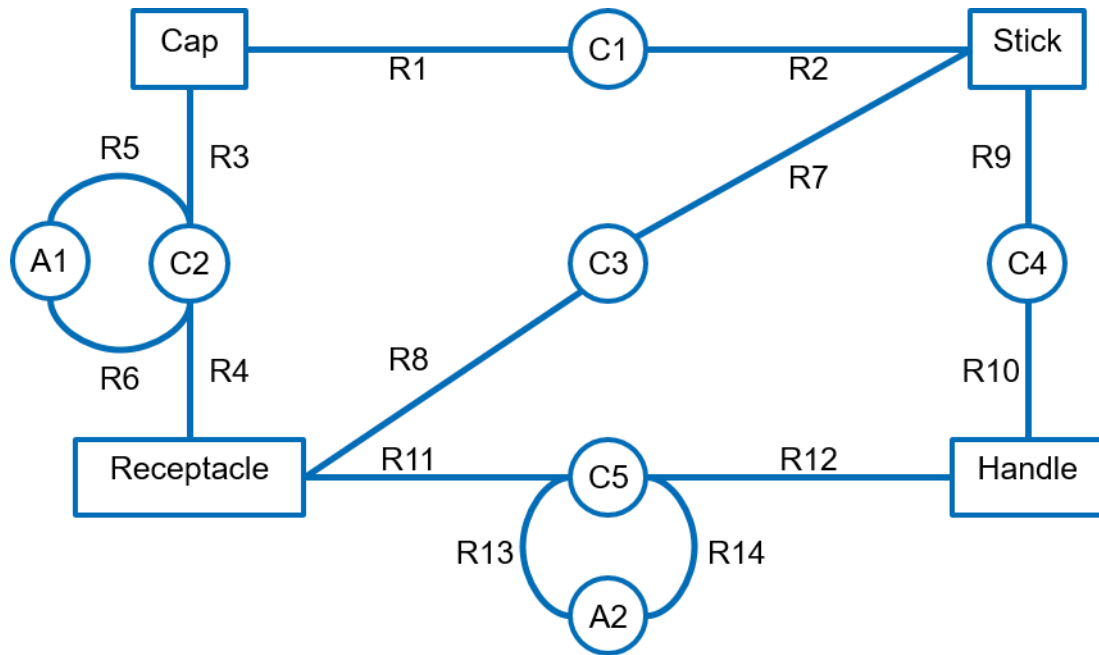


Abbildung 4: Relationales Modell als Graph. ¶[Eigene Darstellung nach [24 S.9]]

Abbildung 5 präsentiert einen vereinfachten Graphen, der lediglich die Kontaktierungen enthält. Dieser ist übersichtlicher und ermöglicht dem Planer die Betrachtung der einzelnen Cut-Sets.

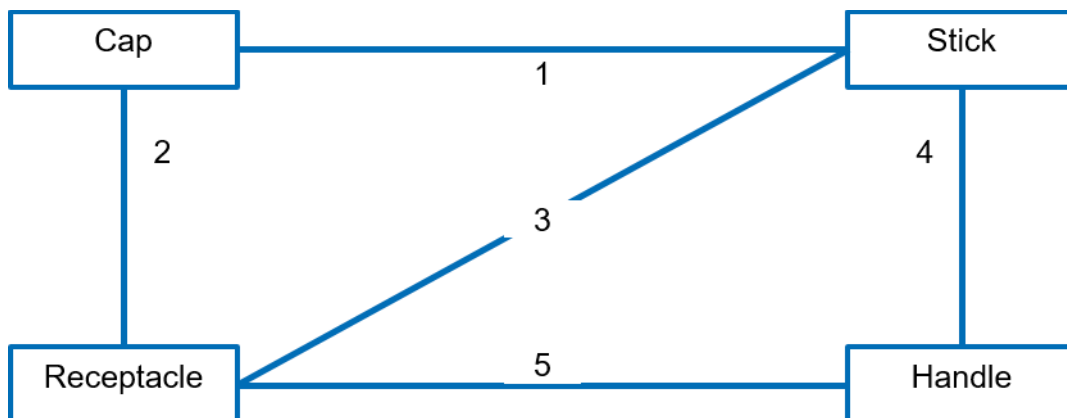


Abbildung 5: Verbindungsgraph ¶[Eigene Darstellung nach [24 S.10]]

Abbildung 6 zeigt die Cuts (Schnitte) auf den Verbindungen von Abbildung 5. Bei der Betrachtung der Cut-Sets, die die Zerlegung des Produktes darstellen, muss auch die Machbarkeit der Zerlegung berücksichtigt werden. Hierzu werden drei Eigenschaften betrachtet [24 S.12]:

1. Geometric Feasibility (Geometrische Durchführbarkeit)
2. Mechanical Feasibility (Mechanische Durchführbarkeit)
3. Stability (Stabilität)

Die **Geometrische Durchführbarkeit** wird bestimmt durch kollisionsfreie Bahnen zwischen zwei zu kontaktierenden Bauteilen. [Vgl. 24 S.12]

Die **Mechanische Durchführbarkeit** berücksichtigt die technische Verbindung zweier Bauteile. Bspw. ob zwei Bauteile, die durch eine Schraubverbindung verbunden sind, auch tatsächlich in Verbindung gebracht werden können, wenn bspw. die Schraubbohrungen durch ein anderes Bauteil überdeckt werden. [Vgl. 24 S.12]

Die **Stabilität** ist die Eigenschaft einer Verbindung, nicht spontan auseinanderzubrechen, wenn eine andere Verbindung gelöst wird. Zudem müssen weitere Bauteile ebenfalls ihre relative Position beibehalten. Grund hierfür kann die Gravitationskraft sein. So sind Kugeln in einem Behälter sicher miteinander in Kontakt. Sobald jedoch der Behälter entfernt wird, gehen die Verbindungen (spontan) verloren. [Vgl. 24 S.12]

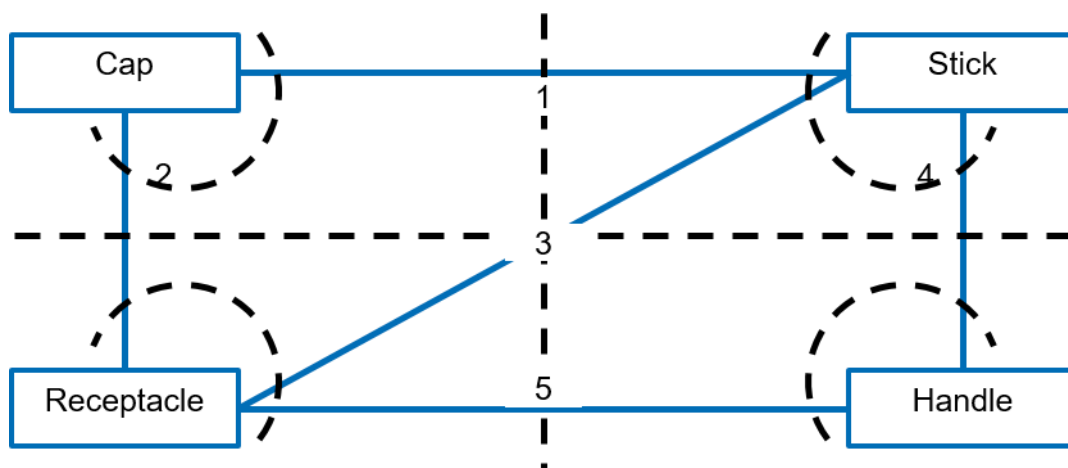


Abbildung 6: Machbare Schnittpunkte zur Generierung der Cut-Sets [Eigene Darstellung nach [24 S.18]]

Zu Ermittlung der Verbaureihenfolgen (assembly sequences) werden basierend auf dem Relationalen Modell die Durchführbarkeit der Verbindungen mittels Algorithmen untersucht. Dabei wird das Produkt ideell zerlegt und die Durchführbarkeit der Zerlegung überprüft. [Vgl. 24 S.17] Zum Beispiel wird zur Ermittlung der geometrischen Durchführbarkeit eines Montageprozesses ein Path-Planning-Algorithmus durchgeführt, um (falls vorhanden) kollisionsfreie Montagebahnen zu erkennen. [Vgl. 24 S.13] Die mechanische Durchführbarkeit wird durch Logiken geprüft, die das Relationale Modell als Grundlage nehmen und die Verbindungen bewerten. [Vgl. 24 S.15] Auch die Stabilität der Verbindung wird mit Sonderlogiken u. a. durch Orientierung des Produktes, Einwirkung der Schwerkraft o. ä. überprüft. [Vgl. 24 S.15-16] Das Ergebnis dieser Herangehensweise wird im AND/OR-Graphen repräsentiert, der im nächsten Kapitel vorgestellt wird. [24 S.17]

AND/OR-Graph

Eine weitere Repräsentationsart der Produktzusammensetzung ist neben der Verbaureihenfolge der AND/OR-Graph. Die Grundidee ist, dass stets alle Bauteile/Baugruppen eines Produktes betrachtet werden. Allerdings müssen diese Bauteile nicht immer Teil des wachsenden oder abnehmenden Produktes sein. Mathematisch bedeutet dies, dass die Bauteile bzw. Baugruppen als Elemente unterschiedlichen Mengen zugehören. Dabei repräsentieren die Knoten des Graphen eine Teilmenge der Bauteile

des Produktes. Diese Teilmenge bildet eine durch De-/Montageprozesse erzielbare Menge und kann als Produktzustand interpretiert werden. Dabei werden zu jederzeit alle Bauteile betrachtet. Der Graph kann als Hypergraph verstanden werden, da hier eine Kante mehrere Zustände erwirkt [Vgl. 28 S.139]. Ein Hypergraph besteht aus Kanten, die auch mehr als zwei Knoten verbinden dürfen. [Vgl. 29 S.366] Wird bspw. ein Produkt durch einen Prozess entzweit, entstehen zwei Teilmengen durch einen Prozess (Kante). [vgl. 28 S.139-141]

GAPP: A Generative Assembly Process Planner

EIMaraghy stellt eine Methode zur automatischen Generierung von Montagereihenfolgen für Produkte mit ausschließlich steifen Bauteilen vor: GAPP – Generative Assembly Process Planner [30]. Diese Methode stellt eine computergestützte Alternative zur Erstellung von Montagesequenzen dar. Aus einer die Randbedingungen enthaltenden Datei eines Produktes wird dieses auf drei Arten von Verbindungsbeziehungen (mating relations) untersucht. Der Verbindungstyp bezieht sich hierbei nicht auf eine Montageoperation, sondern auf geometrische Beziehungen von Körpern. Dabei werden drei Arten unterschieden: (einander) kontaktierend, blockierend und frei. Zur Detektion dieser Beziehungen werden unterschiedliche Bedingungen überprüft: Zunächst wird die Ausrichtung zweier Bauteile zueinander ermittelt. [30]

$$n_1 * n_2 < 0 \quad (1)$$

Dabei ist n_1 der normierte Normalenvektor der Oberfläche von Bauteil 1 und n_2 der normierte Normalenvektor der Oberfläche von Bauteil 2. Zusätzlich wird die Distanz zwischen den Bauteilen überprüft. Als Drittes wird die Dimension des Bauteils samt der Lage überprüft/kontrolliert.

Zudem werden Freiheitsgradmatrizen aufgestellt, die die translatorischen und rotatorischen Freiheitsgrade eines Bauteils unter Beachtung seiner zuvor analysierten Beziehung zu einem weiteren Bauteil beschreiben. Dabei werden sog. Halb-Freiheitsgrade (half degrees of freedom) genutzt. Diese geben die Richtung entlang einer Achse ($T_{x+}, T_{x-}, T_{y+}, T_{y-}, T_{z+}, T_{z-}$) oder eine Rotation um eine Achse ($R_{x+}, R_{x-}, R_{y+}, R_{y-}, R_{z+}, R_{z-}$) an. Die Freiheitsgradmatrizen sind zentraler Bestandteil der Methode.

Zur Erstellung der Verbaureihenfolge bedient sich GAPP der Cut-Set-Methode von de Mello und Sanderson [30 S.286]. Nach der Erstellung des AND/OR-Graphen wird dieser auf durchführbare Sequenzen überprüft. Hierzu müssen die folgenden drei Bedingungen erfüllt sein: Geometrische Durchführbarkeit (Geometric interference constraints), Stabilität der Verbindungen (Stability constraints) und Zugänglichkeit (Accessibility constraints) [30 S. 287]. Die Bedingungen werden durch Anwendung unterschiedlicher Algorithmen überprüft. Zur Überprüfung der geometrischen Durchführbarkeit wird bspw. eine UND-Verknüpfung auf der Matrix der Halbfreiheitsgrade der betrachteten Bauteile durchgeführt. Die Betrachtung realistischer bzw. praktischer Verbaubar-

keiten verkleinert den AND/OR-Graphen. Zur Ermittlung der idealen Verbaureihenfolge werden weitere Präferenzen hinzugezogen wie: Neuorientierung eines Bauteils, Möglichkeiten der Parallelisierung der Prozesse, Stabilität der Bauteile und Clusterung ähnlicher Prozesse [30 S.289].

Diskussion

Die vorgestellten Verfahren erleichtern die Arbeit des Montageplaners, da ihm eine Vorgehensweise bei der Produktanalyse und Durchführbarkeit von Verbindungen, die letztendlich in Montageprozessen resultieren, an die Hand gegeben wird. Es ist jedoch zu beachten, dass gerade die klassischen Methoden auf Fragetechniken beruhen. Diese Fragen muss ein Planer beantworten, indem er das Produkt in Augenschein nimmt. [vgl. 11 S.191] Modernere Ansätze, die ebenfalls in der klassischen Literatur beschrieben sind, nehmen Computer zu Hilfe. [vgl. 30] Es können bspw. auch CAD-Programme Antworten auf die Frage geben, ob Beschränkungen im Freiheitsgrad bestehen oder nicht. Jedoch muss auch hier beachtet werden, dass die Berechnung zeit- und ressourcenintensiv ausfallen kann. Zudem sind diese Methoden nur eingeschränkt anwendbar. Nicht alle Produktarten, Maschinen, Fluide und Elektroplatinen sind gleichermaßen zu betrachten, sodass hier fallabhängig bestimmte Methoden anderen vorzuziehen sind.

3.2 Design for Assembly und Design for Disassembly

In den vorangegangenen Kapiteln wurden Möglichkeiten zur Wiederverwertung von Produkten und einzelner Komponenten beschrieben sowie die Möglichkeit, Produkte zu demontieren (oder zu montieren). Damit bei der Montage bzw. der Demontage Probleme vermieden werden, kann bereits in der Produktgestaltung auf montage- bzw. demontageförderliche Prinzipien zurückgegriffen werden.

Das klassische Verständnis von Design for Assembly (DFA) hat eine eingeschränkte Sicht, wenn es um die Wiederverwertung von Produkten geht. Eine Produktauslegung, die die Effizienz in der Montage steigert, kann unter Umständen erhöhten Aufwand in der Demontage bedeuten. Ein Beispiel hierfür ist die Bewertung von Schraubprozessen im DFA. [vgl. 31 S.427-428] Daher ist für die Nacharbeit auch die Betrachtung der Methoden aus dem Bereich des Design for Disassembly (DFD) sinnvoll. Ein weiterer Begriff, der im Zusammenhang mit Demontage auftaucht, ist „Active Disassembly“ [32, 33]. Hierbei werden intelligente Materialien wie Formgedächtnislegierungen in das Produkt eingebaut und als Verbindungselement genutzt. Bestimmte Voraussetzungen, bspw. die Bestromung des Materials etc. führen zu veränderten Materialeigenschaften. Dies äußert sich durch Verformung des Materials, was bei entsprechend ausgelegter Konstruktion zur (aktiven) Demontage des Produktes führt. [32, 33]

Bei der Demontage von Produkten existieren unterschiedliche Ansätze, die unter verschiedenen Schlagbegriffen zusammengefasst werden, namentlich **Design for Disassembly**, **Design for Repair**, **Design for Maintenance** [34]. Obwohl alle Schlagworte

in ihren Methoden eine große Überschneidung aufweisen, unterscheidet sich das Ziel der Methodensammlung. Design for Disassembly rückt prinzipiell das Auseinandernehmen des Produktes in den Vordergrund. Das Design for Disassembly weist die größte Überschneidung zu den anderen Begriffen auf. Das Design for Repair nimmt in Betracht, dass das Produkt nicht allein zerlegt werden soll, sondern auch wieder zusammengesetzt werden muss. Daher ist es ratsam, bekannte Fehlerteile, falls möglich, an ähnlichen Orten zu platzieren und zugänglich zu machen. Das Design for Maintenance zieht darüber hinaus ebenfalls die Überwachung des Produktes in Betracht und schlägt entsprechende Lösungen zur Fehlerbehebung vor. Das Design for Disassembly ist eine nicht-einheitliche Ansammlung diverser bewährter Praktiken zur Zerlegung von Produkten. Die Ziele können unterschiedlicher Art sein. Ein Ziel liegt in der effizienten Demontage von Produkten. Andere Ziele haben den ökologischen Vorteil durch einen höheren Recyclinganteil im Fokus. Obwohl wenig bis keine Standards für DfD verfügbar sind, überschneiden sich die Prinzipien von unterschiedlichen Autoren. Vergleichbar zur Design for Assembly Methodik nach Boothroyd gilt auch hier: je weniger Bauteile, desto effizienter die Demontage.

1. **Geringe Anzahl verbauter Bauteile:** Je weniger Komponenten Bestandteil des Produktes sind, desto weniger Bauteile müssen demontiert werden. [35]
2. **Geringe Anzahl von Verbindungselementen** [35]
3. **Verwendung gleicher Verbindungselemente** [36]
4. **Vorzug von Schraubverbindungen:** Schrauben sind im Vergleich zu anderen Verbindungsmethoden leichter zu lösen [35]
5. **Verzicht auf permanente Verbindung(smittel)** (Kleber) [Vgl. 36]
6. **Bevorzugte Verwendung von Standardwerkzeugen** [34 S.79]
7. **Bessere Zugänglichkeit zu Bauteilen mit höherer Fehlerrate (Bspw. Batterie)** [34 S.37]
8. **Modulare Konstruktion** [34 S.31]

Weitere bewährte Praktiken zur Demontage beinhalten:

1. Vermeidung giftiger Stoffe im Produkt [34 S.80]
2. Bereitstellung von Anweisungen [Vgl. 34 S.89]

3.3 Einordnung und Notwendigkeit der Nacharbeit

In diesem Kapitel wird der Begriff „Nacharbeit“ näher betrachtet, da dieser vielfältig und zuweilen undifferenziert genutzt wird. In diesem Zuge wird die Nacharbeit in den Produktionsbereich insbesondere vor dem Hintergrund der Ressourcenwiederverwertung eingeordnet. Zudem wird die Notwendigkeit der Nacharbeit erläutert. Dies ist erforderlich, da die Nacharbeit aus Kundensicht keine wertschöpfenden Prozesse umfasst [37 S.30]. Dennoch ist diese häufig integraler oder nachgelagerter Bestandteil des Montagebereichs.

3.3.1 Abgrenzung und Einordnung der Nacharbeit

Der Begriff *Nacharbeit* wird unterschiedlich genutzt. Daher ist an dieser Stelle eine kurze Abgrenzung erforderlich. Die Verwendung des Begriffs meint sowohl die Tätigkeit, die die Behebung der Produktfehler beschreibt als auch den Unternehmensbereich, wo die entsprechenden Prozesse durchgeführt werden. Zur Vermeidung von Irritationen kann im zweiten Fall vom *Nacharbeitsbereich* gesprochen werden. Im Nacharbeitsbereich werden nicht allein Prozesse durchgeführt, die zur Fehlerbehebung führen, sondern auch weitere Prozesse, wie bspw. die Dokumentation der Prozesse.

Eine weitere Abgrenzung des Begriffs „Nacharbeit“ ist zum Begriff „Reparatur“ erforderlich. Beide beschreiben die Beseitigung eines Mangels an einem Produkt und werden daher oft synonym genutzt. DIN EN ISO 9000:2015 beschreibt beide Begriffe und definiert die Nacharbeit wie folgt:

„Maßnahme an einem nichtkonformen Produkt oder einer nichtkonformen Dienstleistung, damit es/sie die Anforderungen erfüllt“ [3.12.8 38]

Die Reparatur hingegen ist wie folgt definiert:

„Maßnahme an einem nichtkonformen Produkt oder einer nichtkonformen Dienstleistung, um diese(s) für den beabsichtigten Gebrauch annehmbar zu machen“ [3.12.9 38]

In der zitierten DIN wird der englische Begriff „Rework“ mit „Nacharbeit“ übersetzt und „Repair“ mit „Reparatur“.

Die IPC 7711/7721 Revision B, die insbesondere für die Elektroindustrie gilt, unterscheidet zwischen Repair (dt.: Reparatur) und Rework (dt.: Nacharbeit). Dabei bezeichnet Repair die einfache Wiederherstellung der funktionalen Anforderungen. Rework umfasst alle Maßnahmen, die dazu dienen, ein ungenügendes Produkt unter Anwendung originaler oder äquivalenter Prozesse in einen Zustand zu versetzen, sodass es den Spezifikationen entspricht.

Müller ordnet den Prozess *Nacharbeiten* den *Sonderoperationen* zu und erklärt diese für notwendig, „falls Montageprozesse nicht fehlerfrei durchgeführt wurden und zur Qualitätssicherung weitere Maßnahmen notwendig sind“ [20 S.710]. Auch müsse man sich bei Nacharbeitsprozessen auf ein „Minimum [...] beschränken“. [20 S.710]

Nach der Definition von Kropnik fasst die Nacharbeit alle Prozesse zusammen, die Mängel an Produkten beheben, wenn diese durch Fehler in der Fertigung [Montage] entstanden sind. [39 S.317]

Tackenberg et al. definieren die Nacharbeit als eine erforderliche Handlung als Konsequenz eines nicht-ausreichenden Arbeitsergebnisses. [40 S.50]

Zusammenfassend lässt sich sagen, dass die Nacharbeit, sobald man diese mit dem englischen Begriff „Rework“ gleichsetzt, solche Prozesse beschreibt, die notwendig sind, um ein Produkt, das den geforderten Spezifikationen nicht entspricht, in eben

diesen spezifikationsgerechten Zustand zu versetzen. Dahingegen beschreibt die Reparatur „Repair“ Prozesse, die vollzogen werden, damit ein nicht anforderungsgerechtes Produkt die Funktionalitäten erfüllt, die (von z. B. einem Kunden) akzeptiert werden. Allerdings fußen diese Funktionen nicht auf der Einhaltung der Spezifikationen, sondern auf alternative Prozesse oder Produktumsetzungen.

Wie die Abgrenzung zwischen den Begriffen „Nacharbeit“ und „Reparatur“ zeigt, ist die Nacharbeit nicht die einzige Maßnahme, bei der ein Produkt zerlegt werden muss. Zur Einordnung des Nacharbeitsbereichs muss dieser einer globalen Betrachtung im Sinne des Produktlebenszyklus unterzogen werden. Dazu wird auf englische Bezeichnungen zurückgegriffen, da diese aufgrund der Anzahl der Arbeiten, die sich mit der Ressourcenwiederverwertung befassen, die Definition der Begriffe unterstützen. Es muss beachtet werden, dass die Begriffe sowohl im Deutschen als auch im Englischen nicht immer eindeutig gebraucht werden. Zudem kann die Verwendung der Begriffe branchenspezifisch ausfallen. Abbildung 7 zeigt die Verortung des „Remanufacturing“ im Wertschöpfungsprozess.

Remanufacturing (dt.: Refabrikation) bezeichnet den Prozess zur Wiederherstellung des neuwertigen Zustands eines gebrauchten Produktes. Für den Kunden ist de facto nicht ersichtlich, ob das Produkt neu oder gebraucht ist [41]. Daher erfordert die Refabrikation die komplette Demontage des Produktes, dessen qualitative Beurteilung und die Beurteilung seiner Bauteile/Baugruppen sowie die Behebung der Fehler durch die Auswechslung defekter Bauteile/Baugruppen und die Wiederaussetzung. Auf refabrizierte Produkte kann der Hersteller seine Garantie geben. Laut britischem Standard BS 8887-2:2009 ist bei der Refabrikation die Ausstellung der Herstellergarantie gefordert [Vgl. BS 8887-2:2009 zitiert nach 42 S.14]. [43 S.2]

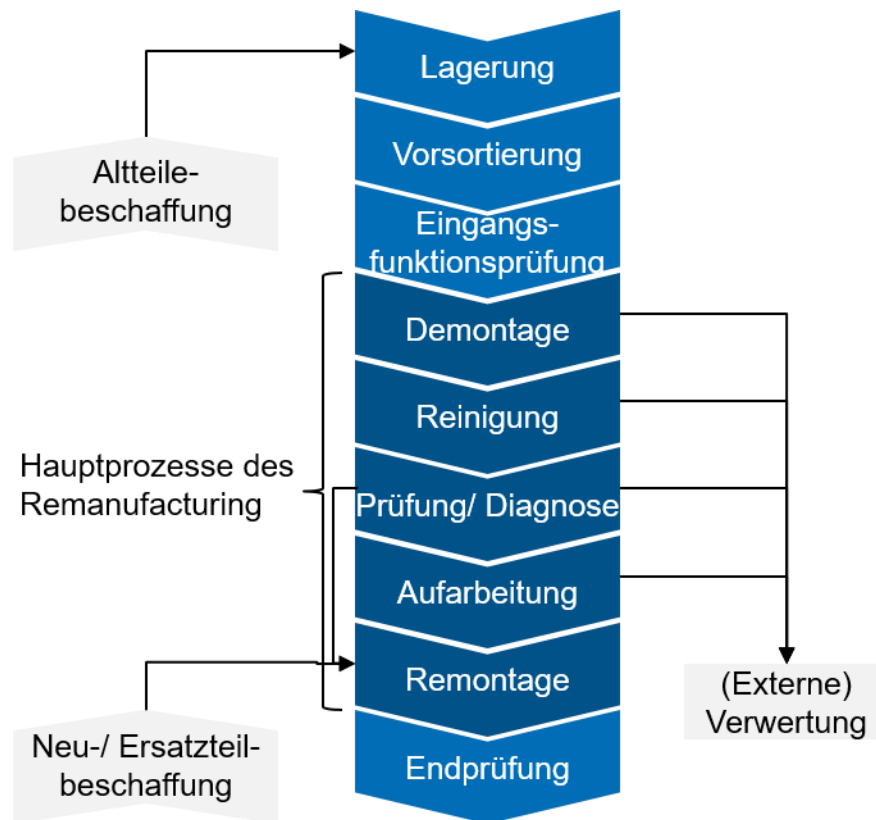


Abbildung 7: Einordnung des Remanufacturing im betrieblichen Wertschöpfungsprozess [42 S. 18]

Refurbishing (dt.: aufpolieren [44 S.9] nach [45]) bezeichnet das Herstellen eines bestimmten Qualitätsniveaus bei gebrauchten Produkten. Dieses ist jedoch von niedrigerem Stand als das ursprüngliche Qualitätsniveau. [46 S.119] Refurbishing ist die qualitative Aufwertung des Produktes im Vergleich zum vorherigen Zustand, aber nicht unbedingt im Vergleich zum ursprünglichen Zustand. [47] nach [48 S.585] Die Produktqualität kann auch höher ausfallen als im Originalzustand, allerdings geht damit nicht zwangsläufig eine Herstellergarantie einher. Tendenzielles Ziel von Refurbishing ist die ästhetische Aufwertung des Produktes. [49 S.6]

Der Begriff „Reconditioning“ ist sehr nah am Begriff „Refurbishing“. Die Vereinten Nationen bzw. die Europäische Kommission definiert Refurbishment nach wie folgt:

„Modification of an object that is a waste or a product to increase or restore its performance and/or functionality or to meet applicable technical standards or regulatory requirements, with the result of making the waste or product a fully functional product to be used for a purpose that is at least the one that was originally intended.“ [50 S. 37] aus [51 S.8] UNEP/CHW/OEWG.10/INF/10 S.8.

Mit dieser Definition werden zwei unterschiedliche Ziele der Wiederverwertung erfasst. Zum einen entspricht das Produkt nach dem Refurbishment in der Funktion dem ursprünglichen Zustand. Zum anderen kann das Produkt nach der Behandlung den ursprünglichen Zustand auch übertreffen.

Auch **Reconditioning** bezeichnet die Verbesserung des Produktzustands. Allerdings wird auch hierbei nicht die Neuwertigkeit des Produktes zum Ziel gesetzt. [49 S.6] Daher kann der Hersteller auf dieses Produkt keine Garantie mehr geben.

Cannibalization bezieht sich auf die Wiederverwendung ausgewählter Bauteile aus einem Produkt. [47 nach, 48 S.585] Ziel ist nicht die Wiederherstellung des verkaufsfähigen Zustands, sondern das Wiederverwerten einzelner Komponenten, möglicherweise in einem anderen Produkt. Bauteile oder Baugruppen werden in bedeutend und unbedeutend eingeteilt [Vgl. 52 S.1769]. Daher wird der Rest des ausgenommenen Produktes entsorgt. [43 S.2-3]

Abbildung 8 zeigt den Zusammenhang unterschiedlicher Recyclingstufen untereinander.

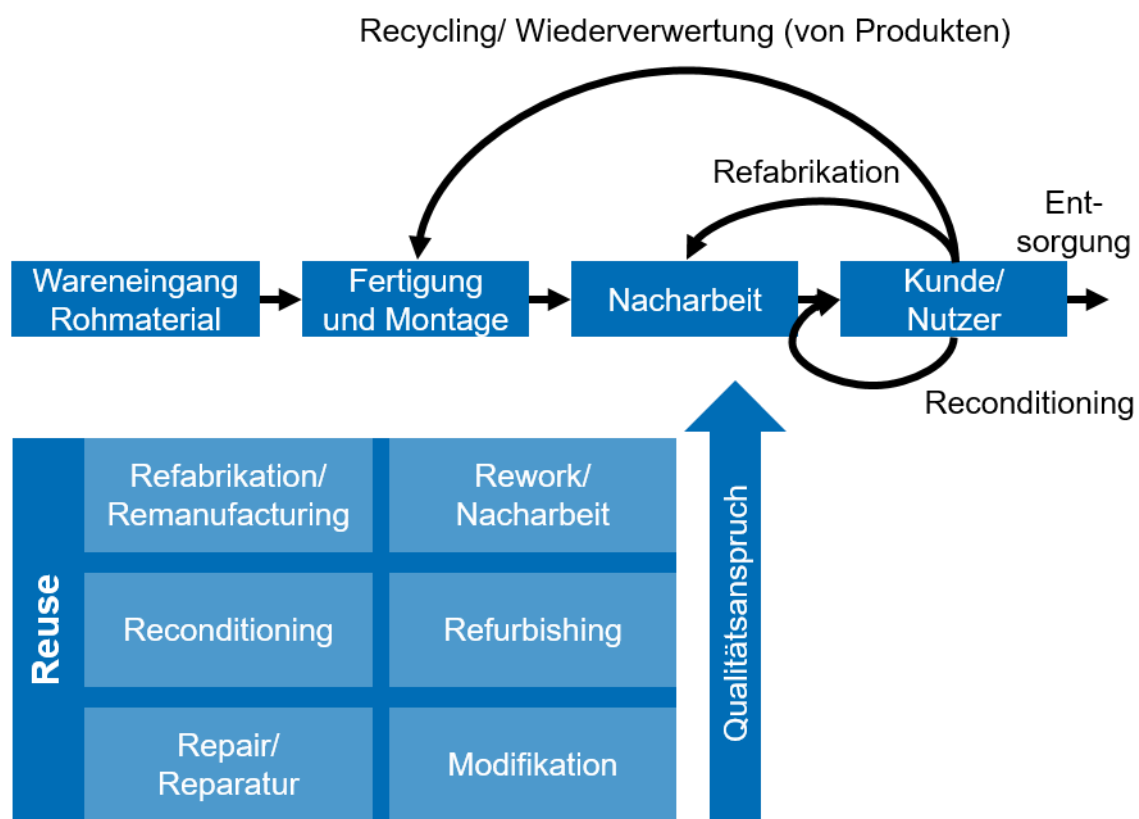


Abbildung 8: Ausprägung der Produktwiederverwertung und deren Einordnung in unterschiedliche Produktentstehungsphasen [Eigene Darstellung nach [53 S.7]]

3.3.2 Notwendigkeit für die Nacharbeit

Die Notwendigkeit zur Nacharbeit aus innerbetrieblicher Sicht wird mit der nachträglichen Fehler- und Mängelbeseitigung eines fehlerhaften Produktes begründet. Ein weiterer Bedarf für die Nacharbeit ergibt sich aus der praktischen Erfahrung fehlerbehafteter Prozesse, die unter Umständen zu mangelhaften Produkten führen können. Ist das Produkt somit nicht mehr spezifikationsgetreu, bleibt dem Produzenten lediglich die Möglichkeit, das Produkt zu verschrotten, den Fehler zu beheben oder dieses unter

Umständen nach einer Qualitätsabstufung zu einem günstigeren Preis anzubieten. [39 S.317, 54S. 36-37, 55 S.235] Allerdings entfällt die letzte Option bei sicherheitskritischen Fehlern sowie bei einem hohen Qualitätsanspruch. In diesem Fall müssen wirtschaftliche Überlegungen angestellt werden, die entweder die Nacharbeit oder die Verschrottung begünstigen. Die Gesamtkosten für die Nacharbeit lassen sich durch die Summe folgender Kosten ermitteln [54 S.37]:

1. Kosten für die Nacharbeit
2. Strafkosten für eventuell nicht eingehaltene Liefertermine
3. Strafkosten für verminderte Ausbringung
4. Kosten für Qualitätskontrolle, Aussortierung und zusätzliche
5. Logistikaufwendungen

Dem gegenüber stehen die Kosten für die Verschrottung, die sich wie folgt zusammensetzen [54 S.36]:

1. Kosten für die erneute Produktion (des Ausschussteils)
2. Strafkosten für eventuell nicht eingehaltene Liefertermine
3. Strafkosten für verminderte Ausbringung (relevant für Engpass)
4. Kosten für Qualitätskontrolle, Aussortierung und weitere Logistikaufwendungen
5. Kosten für die Entsorgung

Trotz fehlender Wertschöpfung kann die Nacharbeit von wirtschaftlichem Interesse sein, wenn durch entsprechende Maßnahmen der Schaden abgemildert werden kann. Neben den wirtschaftlichen Aspekten der Nacharbeit liegt ein weiterer Vorteil in der Rückverfolgung von Produktionsfehlern. Die Untersuchung der Fehler vor oder während der Nacharbeit kann Einblicke in den Produktionsprozess verschaffen und helfen, Fehlerursachen zu identifizieren. Damit können zukünftig Fehler reduziert werden.

3.3.3 Diskussion

Die aus der Literatur hervorgehenden Vorschläge zur Nutzung der Begriffe „Nacharbeit“ und „Reparatur“ können zuweilen kontraintuitiv wirken, da im alltäglichen Sprachgebrauch auch von Reparatur die Rede ist, wenn ein Privatkunde sein Produkt in eine zertifizierte Manufaktur zur Behebung des Fehlers einschickt. Gibt der Betrieb darauf eine Garantie wie der Hersteller, kann hier von Nacharbeit gesprochen werden. Allerdings wird in der alltäglichen Erfahrung von Reparatur gesprochen. Ein wesentlicher Unterschied zur Nacharbeit kann darin bestehen, dass Nacharbeit meist unmittelbar der Endmontage oder Funktionsprüfung nachgelagert ist. Das bedeutet, dass das Produkt zumeist im Verantwortungsbereich des Herstellers liegt. Im Unterschied dazu geht, wenn im alltäglichen Sprachgebrauch von Reparatur die Rede ist, das Produkt in den Verantwortungsbereich einer Herstellerexternenpartei bspw. des Kunden über, bevor es wieder für die Reparatur vom Hersteller oder einem zertifizierten Betrieb temporär verantwortet wird. Dabei bleibt es aber Eigentum des Kunden.

3.4 Digitale Assistenzsysteme

Zur Handhabung der Produktkomplexität und der Variantenvielfalt in der Montage sowie zur Sicherstellung der Produktqualität und -funktionalität bieten Digitale Assistenzsysteme eine Lösung. Aufgabe Digitaler Assistenzsysteme ist die Unterstützung der Mitarbeiter während der Durchführung ihrer Tätigkeiten. Die Unterstützung kann auf unterschiedliche Art und Weisen erfolgen. (Kognitive) Digitale Assistenzsysteme, insbesondere im Bereich der Montage, sind oftmals als sensorbehaftete Arbeitsstationen mit Anzeigevorrichtung wie bspw. einem Bildschirm ausgelegt, wie diese Beispiele zeigen [56], [57], [58], [59].

Im Falle von kognitiver Assistenz liegt der Fokus auf der Entscheidungsunterstützung. Zwei Aspekte stehen hierbei im Vordergrund: Zum einen dürfen keine Fehlentscheidungen getroffen werden. Eine Fehlentscheidung im Sinne der Montage kann bspw. darin liegen, ein nicht vorgesehene Bauteil zu montieren. Zum anderen kann die Entscheidungsfindung beschleunigt und die darauffolgende Handlung schneller vollzogen werden. Das Büro für Technikfolgenabschätzung beim Deutschen Bundestag beschreibt Kognitive Assistenzsysteme wie folgt:

„Kognitive Assistenzsysteme dienen der Unterstützung von Beschäftigten, indem sie diesen, abhängig von den auszuführenden Tätigkeiten und den individuellen Fähigkeiten, passgenaue Informationen zur Verfügung stellen, ihnen Hinweise auf korrekte Arbeitsabläufe geben und die Überprüfung der ausgeführten Arbeitsschritte ermöglichen.“ [60]

Dieser Auszug liefert eine allgemein verständliche Beschreibung von Kognitiven Assistenzsystemen, jedoch sind für eine systematisierte Beschreibung von Assistenzsystemen im Produktionsumfeld weitere Betrachtungen notwendig. In diesem Kapitel werden einige Systematisierungsversuche für Assistenzsysteme insbesondere vor dem Hintergrund der Montage (Demontage) vorgestellt.

3.4.1 Domainenübergreifende Betrachtung von Assistenzsystemen (Unterstützungssystemen) nach Weidner

Weidner betrachtet Assistenzsysteme grundlegend und bereichsübergreifend. Daher ordnet er diese den Unterstützungssystemen zu. [61 S.14, 77] Abbildung 9 stellt die Systematik der Unterstützung nach Weidner dar.

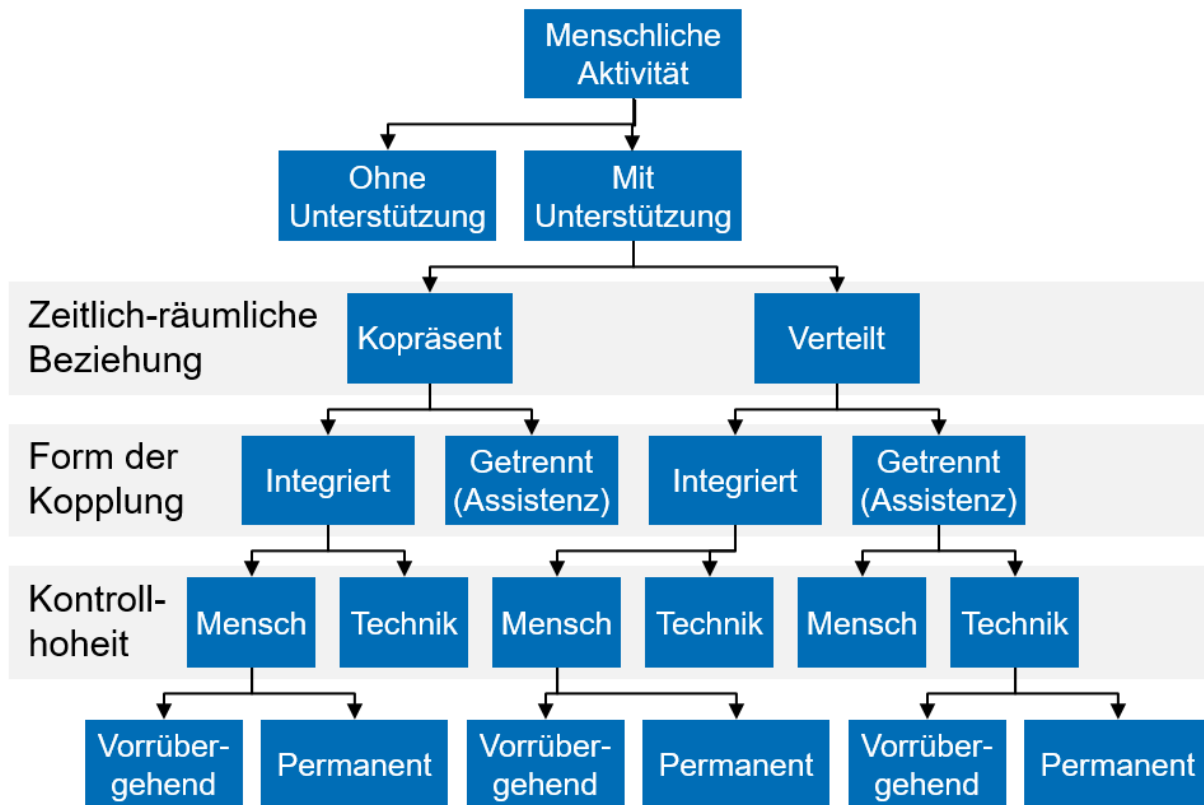


Abbildung 9: Ganzheitliche Systematik von Unterstützungssystemen [Eigene Darstellung nach Weidner [61 S.77]]

Der Inhalt des folgenden Abschnitts beruht im Wesentlichen auf den Ausführungen von Weidner [61 S.77-80]. Er beginnt seine Betrachtung der Unterstützung bei der menschlichen Aktivität. Diese kann entweder unterstützt werden oder nicht. Wird der Mensch unterstützt, kann eine weitere Unterscheidung hinsichtlich zeitlich-räumlicher Kopplung getroffen werden, d. h. die Unterstützung kann im selben Raum und zeitlich nah gekoppelt stattfinden. Als „Kopräsent“ wird dabei die Form der Unterstützung bezeichnet, bei der der Nutzer unmittelbar Unterstützung erfährt, also keine Verzögerung in der Unterstützung wahrnimmt. Dabei darf die Unterstützung nicht durch Bedienung des Nutzers ausgelöst worden sein, wie etwa bei einem Lastenkran mittels Bedienelementen, sondern muss durch Sensorik den Nutzer erfassen und auf ihn reagieren. [61 S.77] Als „Integriert“ wird die Form der Unterstützung bezeichnet, bei der zur Ausführung der Aktivität des Nutzers die Unterstützung zwingend erforderlich (konstitutiv) ist und ohne die die Aktivität in der Form nicht durchführbar ist. [61 S.78] Als „Getrennt“ wird die Form der Kopplung verstanden, die lediglich einem Nutzer die Tätigkeit erleichtert, die er aber prinzipiell auch ohne Unterstützung durchführen könnte. Dies kann auch als Assistenz bezeichnet werden. [61 S.88] Auch bei verteilten Systemen kann die Form der Unterstützung integriert oder getrennt sein. Als Beispiel hierfür kann die Mensch-Maschine-Kooperation angeführt werden, wenn etwa Maschinen den Mitarbeiter unterstützen, der die Tätigkeit auch ohne die maschinelle Unterstützung durchführen könnte. [61 S.79] Zur Klärung der dritten Klassifizierungsdimension wird die Frage der Kontrollhoheit über den unterstützten Prozess gestellt. Entweder liegt

die Kontrollhoheit beim Menschen oder bei der Maschine. [61 S.79f.] Das Klassifizierungsschema ist sequenziell ausgelegt, sodass der Einteilungsversuch bei der zeitlich-räumlichen Beziehung beginnt und bei der Kontrollhoheit endet.

Eine weitere Unterscheidung mit konkreterem Bezug auf die Arbeitswelt, die Weidner vornimmt, ist in Abbildung 10 dargestellt. Danach gibt es Systeme, die den Menschen unterstützen (z. B. Exoskelett) und Systeme, die den Menschen ersetzen/substituieren (z. B. Industrieroboter). [Vgl. 61 S.14]

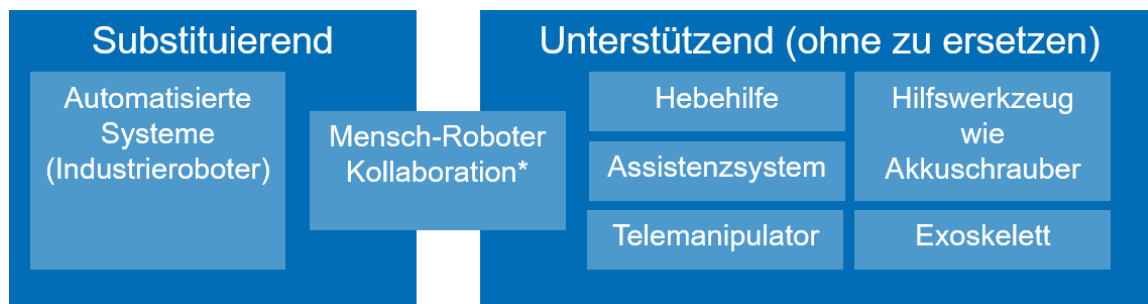


Abbildung 10: Domainenübergreifende Systematik für Unterstützungssysteme [Eigene Darstellung nach Weidner, Redlich, Wulfsberg [61 S.14]]

3.4.2 Beschreibung von Assistenzsystemen nach Wandke

Zur Einordnung von Assistenzsystemen erläutert Wandke zunächst den Begriff „Assistenz (-system)“ genauer. Demnach ist Assistenz gegeben durch Systeme, die die menschlichen Fähigkeiten erweitern. Er gibt aber zu bedenken, dass nach dieser Definition „jedes technische Artefakt zwischen Büroklammer und Raumfähre“ [62 S.5, Übersetzt aus dem Englischen 63 S.132] ein Assistenzsystem wäre. Daher kann diese Definition des Begriffes „Assistenz“ nicht hilfreich zur Systematisierung von Assistenzsystemen herangezogen werden. Zur Einordnung eines Systems als Assistenzsystem legt Wandke zwei Bedingungen fest: Die erste Bedingung besagt, dass das System interaktiv sein muss. Das heißt, teilweise müssen Funktionen dem Menschen und teilweise der Maschine bzw. dem System oder Computer zugewiesen werden. Daher werden voll automatisierte und gänzlich manuelle Systeme nicht den Assistenzsystemen zugeordnet. Die zweite Bedingung ist, dass die den Maschinen immanente Funktionen dem Nutzer zugänglich sein müssen. Ist eine maschinenseitige Funktion nicht existent oder nicht bekannt oder ihre Ausführung mit zu hohem Aufwand verbunden, sodass diese nur rudimentär, ineffektiv oder ineffizient genutzt werden kann, so wird dem Assistenzsystem die grundlegende Funktion zu assistieren (für betrachtete Assistenzfunktion) abgesprochen. Abbildung 11 zeigt die 6 Stufen („Stages“) der Assistenz nach Wandke [62 S.12-14, 63 S.136]:

1. Motivation, Aktivierung und Zielsetzung
2. Wahrnehmung
3. Integration von Information unter Berücksichtigung der aktuellen Situation
4. Fällen von Entscheidungen, Auswahl von Aktionen
5. Ausführung von Aktionen

6. Verarbeitung von Effekten der Aktionen und Rückmeldung (Feedback) darauf

Die Einteilung von Wandke orientiert sich stark an Assistenzsystemen aus dem Automobilbereich. Dies wird anhand seiner angebrachten Beispiele deutlich (automatisiertes Getriebe, elektrische Fensterheber, Antiblockiersystem etc.).

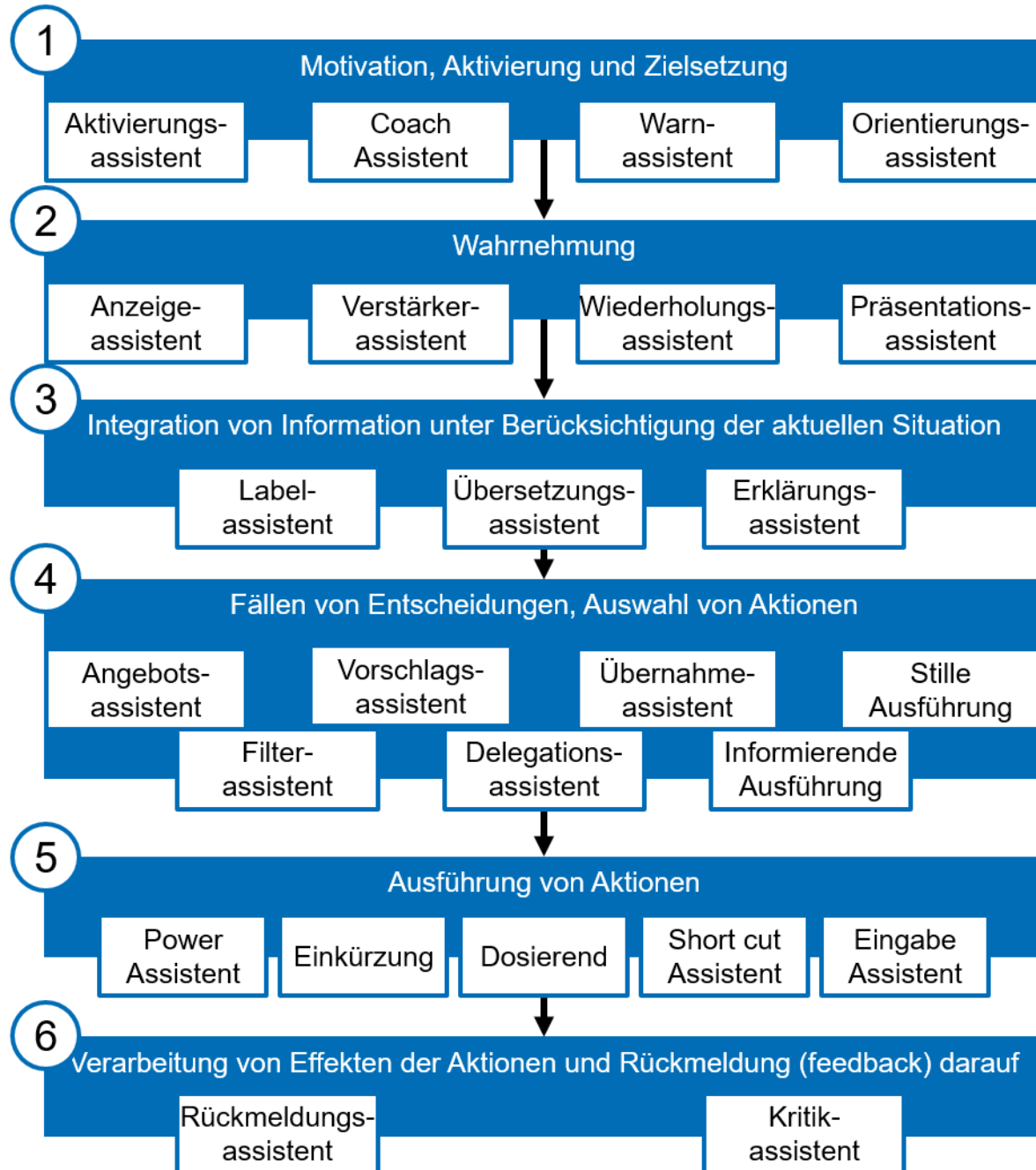


Abbildung 11: Stufen der Assistenz [Eigene Darstellung nach Wandke [63 S.136] übersetzt nach [64]]

Assistenzsysteme erster Stufe motivieren und aktivieren den Nutzer. Zudem werden auf dieser Stufe dem Nutzer Ziele vorgegeben. Unterkategorien dieser Stufe sind **Aktivierungsassistenten**, die den Nutzer stetig auf einem Aufmerksamkeitsniveau halten. **Coach-Assistenten** motivieren den Nutzer zur Aufrechterhaltung einer Aktivität,

wie beispielsweise die Ansage in einer Telefonschleife „Bitte nicht auflegen!“. **Warnassistenten** weisen den Nutzer auf mögliche ungewollte Ergebnisse bei Beibehaltung der Aktion hin. **Orientierungsassistenten** zeigen das Ziel und setzen es immer wieder neu. [63 S.136-137]

Assistenzsysteme zweiter Stufe betreffen die Wahrnehmung des Nutzers. Ziel der Anzeigesassistenten ist die angemessene Bereitstellung von Signalen und Informationen. Verstärkerassistenten verstärken Signale oder Informationen. Wandke nennt bspw. Systeme zur Verbesserung der Sicht („Vision Enhancement systems for driving in darkness or fog.“ [63 S.138]). Zur **Verstärkerassistenten** zählt er ebenfalls Systeme, die zu einem klareren Signal oder Information bspw. mittels Rauschunterdrückung führen. Die **Wiederholungsassistenten** präsentiert die gleiche Information mehrfach etwa, wenn im Fahrzeug auf die Geschwindigkeitsbegrenzung erneut hingewiesen wird, obgleich ein Straßenschild die gleiche Information vermittelt. **Präsentationsassistenten** präsentieren Informationen in unterschiedlicher Art und Weise. Beispiel hierfür sind Text-to-Speech Systeme, bei denen ein visuelles (textuelles) Signal in ein akustisches (sprachliches) Signal umgewandelt wird. [63 S.137-139]

Assistenz in der dritten Stufe zielt auf die (kontextbezogene) Interpretation von Signalen ab. **Labelassistenten** unterstützt durch Kennzeichnung von Signalen und Steuerungen mittels Text, Bilder oder Symbolen. Beispiele hierfür sind Symbole auf der Fernbedienung. Erklärungsassistenten geht über das Kennzeichnen hinaus und erläutert Informationen bspw. durch Kommentare. **Übersetzerassistenten** übersetzt Signale in eine andere Sprache, wobei Sprache hier nicht allein als verbales Werkzeug zur zwischenmenschlichen Kommunikation zu verstehen ist, sondern weitergefasst wird z. B. indem Symbole miteingeschlossen werden. Ein Beispiel hierfür ist die Verwendung von Universalfernbedienungen zum Bedienen mehrerer Geräte (unterschiedliche Sprachen). Hier bewirken die gleichen Tasten (Lautstärkenregelung, Kanalwechsel) gleiche Aktionen auf unterschiedlichen Geräten. [63 S.139-140]

Die vierte Stufe der Assistenz betrifft die Entscheidungsfindung. Die **Angebotsassistenten** stellt dem Nutzer alle Möglichkeiten zur Auswahl einer Aktion zur Verfügung. Die **Filterassistenten** stellt lediglich eine Auswahl von Aktionen bereit. Die **Vorschlagsassistenten** bietet dem Nutzer lediglich eine Option zur Auswahl der Aktion an. Der Nutzer darf diese Option allerdings ignorieren. Der **Delegationsassistenten** bietet ebenfalls nur eine Aktion an, übernimmt aber auch die Ausführung der Aktion, wenn der Nutzer dies gestattet. Der **Übernahmeassistenten** übernimmt die Ausführung der Aktion ebenfalls automatisiert, es sei denn, der Nutzer unterbindet dies aktiv. Die **Informierende Ausführung** übernimmt die Aktion ebenfalls und teilt diese dem Nutzer auch mit. Der Nutzer hat allerdings nicht die Möglichkeit auf die Aktion Einfluss zu üben, es sei denn, er deaktiviert das Assistenzsystem. Beispiel hierfür ist der Autopilot. Die **Stille Ausführung** operiert im Hintergrund und setzt den Nutzer darüber nicht in Kenntnis. Ein Automatikgetriebe ist hierfür ein Beispiel. [63 S.140-143]

Die fünfte Stufe zielt auf die Ausführung ab. Die **Powerassistenz** verstärkt die vom Nutzer aufgebrauchte Kraft (bspw. Bremskraftverstärker). **Einkürzung** regelt die vom Nutzer aufgebrauchte Aktion ab (Abriegelnder Tacho). Die **Dosierungsassistenz** kombiniert Verstärkungsassistenz mit der Einkürzung. Die **Short Cut Assistenz** kürzt lange Prozesssequenzen auf einige wenige Prozessschritte runter. Beispielsweise kann ein Gang über die Gangschaltung eingelegt werden. Es ist aber auch möglich, durch Tippbewegungen einen Gang zu schalten. Die Eingabeassistenz bietet dem Nutzer unterschiedliche Modalitäten zur Eingabe. [63 S.143-144]

Die Assistenz sechster Stufe betrifft die Verarbeitung von Aktionen. Zum einen kann der Nutzer eine simple **Rückmeldung** erhalten und zum anderen kann die Rückmeldung auch bewertet sein, was dann der **Kritik** entspricht. [63 S.144-145]

3.4.3 Klassifikation von digitalen Assistenzsystemen nach Apt

Einen auf die Produktion fokussierten Ansatz der Klassifizierung von (Digitalen) Assistenzsystemen liefert Wenke Apt. Danach lassen sich Assistenzsysteme in drei Dimensionen einteilen, die in Abbildung 12 dargestellt werden:

- Grad der Unterstützung
- Art der Unterstützung
- Zielsetzung der Unterstützung



Abbildung 12: Klassifizierungsansatz für (Digitale) Assistenzsysteme ¶[Eigene Darstellung nach Apt [65 S.19]]

Der Grad der Unterstützung ordnet die assistierende Tätigkeit hinsichtlich des Unterstützungsanteils bei der Bewältigung der Arbeitsaufgabe ein. Hierfür nennt Apt die Kategorien: Niedrig, Mittel, Hoch und Variabel. Die Art der Unterstützung wird in drei Klassen unterteilt, namentlich [65 S.20-21]:

- Physische Unterstützung, die darauf hinweist, den Nutzer körperlich zu entlasten
- Sensorische Unterstützung, die auf die Wahrnehmungsfähigkeit des Nutzers abzielt
- Kognitive Unterstützung, die dem Mitarbeiter bei der Entscheidungsfindung unterstützt

Zuletzt folgt die Einteilung nach Zielsetzung der Unterstützung. Diese kann erfolgen, weil der (benachteiligte) Nutzer ohne Unterstützung eine in der Regel durch Menschen erfüllbare Aufgabe nicht umsetzen kann (**Kompensatorisch**). Als „**Erhaltend**“ wird dabei die Form der Unterstützung bezeichnet, ohne die der Nutzer die Aufgabe nur unter erschwerten Bedingungen und unter Inkaufnahme langfristiger körperlicher Schäden erfüllen kann. Diese Tätigkeiten werden häufig als unergonomisch bewertet. **Erweiternde Assistenzsysteme** erweitern die Möglichkeiten des Nutzers und erweisen sich als nutzbringend bei Aufgaben, die Menschen in der Regel nicht umsetzen können. [65 S.29]

In einem zweiten Klassifikationsversuch für Digitale Assistenzsysteme unterscheidet Apt zwei „Unterstützungsansätze“: physisch und kognitiv. Da in dieser Arbeit der Fokus auf kognitiver Unterstützung liegt, werden diese weiter betrachtet. Kognitive Assistenzsysteme lassen sich nach Apt in drei weitere Systeme unterscheiden: Hilfesysteme, Adaptive Assistenzsysteme und Tutorielle Assistenzsysteme. Als „**Hilfesysteme**“ werden Kognitive Assistenzsysteme bezeichnet, die dem Nutzer „analoges Wissen in digitaler Form [bereitstellen]“, wie etwa digitale Handbücher. **Adaptive Assistenzsysteme** betrachten den Arbeitsfortschritt des Nutzers und stellen Informationen als Entscheidungsgrundlage kontextbezogen bereit. **Tutorielle Assistenzsysteme** sind ähnlich zu den Adaptiven Assistenzsystemen, rücken aber die Qualifikationskomponente („lernförderliche Arbeitsumgebung“) in den Vordergrund. Somit liegt das Hauptaugenmerk auf der (Weiter-)Bildung des Nutzers. [66 S.8]

3.4.4 Intelligente Tutorielle Systeme

Mit den Tutoriellen Assistenzsystemen können Intelligente Tutorielle Systeme (Intelligent Tutoring Systems) assoziiert werden. Da Assistenzsysteme den Fokus vornehmlich auf die Unterstützung des Menschen legen, wohingegen Intelligente Tutorielle Systeme den Lernerfolg fokussieren, sind beide Arten zunächst getrennt zu betrachten. Allerdings können hierbei Synergien genutzt werden. Assistenzsysteme können gleichzeitig eine lernförderliche Umgebung schaffen, die dem Mitarbeiter das Erlernen der Tätigkeiten erleichtert und unter Umständen zusätzlich zu einer Verbesserung der Effizienz beitragen.

3.4.5 Montageassistenzsysteme

In diesem Kapitel wird eine Auswahl herkömmlicher Montageassistenzsysteme vorgestellt. Zudem wird ein kurzer Überblick über Ziel und Funktionsumfang dieser Montageassistenzsysteme gegeben.

Büttner stellt ein Assistenzsystem vor, das wie ein Montagearbeitsplatz aufgebaut ist. Ziel des Assistenzsystems ist das Erkennen des Montagefortschritts sowie die Einblendung zugehöriger Prozessinformationen. Dabei ist ein Montagearbeitsplatz mit assistierenden Betriebsmitteln aufgerüstet. In dem Assistenzsystem sind ein Schrauber, Positionssensoren zur Erfassung von Betriebsmittelpositionen sowie ein LED-Projektor und zwei Tiefenbildkameras integriert. Das Assistenzsystem verfolgt den Montagefortschritt durch das Überwachen von Hand bzw. Handbewegungen mittels Tiefenbildkamera. Die Kameras decken den gesamten Montagebereich ab. Vor der Inbetriebnahme werden dem Assistenzsystem mittels Hüllkörper (Bounding Boxes) Positionsdaten (im Kamerabild) bestimmter Materialboxen mitgegeben. Durchdringt eine Hand einen Hüllkörper, wertet das Assistenzsystem dies als einen Entnahmeprozess, der zu einem Montagefortschritt führt.[67]

Banner präsentiert ein Assistenzsystem, das ebenfalls einen aufgerüsteten Montagearbeitsplatz darstellt. Der Arbeitsplatz ist mit einer über der Arbeitsplatte angebrachten Kamera ausgerüstet. Des Weiteren beinhaltet das Assistenzsystem einen Bildschirm, einen Projektor, der Informationen auf die Arbeitsplatte projiziert sowie zwei Fußtaster. Mithilfe der Fußtaster kann der Montagemitarbeiter am Arbeitsplatz im Prozess vor- bzw. zurückschreiten. Damit wird nicht etwa der Prozess rückgängig gemacht, sondern die mit dem Prozess verknüpften Informationen und Anweisungen weiter oder zurückgeschaltet. Mit der Kamera wird sowohl die Hand des Montagemitarbeiters als auch ein Materialbehälter erkannt. Kommt es zu einer Überschneidung der Handposition mit der Position eines erkannten Materialbehälters, wird bei Überschreiten eines zeitlichen Schwellenwertes auf die Entnahme des Materials geschlossen. Dabei ist dem System der Inhalt des Materialbehälters bekannt. Im Assistenzsystem ist eine Produktstruktur in Form eines Graphen hinterlegt, sodass unter Außerachtlassung von Demontageprozessen alle möglichen Prozessalternativen berücksichtigt werden können. Gemeinsam mit der Kenntnis über den Behälterinhalt wird der Ast des Graphen weiterverfolgt. Ein beiläufiger Vorteil dieser Entwicklung ist die freie Positionierbarkeit des Materialbehälters.[68]

Gorecky et. al. präsentieren ein aus dem Forschungsprojekt COGNITIO hervorgegangenes Assistenzsystem, das eine selbstlernende Komponente besitzt. Dazu nutzt das System eine Kombination aus Bewegungsarten des Montagemitarbeiters, die zum einen aus der in der Arbeitskleidung des Mitarbeiters integrierten Inertialsensorik bestehen und zum anderen aus Bilddaten von einer Kamera. Dabei werden Bewegungsdaten des Montagemitarbeiters ausgewertet. Aus den Bewegungsdaten lassen sich Mus-

ter ableiten, die einen Rückschluss auf den Montageprozess zulassen können. Basierend darauf ist es dem Assistenzsystem möglich, Informationen kontextsensitiv einzublenden.[69]

Lampen et al. stellen ein Assistenzsystem am Beispiel der Automobilmontage vor, das dem Mitarbeiter über eine Augmented-Reality-Brille (AR-Brille) Informationen kontextbezogen anzeigt. Dazu wird zunächst ein virtuelles Modell der Arbeitsumgebung erstellt. Zur Assistenz wird die reale Arbeitsumgebung erfasst, wobei Änderungen in der realen Welt in die digitale Welt übertragen werden. Im digitalen Modell wird der bevorstehende Montageprozess simuliert, wobei das Ergebnis als Information in die AR-Brille übertragen wird. Dem Mitarbeiter wird die nächste Prozessanweisung somit visuell mitgeteilt. [70]

Wiesbeck stellt eine „Struktur zur Repräsentation von Montagesequenzen für die situationsorientierte Montageassistenz“ vor. Ziel ist es, dem Mitarbeiter unterschiedliche bzw. geeignete Montagesequenzalternativen zur Verfügung zu stellen. Er greift hierbei auf eine Graphenstruktur [22 S.134] zurück, die das Produkt in seinen möglichen Bauzuständen repräsentiert. Die Knoten des Graphen stellen die Produktzustände dar, wobei diese als Binärvektoren beschrieben werden. [22 S.83] Durch Anwendung eines „Online-Algorithmus“ wird eine Montagesequenz ausgegeben, die das Komplexitätsmaß und die verfügbaren Ressourcen berücksichtigt [22 S.134].

Zheng et al. präsentieren ein Assistenzsystem zur Unterstützung bei der Verkabelung in Flugzeugen. Das Assistenzsystem nutzt eine tragbare Kamera zusammen mit einer AR-Brille. Mithilfe der Kamera können visuelle Prüfungen vorgenommen werden. Beispielsweise ist es möglich, den Text auf einem verlegten Kabel zur Bestimmung des Kabeltyps oder die Variante eines Kabelspanners zu erkennen. Das AR-Device wird zur Informationseinblendung bezüglich Kabelmontage verwendet.[71]

3.4.6 Diskussion

Der Begriff „Assistenzsystem“ umfasst eine breite Anzahl an unterstützenden Systemen. Der allgemeinen und umfassenden Betrachtung Wulfsbergs [61] folgend, sind Assistenzsysteme in vielen Felder wie bspw. der Medizin, dem Endverbrauchermarkt oder in der Industrie vorzufinden. Je nach Definition sind einfachste Systeme als Assistenzsysteme einzuordnen, wenn diese eine bestimmte Form der Unterstützung anbieten. Apt hingegen betrachtet Assistenzsysteme aus dem Produktionskontext heraus. Im Produktionsbereich hat sich in den letzten Jahren der Begriff des „Kognitiven Assistenzsystems“ für Systeme zur Mitarbeiteranweisung insbesondere in der Montage etabliert. Der Begriff wird bspw. im Themenkurzprofil Nr. 38 „Kognitive Assistenzsysteme“ von Marc Bovenschulte genutzt [60]. Eine weitere Nutzung des Begriffs im Zusammenhang mit der Montage findet sich im Artikel „Kognitive Assistenzsysteme in der manuellen Montage“ von Zäh et al. [72]. Auch im Rahmen dieser Arbeit wird daher der Begriff vor dem Hintergrund der Montage genutzt. Ein weiterer, allerdings in der Begriffszusammensetzung unzureichender Begriff für Kognitive Assistenzsysteme lautet Werkerführungssystem.

Viele bestehende Assistenzsysteme, aber auch Forschungsansätze fokussieren zu meist Produktionsbereiche mit wenig Freiheiten und Varianz in den Prozessen. Daher sind die meisten Arbeiten kaum für den Einsatz in dynamischen Umgebungen wie dem Nacharbeitsbereich geeignet. Ein großer Fokus der Forschungsarbeiten liegt zudem auf der Anwendung von immersiven Technologien, wie bspw. Augmented Reality, da diese den Vorteil bieten, Information unmittelbar in das Sichtfeld des Mitarbeiters einzublenden. Hierdurch werden Mitarbeiter zwar ergonomisch(er) unterstützt, aber nicht alle Produktionsbereiche können hierdurch abgedeckt werden. Wiesbeck präsentiert in seiner Arbeit ein Assistenzsystem, das auf die klassischen Methoden zur Repräsentation von Montagesequenzen zurückgreift [72]. Dieser Ansatz wird auch im Verlauf dieser Arbeit als Grundlage für das Assistenzsystem gewählt.

3.5 Gestaltung Digitaler Assistenzsysteme unter Berücksichtigung menschlicher kognitiver Leistungen

Digitale Kognitive Assistenzsysteme stellen mit dem Menschen interagierende Systeme dar. Daher fallen Assistenzsysteme unter die Mensch-Maschine-Systeme mit Interaktionsmöglichkeit. Neben der technischen Fähigkeit der Assistenzsysteme bietet die kognitionsgerechte Auslegung dieser Systeme bei optimierter Auslastung einen weiteren Mehrwert.

Zur Beschreibung menschlicher kognitiver Fähigkeiten sowie deren Beanspruchung existieren unterschiedliche Ansätze. Abbildung 13 stellt die Kanäle der menschlichen Interaktionsmöglichkeiten mit der Maschine dar. Der Mensch nimmt hierbei Informationen über seine Sinne wahr. Gleichzeitig besitzt der Mensch Aktionsmöglichkeiten, die er zur Übermittlung von Informationen nutzen kann. Eine Maschine, die mit dem Menschen interagieren soll, muss einerseits genau diese Sinne ansprechen und andererseits durch einen geeigneten Sensor die Aktionen des Menschen bspw. in Form von Sprache oder Bewegung/Gestik aufnehmen.

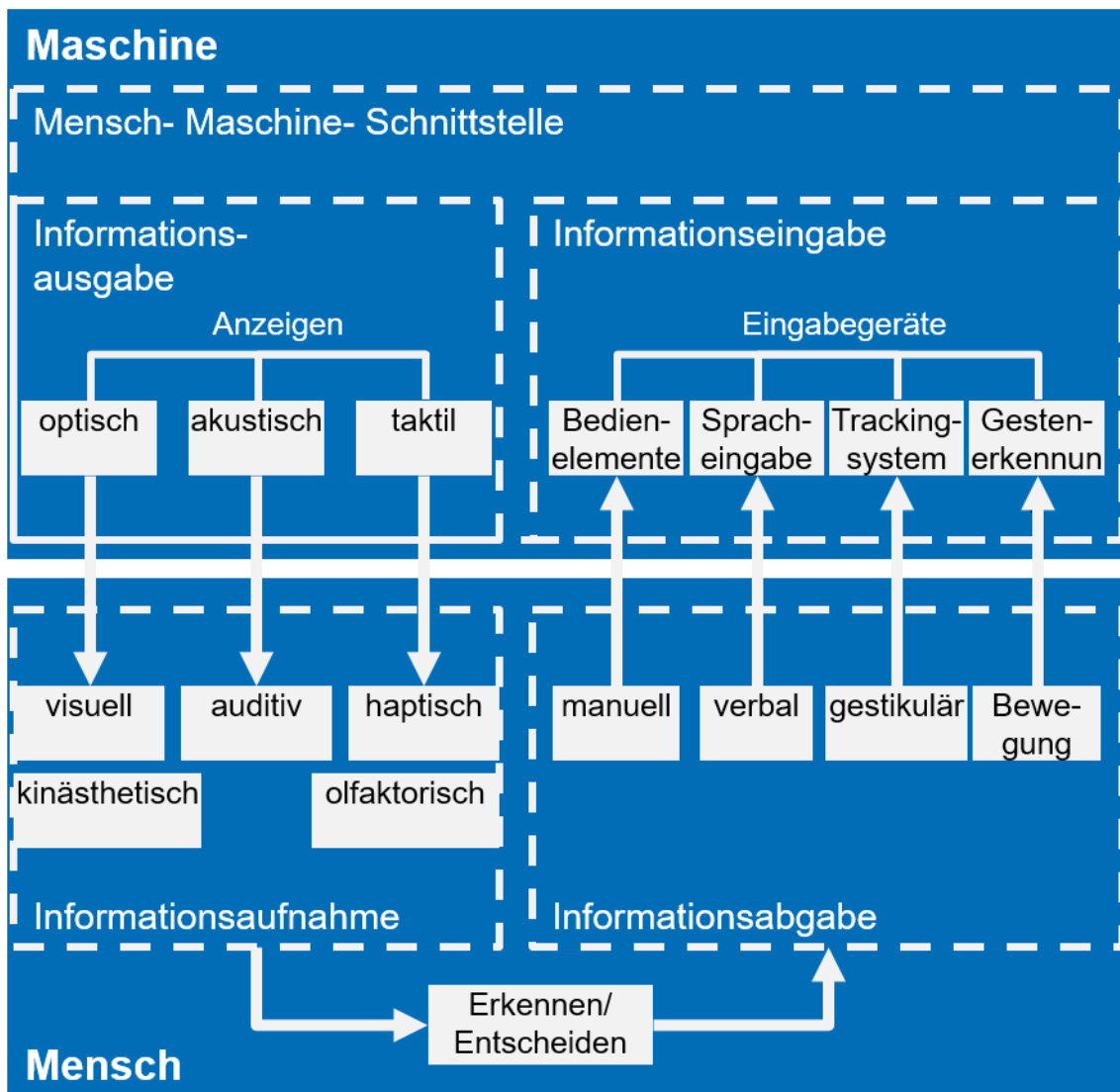


Abbildung 13: Informationsschnittstellen in Mensch-Maschine-Systemen [Eigene Abbildung angelehnt an [62 S.971]]

In den folgenden Kapiteln werden einige Modelle samt der dahinterliegenden Theorie erläutert.

3.5.1 Theorie zur Kognitiven Belastung – Cognitive Load Theory

Ein Ziel der Assistenz ist die kognitive Entlastung des Mitarbeiters während der Entscheidungsfindung bei komplizierten Sachverhalten. Ein Modell zur Beschreibung der Kognitiven Last wird in der Cognitive Load Theory beschrieben. Die Cognitive Load Theory bedient sich eines dreistufigen Gedächtnismodells. Dieses Modell besteht aus einem Ultrakurzzeitgedächtnis, einem Kurzzeitgedächtnis (auch Arbeitsgedächtnis) sowie dem Langzeitgedächtnis. Hierbei werden sensorische Informationen aus dem Ultrakurzzeitgedächtnis an das Arbeitsgedächtnis übermittelt. Je nach Verarbeitung werden danach Informationen an das Langzeitgedächtnis übergeben. Das Arbeitsgedächtnis ist somit wesentlich für die Übertragung der empfangenen Information an das

Langzeitgedächtnis verantwortlich. Die Cognitive Load Theory nach Sweller beschreibt abstrahiert die Mechanismen des Arbeitsgedächtnisses. Danach steht im Arbeitsgedächtnis eine begrenzte Anzahl an Kapazitäten zur Verfügung, um Informationen zu verarbeiten. Diese werden drei Bereichen zugeordnet, wie Abbildung 14 zeigt: Intrinsic Load, Extraneous Load und Germane Load. [73 S. 87] Der **Intrinsic Load** bezieht sich auf die intrinsische Belastung, die die Lernaufgabe mit sich bringt. [74 S.64] Abstrahiert ausgedrückt bedeutet das, dass es ein Mindestmaß an Kapazitätenvereinnahmung gibt, die allein in der Komplexität des Lerninhaltes begründet liegt. Als Beispiel kann der Vergleich herangezogen werden, dass die Komplexität des Lerninhaltes einer naturwissenschaftlichen Abhandlung deutlich mehr Kapazitäten in Anspruch nimmt als das Erlernen des Alphabets. Der **Extraneous Load** nimmt ebenfalls Kapazitäten ein, die allerdings nicht auf den Lerninhalt zurückzuführen sind, sondern auf die kognitive Last, die bspw. durch die Art und Weise der Wissensvermittlung verursacht wird und daher optimiert werden kann. [74 S.66] Das heißt, der Lerninhalt kann so gestaltet werden, dass hierdurch der Extraneous Load möglichst wenige Kapazitäten einnimmt. Durch Veränderung des Extraneous Loads wird der Lerninhalt nicht verändert. Der letzte Bereich ist der **Germane Load**. Dieser betrifft die Belastung, die allein durch die Verarbeitung der Information (Intrinsic Load) verursacht wird. Im Unterschied zum Intrinsic Load bezieht es sich nicht auf Zusammenhänge und die Struktur der Information (bspw. was ist womit verknüpft), sondern befasst sich mit deren Verarbeitung. [Vgl. 73 S.88] Der Intrinsic Load einer Lernaufgabe lässt sich nicht optimieren und ist somit konstant. Allein durch Optimierung bzw. Minimierung des Extraneous Load können Ressourcen für den Germane Load freigegeben werden.

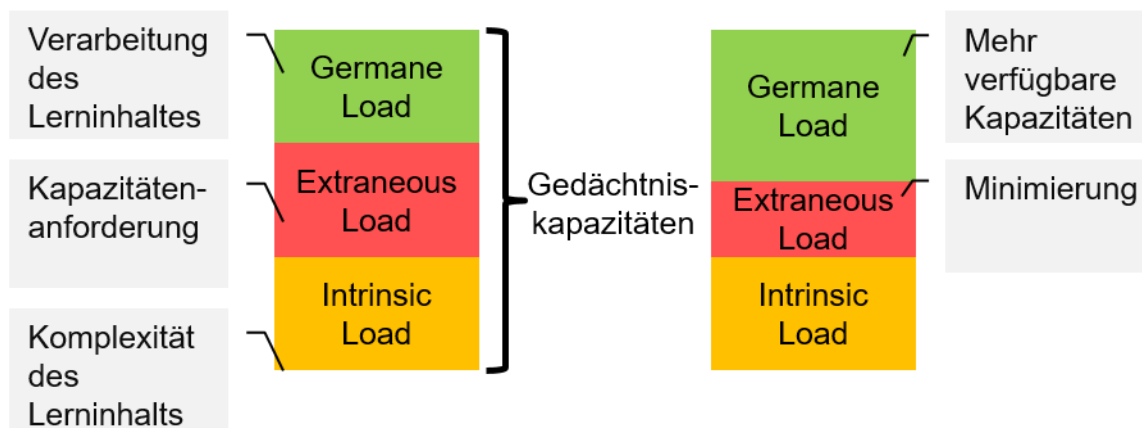


Abbildung 14: Drei Belastungsbereiche des Gedächtnisses in der Cognitive Load Theory nach Sweller [Eigene Darstellung angelehnt an [75 S.38] nach [73 S.88]]

3.5.2 Baddeley's Modell des Working Memory

Ein weiteres Modell zur Beschreibung des Arbeitsgedächtnisses liefern Baddeley und Hitch. Danach unterteilt sich das Arbeitsgedächtnis in vier Komponenten, wobei die **Zentrale Exekutive** die Einheit des Arbeitsgedächtnisses ist, die eine steuernde Aufgabe wahrnimmt („attentional control of working memory“ [76 S. 201]). Hierunter sind

drei weitere Speichereinheiten angesiedelt. Zum einen der **Räumliche-Visuelle Skizzenblock**, der für die Verarbeitung räumlich-visueller Information zuständig ist. Zum anderen die **Phonologische Schleife**, die für die Verarbeitung von sprachlichen Informationen – die sowohl akustisch, aber auch textuell repräsentiert sein können – verantwortlich ist. Zudem besitzt das Arbeitsgedächtnis den **Episodischen Puffer**, der Informationen aus unterschiedlichen Quellen zusammenträgt und diese in Zusammenhang bringt (Chunks). Im Gegensatz zur Zentralen Exekutive, besteht die Hauptaufgabe des Episodischen Puffers in der Abspeicherung von Informationen [76 S. 203].

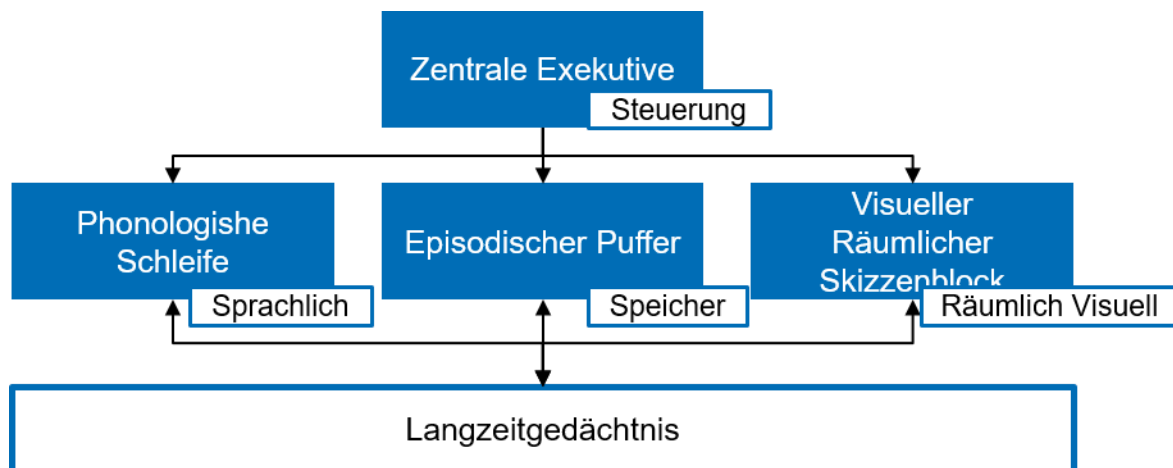


Abbildung 15: Komponenten des Arbeitsgedächtnisses nach Baddeley [Eigene Darstellung mit Ergänzung nach [77 S. 439]]

3.5.3 Multiple - Resource Modell

Wiesbeck betrachtet in seiner Arbeit zu Assistenzsystemen Aufmerksamkeitsmodelle. [22, S.43-45] Eines dieser Modelle ist das Multiple-Ressource-Modell nach Wickens, das die Fähigkeit zur zeitgleichen Widmung der Aufmerksamkeit bei Beanspruchung unterschiedlicher (mentaler) Ressourcen beschreibt. Die folgenden Erläuterungen des Modells basieren auf der Arbeit von Wickens. [78 S.163] Das Modell besteht aus vier Dimensionen: Processing Stages, Perceptual Modalities, Visual Channels und Processing Codes, wie in Abbildung 16 zu sehen ist. [78 S.163]

Die Processing **Stages** geben die Stufen der Informationsverarbeitung an. Ressourcen die Perception (Wahrnehmung) und Cognition (kognitive Verarbeitung) beanspruchen, sind nicht die Gleichen, die die Response, also die Ressourcen zur Reaktion beanspruchen. Je nach Art der Response wird unterschieden in Reaktionen des Menschen manuell-räumlicher oder stimmlich-verbaler Art. [78 S.163-164] Beispiel hierfür ist die sprachliche Erläuterung eines Sachverhalts (verbal) bei gleichzeitig gestikulärer Skizzierung des Sachverhaltes (manuell).

Die Dimension Perceptual **Modalities** beschreibt den Zugriff auf die Ressourcen durch die Sinne, wobei das Modell lediglich visuelle und auditive Wahrnehmungssinne berücksichtigt. Die visuelle Wahrnehmung wird nochmal unterschieden in focal (fokale Wahrnehmung) und ambient (peripherer Wahrnehmung). [78 S.164-165]

Die dritte Dimension wird als Visual **Channels** bezeichnet und bezieht sich auf die Verarbeitung visueller Informationen. Hierbei wird zwischen focal visual processing (fokussierter visueller Wahrnehmung) und ambient visual processing (periphere visuelle Wahrnehmung) unterschieden. [78 S.165-166]

Die vierte Dimension bilden die Processing **Codes**. Die Codes beschreiben die Art der Informationsübermittlung und zeigen die Abgrenzung bei Beanspruchung der Ressourcen bei verbalen (sprachlicher) bzw. räumlich ausgerichteten Prozessen. [78 S.166]

Aufgaben, die dieselbe Ressource erfordern, können nur mit erhöhtem Aufwand zeitgleich erledigt werden. So können Informationen durch Sehen und Hören gleichzeitig verarbeitet werden, wohingegen das Zuhören zweier Personen zur gleichen Zeit schwieriger ist. Dies ist dadurch zu erklären, dass im ersten Fall die Modalities visual und auditory beansprucht werden, wohingegen im zweiten Fall zweimal die Ressource auditory.

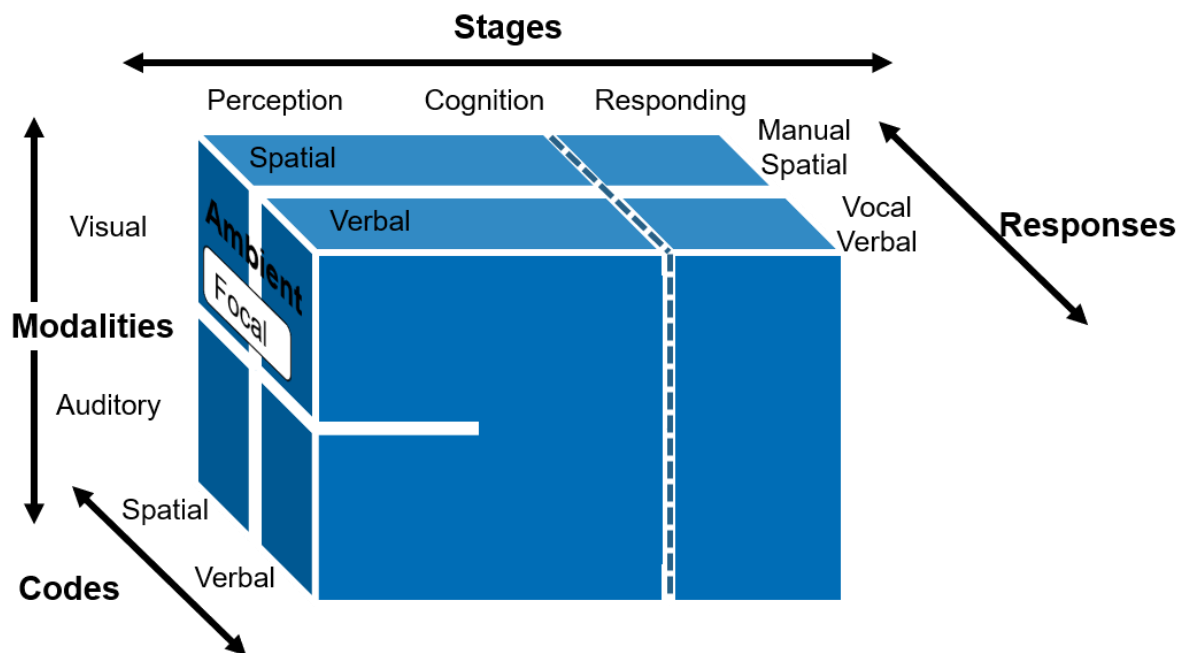


Abbildung 16: Multiple-Resource Modell zur Beurteilung der Aufmerksamkeitslast nach Wickens [Eigene Darstellung nach [22] nach Wickens [78]]

4 Theoretische Grundlagen

In diesem Kapitel werden die für den weiteren Verlauf der Arbeit bedeutsamen theoretischen Grundlagen aufgearbeitet. Da Graphenkonstrukte meist als Grundlage für Prozesse und Prozessabläufe sowie deren Abbildung dienen, wird zuerst die Graphentheorie behandelt. Sie wird oft als Werkzeug zur Beschreibung und Darstellung verketteter Systeme wie bspw. zur Abbildung der Produktion insbesondere auch in der Montage herangezogen [Friedmann nach 79 S.176]. Darauf aufbauend werden Markov-Prozesse erläutert. Hierbei handelt es sich um stochastische Modelle, die häufig zur Entscheidungsfindung genutzt werden. Zudem wird eine Auswahl von Metaheuristiken betrachtet, die Anwendung in der optimierten Lösungsgenerierung findet. Als letztes werden Wartesysteme betrachtet, die sich bspw. zur Modellierung von Angebots- und Nachfrageverhältnissen eignen und bspw. die Verweildauer von Aufträgen im (Produktions-)System beschreiben.

4.1 Graphentheorie

Montagesequenzen können als Graphen dargestellt werden. Daher werden in diesem Kapitel für diese Arbeit relevante Aspekte aus der Graphentheorie abgehandelt.

Ein Graph G ist definiert durch ein Knoten-Kanten-Paar (V, E) , wobei V für eine Menge an Knoten (engl.: vertex) und E für eine Menge von Kanten (engl.: edge) von G steht. [80 S.2] [81 S.11] Die Kanten verweisen ausgehend von einem Knoten u auf einen Knoten v . Beispielsweise gilt für den Graphen in Abbildung 17:

$V = A, B, C, D$ mit $E = AB, AC, BC, CD$

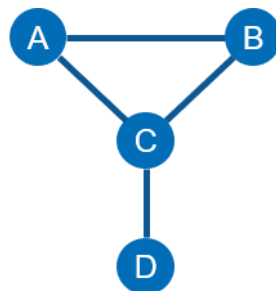


Abbildung 17: Einfacher Graph [Eigene Darstellung]

4.1.1 Grundbegriffe der Graphentheorie

Einführend sollen einige relevante Grundbegriffe aus der Graphentheorie erläutert werden, deren Verständnis ebenfalls für den weiteren Verlauf der Arbeit notwendig ist.

Ein **Knoten** bezeichnet ein Grundelement im Graphen, das ein Objekt repräsentiert [81 S. 1]. Knoten, die direkt auf einen vorhergehenden Knoten folgen, werden als Nachbarn bezeichnet.

Kanten bilden das zweite Grundelement eines Graphen. Sie verbinden zwei Knoten und stellen deren Beziehungen zueinander dar. Jedoch ist die Art der Beziehung nicht vorgegeben. [81 S. 1]

Kanten können Attribute, wie **Gewichtungen** enthalten. Gewichtungen sind ein Maß für den Aufwand, mit dem ein Übergang von Knoten u nach Knoten v über eine Kante $e(u, v)$ verbunden ist. Als Analogie kann das Beispiel einer Wegstrecke von A nach B herangezogen werden, wobei die Gewichtung als Streckenkosten bspw. in Form der Entfernung interpretiert werden kann.

Schleifen werden durch Kanten dargestellt, in denen Start- und Endknoten identisch sind. Allerdings dürfen Schleifen in einem einfachen Graphen nicht vorkommen. [81 S.11]

Ein **Teilgraph** G_1 von G ist gegeben, wenn G_1 in G enthalten ist. Mathematisch gilt die Beziehung: Die Knotenmenge von G_1 , also V_1 ist in der Knotenmenge von G , also V enthalten. $V_1 \subseteq V$. Zusätzlich ist die Kantenmenge E_1 in E enthalten. $E_1 \subseteq E$. [82 S.28]

Als **Pfad** wird ein (Teil-) Graph P (Path) bezeichnet, der aus inneren Knoten und Endknoten besteht. Innere Knoten sind mit zwei weiteren Knoten verbunden, Endknoten dagegen nur mit einem Knoten. In einem Pfad gibt es lediglich zwei Endknoten. [80 S.6]

Vereint man beide Endknoten eines Pfades, so liegt ein Zyklus vor. Liegt ein Pfad in der Form $P = x_0 \dots x_{k-1}$ vor, so gilt für einen **Zyklus** $C := P + x_{k-1}x_0$. [81 S.19] [80 S.8]

Ein **Kantengraph** (eng: Line Graph) $L(G)$ besitzt Knoten analog zu den Kanten von G , wobei Knoten in $L(G)$ benachbart (adjazent) sind, wenn auch die Kanten in G benachbart sind [83 S.8]. Vereinfacht ausgedrückt bedeutet dies, dass unter bestimmten Bedingungen ein Kantengraph $L(G)$ die Knoten eines einfachen Graphen G in Kanten und dessen Kanten in Knoten umwandelt. Abbildung 18a) zeigt einen einfachen Graphen und Abbildung 18b) den zugehörigen Kantengraphen $L(G)$.

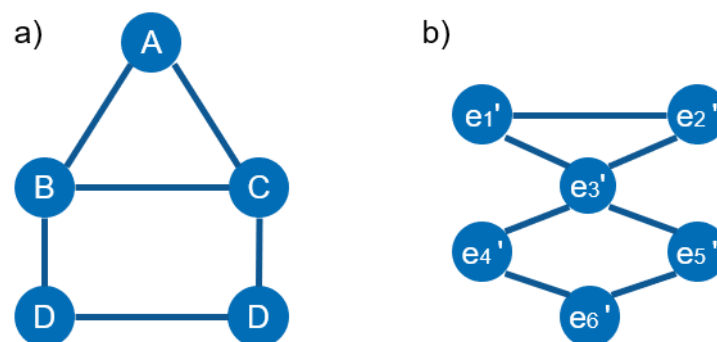


Abbildung 18: Graphenumwandlung: a) Einfacher Graph G ; b) Zu G gehöriger Kantengraph $L(G)$. [Eigene Darstellung]

4.1.2 Arten von Graphen

Graphen lassen sich in unterschiedliche Klassen einteilen. In diesem Kapitel werden die für die Erstellung der Arbeit relevanten Graphenklassen vorgestellt und erläutert.

Eine mögliche Einteilung von Graphen ist die Unterscheidung zwischen gerichteten und ungerichteten Graphen. Erstere enthalten Kanten, die von einem Knoten ausgehen und in einem Knoten enden. Aus Anfangs- und Endknoten wird eine Richtung der Kante vorgegeben. Mathematisch wird ein gerichteter Graph durch zwei Abbildungen $init: E \rightarrow V$ und $ter: E \rightarrow V$ beschrieben. Dabei werden einer Kante ein Ausgangsknoten (initial Node/ $init$) und ein Endknoten (terminal Node/ ter) zugeordnet, wobei die Richtung einer Kante e von $init(e)$ nach $ter(e)$ geht. [80, p. 27 S.27-28]

Ungerichtete Graphen bestehen aus Kanten, die keine Richtung beschreiben.

Eine weitere Form der Unterscheidung ist die Unterteilung in einfache Graphen und Multigraphen [Vgl. 81 S.11]. Für einfache Graphen gilt die Regel, dass zwei Knoten nicht durch mehr als eine Kante verbunden sein dürfen. Einfache Graphen dürfen ebenfalls keine Schleifen aufweisen, d. h. auf sich selbst verweisende Knoten enthalten. [81S. 11] Abbildung 19 stellt verschiedene Klassen von Graphen dar. Während in Abbildung 19a) ein einfacher Graph ohne Schleifen und ohne Mehrfachkanten dargestellt ist, zeigt Abbildung 19b) einen gerichteten Graphen. Die Richtung der Kanten wird dabei durch die Pfeile gekennzeichnet. Abbildung 19c) stellt einen ungerichteten Multigraphen dar, da die Kanten e_4 und e_5 die gleichen Knotenpaare C und D besitzen. Abbildung 19d) dagegen zeigt einen gerichteten Multigraphen, bei dem die Kanten e_4 und e_5 wiederum dem gleichen Knotenpaar (C, D) zugewiesen sind.

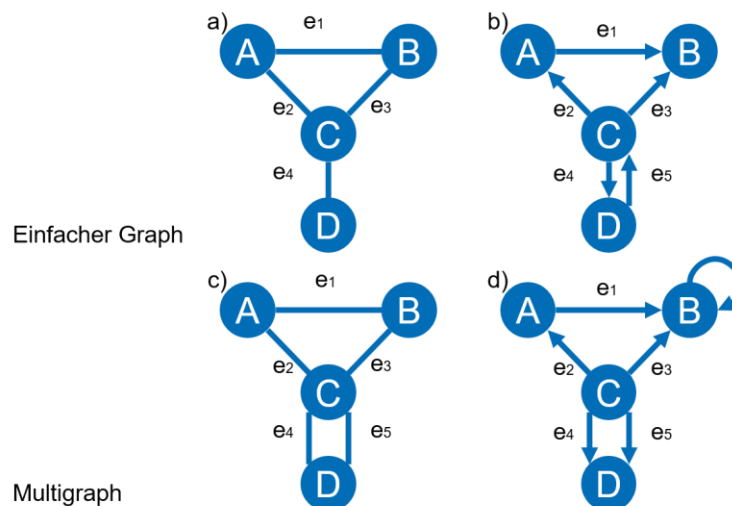


Abbildung 19: Auswahl unterschiedlicher Graphenarten: a) Ungerichteter einfacher Graph, b) Gerichteter einfacher Graph, c) Ungerichteter Multigraph, d) Gerichteter Multigraph. [Eigene Darstellung]

4.1.3 Darstellungsformen und Datenstrukturen von Graphen

Häufig werden für Graphen Diagramme als Darstellungsform gewählt, wobei Knoten durch Kreisflächen und Kanten durch Linien symbolisiert werden, wie in Abbildung 20a zu sehen ist. Allerdings besteht ebenfalls die Möglichkeit, Graphen in Form von Matrizen und Tabellen zu beschreiben, da auch hier die Möglichkeit besteht, Kanten-Knoten-Paare abzubilden. Abbildung 20 stellt verschiedene Darstellungsformen von Graphen dar. In Abbildung 20b ist eine Adjazenzmatrix dargestellt, die in den Zeilen sowie in den Spalten Assoziationen zu Knoten beinhaltet. Ein Eintrag a_{uv} stellt die Verbindung zwischen zwei benachbarten Knoten u und v dar [Vgl.84 S. 70]. In ungerichteten Graphen ist die Adjazenzmatrix symmetrisch. Eine weitere Darstellungsform bilden Inzidenzmatrizen, wie in Abbildung 20c zu sehen ist. Inzidenzmatrizen bilden in den Zeilen die Assoziation zu Knoten ab und in den Spalten die Assoziation zu einer Kante. In einfachen Graphen bedeutet ein Eintrag, dass eine Kante die Endknoten mit dem entsprechenden Eintrag besitzt. Abbildung 20d zeigt eine Adjazenzliste. Abbildung 20e zeigt einen gerichteten Graphen, die zugehörige Adjazenzmatrix in Abbildung 20f enthält jeweils einen Eintrag, wenn der Knoten aus der Zeile eine Kante in Richtung eines Knotens in der Spalte enthält. Die Inzidenzliste eines gerichteten Graphen wie in Abbildung 20g zu sehen, enthält den Eintrag 1, wenn die Kante von dem Knoten weggeführt und eine -1, wenn die Kante zum Knoten hinführt [Vgl. 85 S. 52]. Abbildung 20h zeigt die Adjazenzliste zum gerichteten Graphen.

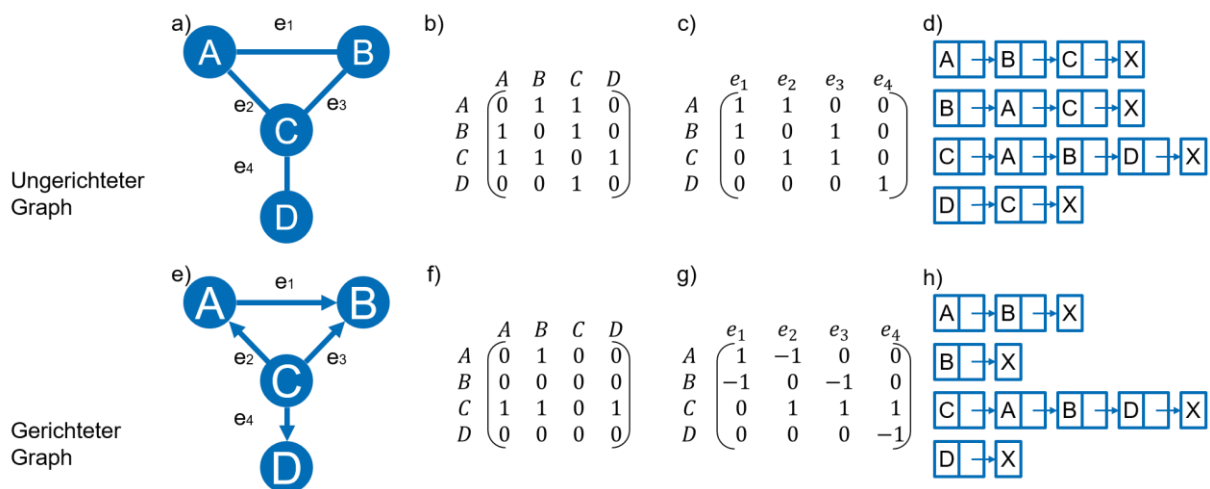


Abbildung 20: Darstellungsformen von Graphen a) Ungerichteter Graph als Diagramm, b) Ungerichteter Graph als Adjazenzmatrix, c) Ungerichteter Graph als Inzidenzmatrix, d) Ungerichteter Graph in Listendarstellung, e) Gerichteter Graph als Diagramm, f) Gerichteter Graph als Adjazenzmatrix, g) Gerichteter Graph als Inzidenzmatrix, h) Gerichteter Graph in Listendarstellung [Eigene Darstellung]

4.2 Metaheuristiken zur Lösung von Optimierungsproblemen

Zur algorithmischen Lösung von Optimierungsproblemen in großen Suchräumen eignet sich der Einsatz von Metaheuristiken. Im Gegensatz zu Heuristiken beschreiben Metaheuristiken Prinzipien, die sich für unterschiedliche Arten von Problemen eignen. [Vgl. 86 S.817-818] Heuristiken hingegen eignen sich lediglich für die Lösung eines bestimmten Problems. [87] Metaheuristiken sind nützlich zur Lösung von Optimierungsproblemen, lösen diese aber nicht vollständig analytisch. Daher kann ihre Lösung lediglich als Näherungslösung betrachtet werden. [88 S.22, 89 S.817f.]

Im Folgenden werden einige bekannte Metaheuristiken vorgestellt.

Hill-Climbing (Bergsteigeralgorithmus)

Der Hill-Climbing-Algorithmus bedient sich der Analogie eines blinden Bergsteigers, der den höchsten Punkt eines Berges sucht, indem er erkennt, ob der nächste Schritt steigend oder fallend ist. [90 S.81]

Der Algorithmus erzeugt eine Initiallösung und evaluiert diese. Im nächsten Schritt wird eine Nachbarlösung, also eine Lösung, die nahe an der ursprünglichen Lösung liegt, generiert und bewertet. Fällt die Bewertung der zweiten Lösung besser aus, wird diese als neue Referenzlösung angenommen. Tut sie es nicht, wird die alte Lösung beibehalten und eine alternative Nachbarlösung ausgewählt. Der Vorgang wird so lange wiederholt, bis keine Verbesserung mehr erzielt wird. Der Algorithmus wird als Greedy-Algorithmus klassifiziert. Das bedeutet, dass der Algorithmus keine Verschlechterung der Situation berücksichtigt. Daher kann, basierend auf der Initiallösung, die Lösung ein lokales Optimum (z. B. Maximum) darstellen und weit vom globalen Optimum abweichen. [91 S 4][92 S. 122f.]

Durch wahrscheinlichkeitsbasierte Verfahren können lokale Optima überwunden werden. Dabei wird versucht, mit einer gewissen Wahrscheinlichkeit auch schlechtere Lösungen zu wählen. In Analogie zum ursprünglichen Bergsteiger-Beispiel kann das bedeuten, dass dieser auch bereit ist, den Berg abzustiegen, um dadurch die Chancen einen höheren Punkt auf dem Berg zu finden, zu erhöhen. [Vgl. 88 S.32]

Genetische Algorithmen

Genetische Algorithmen gehören zur Klasse der Evolutionären Algorithmen, die ihre Motivation aus der Evolutionsstrategie der Natur ableiten, um bestmögliche (am besten angepasste) Lösungen zu generieren. Sie wurden von John H. Holland entwickelt und bekanntgemacht [93 S 481].

Abbildung 21 stellt die Phasen im Ablauf des Genetischen Algorithmus bildlich dar.

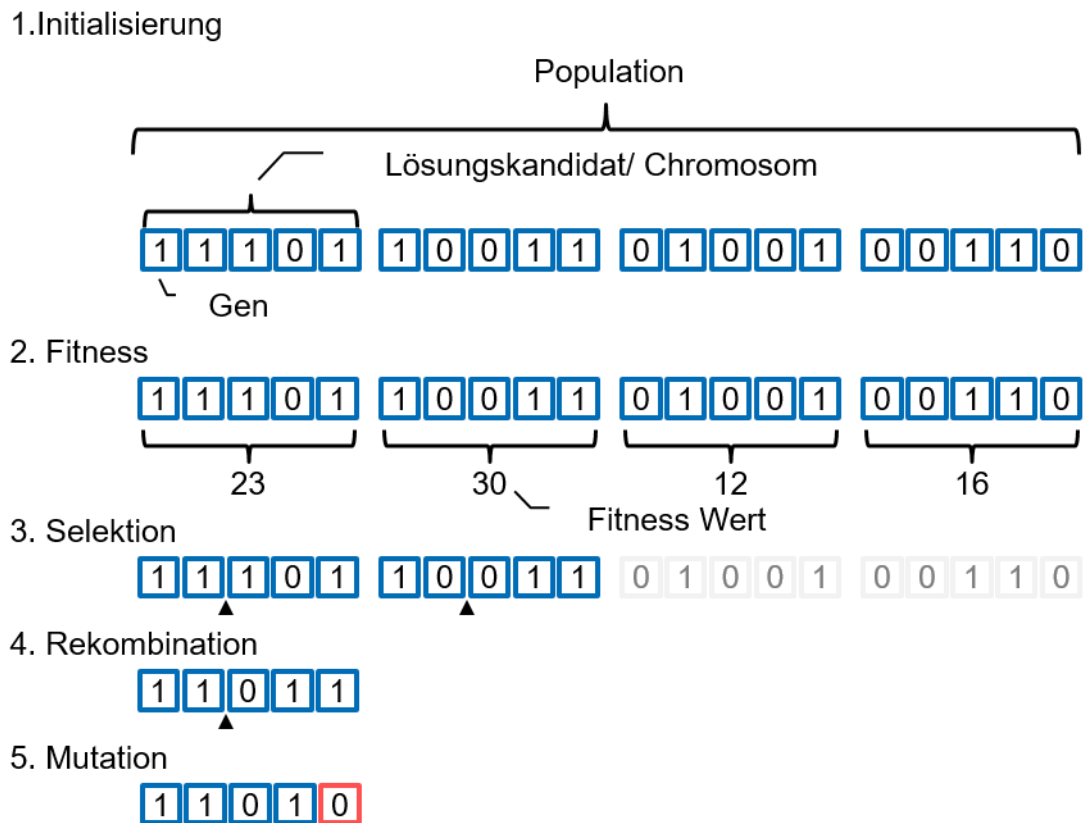


Abbildung 21: Phasen des Genetischen Optimierungsalgorithmus [Eigene Darstellung]

In der Initialisierungsphase, die lediglich zu Beginn auftritt, wird eine Menge beliebiger Lösungskandidaten erzeugt (**Initiale Population**). Die Menge der Lösungskandidaten wird als Population bezeichnet. In der zweiten Phase wird jeder Lösungskandidat mittels Fitnessfunktion bewertet. Diese Bewertung soll dabei helfen, bessere Lösungskandidaten von schlechteren Lösungskandidaten zu unterscheiden. In der dritten Phase werden Lösungskandidaten ausgewählt (**Selektion**), wobei die Auswahlwahrscheinlichkeit nicht gleichverteilt ist, sondern von der Fitness abhängt: Hierzu eignen sich unterschiedliche Methoden. Prinzipiell gilt jedoch: Je besser die Bewertung, desto höher die Auswahlwahrscheinlichkeit. In der vierten Phase werden für je zwei Lösungskandidaten, die nun als Eltern bezeichnet werden, Schnittpunkte gesetzt. Je zwei Paare werden dann so rekombiniert (**Rekombination**), dass sie sich ergänzen. Auch hierfür existieren unterschiedliche Methoden. Unter Beachtung von Randbedingungen gilt oftmals, dass der erste Teil des ersten Elternteils mit dem Rest des zweiten Elternteils zusammengesetzt wird. Somit entstehen aus einem Elternpaar zwei Kinder. Die Kinder werden nun der nachfolgenden Generation zugerechnet. Die Auswahl des Rekombinationsoperators muss gut durchdacht sein, da dieser maßgeblich die Lösungskandidaten erstellt. So erfordern bestimmte Probleme, dass ein Element nicht mehr als einmal vorkommen darf. Bestimmte Rekombinationsmethoden erfüllen diese Bedingung. Im Folgenden wird eine kurze Auswahl dieser Rekombinationsmethoden gelistet [94 S.1083, 1086-1087, 95 S.135]:

- 1-point crossover

- k-Point Crossover
- partially mapped crossover
- order crossover
- order-based crossover
- position-based crossover
- alternating-position crossover
- cycle crossover

Die **Cycle-Crossover**-Methode (CX) ist eine Möglichkeit zur Generierung von Rekombinationen in Zeichenketten. Abbildung 22b stellt den Ablauf der Methode grafisch dar. Die Idee beruht darauf, zwischen zwei Elternchromosomen Zyklen zu bilden. Diese Zyklen stellen die Grundlage zur Erstellung von Bausteinen, aus denen später die Kinder generiert werden. Diese Methode beruht auf der Annahme, dass beide Elternchromosomen aus den gleichen Genen bestehen, wobei lediglich die Sequenz den Unterschied macht. Zunächst wird beginnend beim ersten Gen eines Elternchromosoms (bspw. $P1[0]=A$), das Gen auf der gleichen Position im P2 ($P2[0]=C$) ermittelt. Ausgehend von $P2[0]=C$ wird in P1 die Position des entsprechenden Gens ermittelt, das dem zuletzt ermittelten Gen in P2 entspricht. Im vorliegenden Beispiel wäre die Position des Elementes $P2[0] = C$ in P1 $\rightarrow 2$. Nun beginnt der Prozess von vorne, wobei der Zyklus abgeschlossen ist, wenn die Position des letzten in P2 ermittelten Gens das Gen in P1 ist, mit dem der Zyklus ursprünglich begann. Für das Beispiel aus Abbildung 22b bedeutet das, dass folgende Sequenz gilt:

$$P1[0] = A \rightarrow P2[0] = C \rightarrow P1[2] = C \rightarrow P2[2] = F \rightarrow P1[5] = F \rightarrow P2[5] = A \rightarrow P1[0] = A$$

Mit dem letzten Gen ist der Graph geschlossen (Zyklus). Damit ergeben sich die ersten Bauelemente für die Kinder der nächsten Generation. Hierbei werden Gene, die in P1 sind, auch als Teil von P1 betrachtet, Gene, die Teil von P2 sind, werden als Teil von P2 betrachtet. Somit liefert der erste Zyklus bereits zwei Bausteine. Da nicht alle Gene im ersten Zyklus enthalten sind, wird ein weiterer Zyklus generiert, wobei das erste Element des Zyklus, das erste noch nicht berücksichtigte Gen aus P1 bildet. Im vorliegenden Beispiel ist dies $P1[1]=B$. Der Vorgang wird so lange wiederholt, bis alle Gene in einem Zyklus sind. Ein Kind kann bspw. durch wechselseitige Ergänzung der Bausteine generiert werden. [Vgl. 96 S.224-230] nach [97 S.73-74]

Die **Order-Crossover**-Methode ist in Abbildung 22a dargestellt. Ausgangspunkt sind zwei Elternchromosomen P1 und P2. Diese stellen die bisherige Lösung in Form von Listen bzw. Vektoren dar. Beide Chromosomen werden mit jeweils zwei Schnittpunkten versehen, sodass pro Elternchromosom 3 Submengen entstehen. In Abbildung 22a sind beide Elternchromosomen untereinander positioniert. Aus jedem Elternchromosom wird eine gleichgroße Submenge ausgewählt, deren Gene jeweils austauschbar sein sollen. In Abbildung 22a betrifft dies $D \leftrightarrow H$, $E \leftrightarrow G$, $F \leftrightarrow A$. Somit können zwei Kindchromosomen erstellt werden, die jeweils die Gene H, G, A bzw. D, E, F enthalten. Betrachtet man zunächst nur das erste Kindchromosom (Offspring1 = O1), so enthält es acht (A-H) Plätze, wobei zunächst lediglich drei belegt sind. Für O1 gilt, dass es

sich an P1 orientiert, allerdings werden Genkopien aus P1 durch die Entsprechungen aus P2 substituiert. Existiert keine Entsprechung, wird die Genkopie aus P1 ohne weiteres übernommen. Für O2 gilt die gleiche Herangehensweise, wobei die Genkopie aus P2 entnommen und die Entsprechung aus dem Subset von P2 übernommen wird. [98] nach [97 S.73]

Eine dritte Möglichkeit der Rekombination ist das **Alternating-Position-Crossover (AP)**, das in Abbildung 22c) skizziert ist. Hierbei werden dem Offspring auf jedem Platz im Wechsel (alternierend) die Gene der Eltern (P1 und P2) zugewiesen. Bspw. gilt für $O1[0]=P1[0]$, $O1[1]=P2[1]$, $O1[2]=P1[2]$ etc. Ist ein Chromosom im Offspring bereits vorhanden, wird das Wechselprinzip fortgesetzt und zum nächsten Gen übergegangen. Das Alternating-Position Crossover bietet eine besonders einfache Umsetzungsmethode. Wählt man beim AP verschiedene Startpunkte, so kann eine leichte Varianz in der Nachkommenschaft erzielt werden. [99] nach [95 S.147]

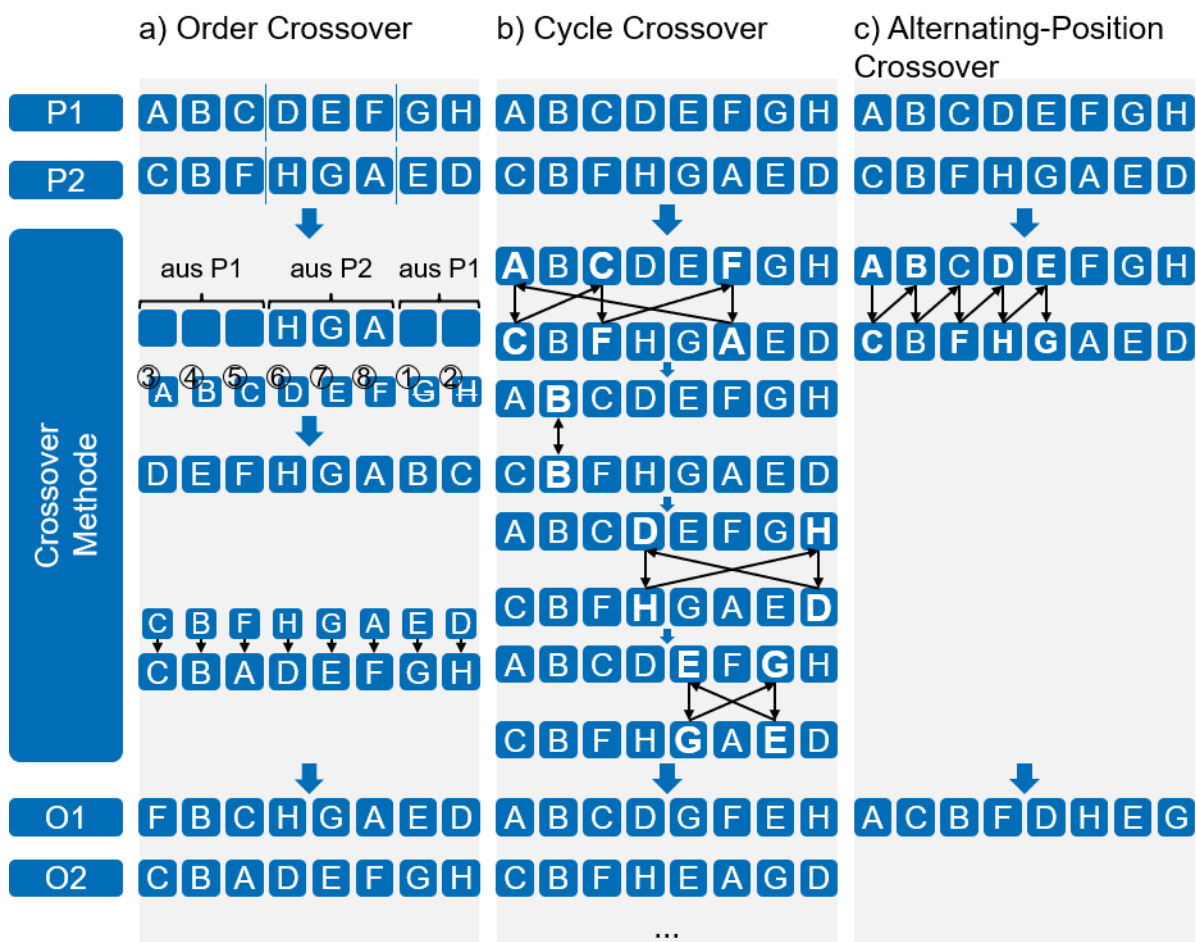


Abbildung 22: Auswahl von Crossover Methoden für Listen bzw. Zeichenketten [Eigene Darstellung]

Die vorgestellten Rekombinationsmethoden stellen lediglich eine Auswahl dar. Jede präsentierte Methode erfüllt die erste Bedingung: Keine Mehrfachnennung von Aufträgen. Allerdings muss die Beibehaltung der Indizes ebenfalls berücksichtigt werden.

Der Cycle Crossover ist eine geeignete Rekombinationsmethode, da er pro Elternpaar die Beibehaltung der Indizes weitestgehend berücksichtigt. Einzelne Verschiebungen können als Mutationsschritt verstanden werden.

Im letzten Schritt findet eine zufällige **Mutation** der Lösung statt. Der Vorgang wird beginnend bei Phase 2 wiederholt, bis ein Abbruchkriterium erreicht ist. [92 S. 127-128] [100 S 21-27]

Simulated Annealing

Das Simulated Annealing bezeichnet eine weitere naturanaloge Metaheuristik, deren Motivation aus der Metallurgie stammt. Die Idee liegt darin, das Glühverfahren bei Metallen nachzubilden. Dies wird ebenfalls durch die Bezeichnung der Parameter des Algorithmus deutlich.

1. Setze $T = T_{Start}$
2. Setze Startlösung $x = x_{init}$
3. Solange Abbruchbedingung nicht erreicht ist
 - Wähle eine Lösung y aus der Nachbarschaft von x
 - Werte die Lösung x aus
 - Wenn $f(y) \leq f(x)$ setze $x = y$
 - Andersfalls setze $x = y$ mit der Wahrscheinlichkeit $e^{-\frac{f(y)-f(x)}{T}}$
 - Reduziere T
 - Prüfe, ob Abbruchbedingung erreicht ist

Initial wird eine zufällige Lösung x_{init} gewählt und eine hohe Temperaturvariable T_{Start} mit einem hohen Wert gesetzt. Diese Lösung wird gegen eine problemspezifische Kostenfunktion bewertet und dient initial ebenfalls als Referenzlösung, die zunächst als akzeptiert gilt. Im nächsten Schritt wird eine zweite Lösung generiert, die aus der Nachbarschaft der ersten Lösung stammt. Auch diese wird mittels einer Kostenfunktion einer Bewertung unterzogen. Ist die zweite Lösung besser als die erste Lösung, wird diese Lösung als aktuelle Referenzlösung übernommen. Ist die zweite Lösung jedoch schlechter als die Referenzlösung, wird diese nur zu einer bestimmten Wahrscheinlichkeit von $e^{-\frac{f(y)-f(x)}{T_t}}$ (inspiriert vom Boltzmannfaktor) übernommen. Anschließend wird die Temperaturvariable T_t gesenkt (Annealing). Dadurch wird über mehrere Iterationen die Wahrscheinlichkeit reduziert, eine schlechtere Lösung zu wählen. Allerdings ist zu Beginn eine höhere Wahrscheinlichkeit zur Annahme auch schlechterer Lösungen wichtig, um lokale Minima verlassen zu können. [Vgl. 101][88 S.59-63]

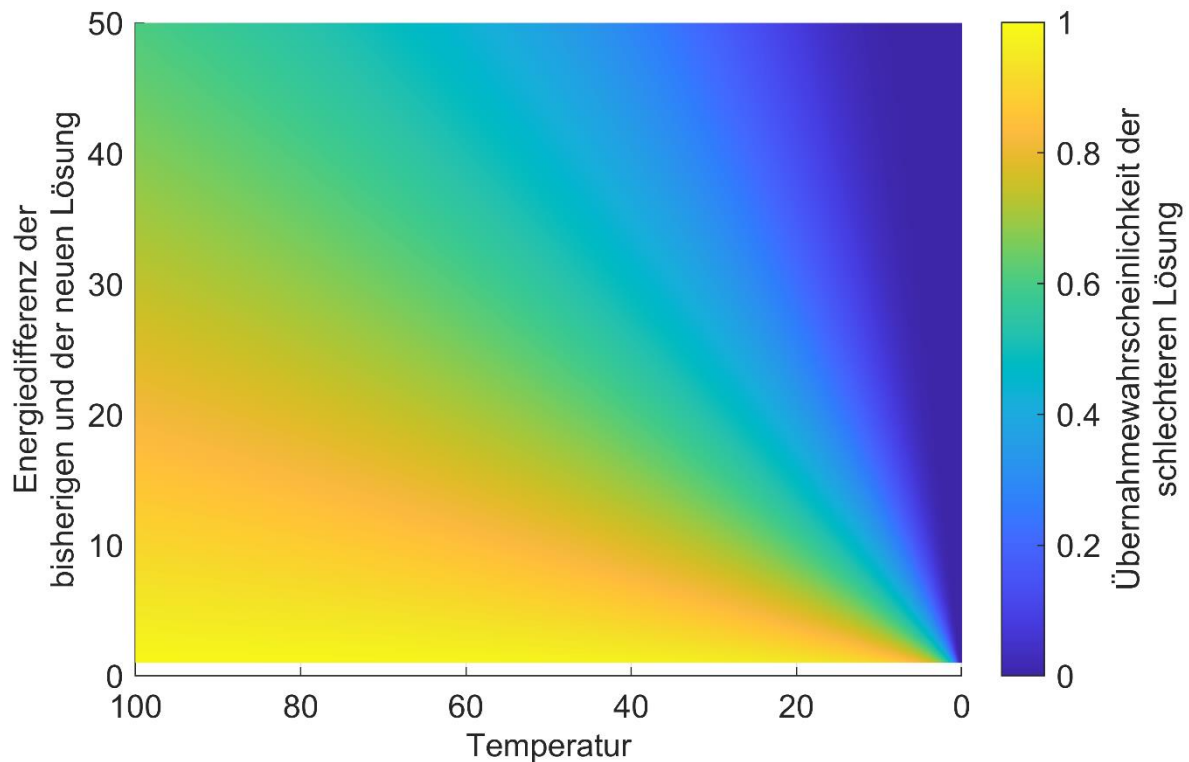


Abbildung 23: Wahrscheinlichkeitsverlauf in Abhängigkeit der Temperatur und Energiedifferenz der Optimierungslösungen [Eigene Darstellung]

Abbildung 23 zeigt die Wahrscheinlichkeit in Abhängigkeit von der Temperatur sowie der Energiedifferenz der neuen und der bisherigen Optimierungslösung. Wird über die Laufzeit des Algorithmus der Temperaturparameter reduziert, so nimmt auch die Wahrscheinlichkeit ab, schlechtere Lösungen zu akzeptieren. Da die Übernahmewahrscheinlichkeit erst herangezogen wird, wenn die Energie der neuen Lösung größer ist als die Energie der alten Lösung $f(y) > f(x)$, genügt eine Differenzbetrachtung im positiven Bereich. Der Algorithmus bricht ab, sobald ein Abbruchkriterium wie bspw. eine hinreichend akzeptable Lösung hinsichtlich ihrer Kosten erzielt wurde.

Ant-Colony-Optimization

Der Ant-Colony-Optimization-Algorithmus ist inspiriert vom Prozess der Nahrungsmittelgewinnung von Ameisen. Diese finden über die Zeit einen kürzesten Weg zwischen ihrem Nest und der Nahrungsquelle [102 S.768]. Finden Ameisen eine Nahrungsquelle, so bringen sie die Nahrungsmittel in ihr Nest. Zu Beginn laufen die Ameisen sowohl längere als auch kürzere Wege ab. Über die Zeit wählen diese dann nur noch vereinzelt die längeren Wege, die Mehrheit beschreitet den kürzeren Weg. Die Begründung liegt darin, dass Ameisen entlang des Weges Pheromone ausschütten, die von anderen Ameisen wahrgenommen werden. Je kürzer der Weg, desto stärker die Pheromonkonzentration entlang des Weges, da der Botenstoff sich noch nicht verflüchtigt hat. Daher folgen weitere Ameisen diesem Weg, wobei diese wiederum Phe-

romone ausschütten. Das beschriebene Verfahren ist in diversen abgewandelten Algorithmen umgesetzt worden. Zwei bekannte Algorithmen lauten Ant System sowie Ant-Colony-System [88 S. 92, 102 S. 767ff.].

Artificial-Bee-Colony

Der Artificial-Bee-Colony-Algorithmus (ABC) stellt einen weiteren populationsbasierten Optimierungsalgorithmus dar. Dieser wurde 2005 von Karaboga vorgestellt. Der Algorithmus ist inspiriert von der Nahrungsquellensuche der Bienen.

Bei der Suche nach einer Nahrungsquelle übernehmen Bienen unterschiedliche Aufgaben. Im ABC wird die Bienenpopulation in drei Arten unterteilt: Die Arbeiterinnen (Employed Bees), die Kundschafterinnen (Scouts) sowie die Sammlerinnen (Onlooker). [103 S.790f.]

Der Algorithmus interpretiert diese Rollen in seinen unterschiedlichen Phasen. Zunächst werden initiale Lösungen generiert. Auf Basis dieser Lösungen werden umliegende Lösungen bewertet. Umliegende Lösungen sind meist Lösungen, die bezugnehmend auf einer Metrik lediglich eine geringe Abweichung zur bisherigen Lösung darstellen. Die bisher ermittelten Lösungen werden im nächsten Schritt einer Bewertung unterzogen, wobei lediglich eine bestimmte Auswahl an Lösungen zur weiteren Betrachtung herangezogen wird. Ist ein Lösungsbereich ausgeschöpft, werden zufällige Lösungen generiert. Ein Lösungsbereich ist ausgeschöpft, wenn keine signifikanten Verbesserungen erzielt werden. Diese Kriterien werden im ABC-Algorithmus vom Anwender vorgegeben. In der ursprünglichen Beschreibung wird der Algorithmus zum Einstellen unterschiedlicher Parameter verwendet. [Vgl.104 S.25f., 105, 106 S.11-12]

4.3 Wartesysteme und Warteschlangentheorie

In Kapitel 4.1 und 4.2 wurden Grundlagen erklärt, die einen direkten Bezug zur Funktion des Assistenzsystems haben. Die in diesem Kapitel erläuterten Grundlagen werden nicht für die Funktion des Assistenzsystems benötigt, sondern für den nachfolgenden Gebrauch der im Assistenzsystem aufgenommenen Erkenntnisse bezüglich der Zuweisung der Aufträge an die Mitarbeiter.

Neben der Assistenz während der Durchführung der Prozesse liegt ein weiterer Fokus auf der zeitlichen Betrachtung der Aufträge insgesamt. Hierzu ist es notwendig, eine Einschätzung zu treffen, wie lange bestimmte Aufträge zur Bearbeitung benötigen. Dabei können Modelle aus dem Bereich der Wartesysteme genutzt werden. Wartesysteme beschreiben Nachfrage-Angebot-Systeme mit dem Anspruch, alle Kunden im System zu bedienen. Hierin liegt der große Unterschied zum Markt, in dem lediglich eine Auswahl von Interessenten zu den Kunden zählen kann. In Wartesystemen kommt es aufgrund begrenzter (Dienstleistungs-)Kapazität zu Wartesituationen. [107 S.1] In diesem Kontext sind Kunden mit eingehenden Aufträgen gleichzusetzen.

4.3.1 Grundaufbau und Notation in einem Wartesystem

Ein Wartesystem (auch Bediensystem) ist ein System, in dem über einen oder mehrere Eingänge Kunden bzw. Nutzer eintreten und eine meist durch begrenzte Kapazitäten charakterisierte Dienstleistung in Anspruch nehmen möchten. Diese Dienstleistungen werden an einer oder mehreren Stationen angeboten. [108 S. 365] Ein typisches Beispiel eines Wartesystems sind Supermarktkassen. Hier müssen sich Kunden aufgrund der begrenzten Anzahl der Kassen (Kapazitäten) in Warteschlangen einreihen, wobei die Dienstleistung von den Kassierern ausgeführt wird. [Vgl. 109 S.15, 110 S.161] Abbildung 24 zeigt ein Diagramm eines einfachen Wartesystems. In der Warteschlange reihen sich die Kunden zur Bedienung ein. Der Ankunftsprozess beschreibt die neu-eintreffenden Aufträge des Wartesystems. An Bedienstationen werden die Dienstleistungen angeboten. Die Leistung der Bedienung wird durch die Abfertigungsprozesse beschrieben.

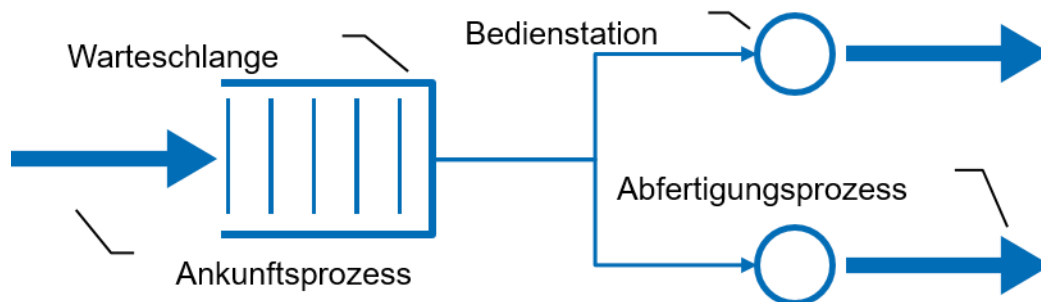


Abbildung 24: Aufbau und Elemente eines einfachen Wartesystems [Eigene Darstellung]

Um Wartesysteme zu beschreiben, werden unterschiedliche Teilsysteme dargestellt, wobei hierfür die Notation nach Kendall verwendet wird. [111 S.340] [108 S.366] [112 S.80] [113 S. 108].

Die Kendallnotation ist ein 3 – 6 Tupel, das die Teilbereiche des Wartesystems in verkürzter Form beschreibt. [Vgl. 108 S.366] Die Notation enthält die folgenden Parameter:

$A/B/c/n/v/s$ [113 S.108] [112 S.195]

- A Statistische Verteilung der Ankunftszeiten (Ankunftsprozess)
- B Statistische Verteilung der Bedienzeiten (Abfertigungsprozess)
- c Anzahl der Bedienstationen
- n Kapazität der Warteschlange
- v Populationsgröße des Wartesystems
- s Abfertigungsdisziplin des Wartesystems

Die Verteilung der Ankunftszeit A wird oftmals auch mit dem Ankunftsprozess gleichgesetzt. (Auftrags-)Ankünfte gehen in der Praxis zufällig ein, können aber durch stochastische Gesetzmäßigkeit angenähert werden. Oftmals werden Ankünfte durch

Ankünfte pro Zeiteinheit oder den (mittleren) Zeitabstand zwischen Ankünften beschrieben. [112 S.196, 114 S.2] Eine wichtige Größe der Ankunftsprozesse ist die durchschnittliche Ankunftsrate λ . Diese ist definiert als der Kehrwert des Erwartungswertes E der mittleren Ankunftsabstände. [112 S.198] Der Erwartungswert kann durch verschiedene stochastische Modelle ermittelt werden.

$$\lambda = \frac{1}{E(A)} \quad (2)$$

Die durchschnittliche Bedienzeit bezeichnet die Zeit, in der im Durchschnitt eine Service-Station einen Auftrag bearbeitet. Sie enthält somit nicht die Wartezeit des Auftrags. Die Bedienrate μ ist der Kehrwert des Erwartungswerts der Bedienzeit und gibt an, wie viele Aufträge pro Zeiteinheit bearbeitet werden [112 S. 196-197, Vgl. 115 S.30]:

$$\mu = \frac{1}{E(B)} \quad (3)$$

Der Dienstleistungsbereich kann aus mehreren Service-Stationen bestehen.

Die vierte Größe des Tupels gibt die Kapazität der Warteschlange an, die auch als unbegrenzt mit dem Symbol ∞ beschreiben werden darf. Die Populationsgröße bezeichnet die potenzielle Menge der Aufträge, die nicht in der Warteschlange sind, aber Aufträge in der Warteschlange sein können. Diese kann unter Umständen ebenfalls den Wert ∞ annehmen.

Die letzte Größe ist die Abfertigungsdisziplin, die die Strategie [112 S.85] bezeichnet, nach der Aufträge zur Bearbeitung aus der Warteschlange ausgewählt werden. Hierbei wird in der Regel unterschieden zwischen [112 S.195]:

- First-in-First-Out (FIFO)
- Last-in-First-Out (LIFO)
- Prioritätsbasierte Auswahl (PRIO)
- Selection-in-Random-Order (SIRO)

In der FIFO-Disziplin werden Aufträge chronologisch nach dem Eintritt in der Schlange bedient. Die LIFO-Disziplin bevorzugt Aufträge, die später in die Warteschlange eintreten. Je später ein Kunde der Warteschlange beitrifft, desto eher wird er bedient. Die prioritätsbasierte Auswahl ordnet Aufträgen verschiedene Bedeutungen zu, wobei diese über die Auswahl des Auftrags unterscheiden. Bei der SIRO-Disziplin hat der Eintrittszeitpunkt des Auftrags in die Schlange keine Bedeutung. Aufträge werden nach Belieben ausgewählt. [108 S.366][112 S. 85]

4.3.2 Modellierungsmethoden der Ankunfts- und Abfertigungsrate

Sowohl Auftragsankünfte als auch die Abfertigungen unterliegen gewissen Schwankungen und können daher durch statistische Modelle abgebildet werden [114 S.15]. Die Auswahl des geeigneten Modells hängt von dem Modellierungsproblem sowie von

den gegebenen Daten ab. Im Folgenden werden einige vornehmlich diskrete Wahrscheinlichkeitsverteilungen vorgestellt, die besonders häufig in Wartesituationen angewendet werden.

Bernoulli-Verteilung

Die Bernoulli-Verteilung beschreibt die Wahrscheinlichkeitsverteilung für binäre Zufallsvariablen, die in der Regel als Erfolg oder Misserfolg interpretiert werden [Vgl. 116 S.804].

Intuitives Beispiel: Es wird das Werfen einer Münze herangezogen, bei der Kopf als Erfolg und Wappen als Misserfolg verstanden würde. Dabei müssen die Wahrscheinlichkeiten nicht gleich verteilt sein. Auch das Würfeln kann als Bernoulli-Verteilung modelliert werden, wenn bspw. ein bestimmter Wert (z. B. $x = 6$) als Erfolg und die restlichen Ergebnisse als Misserfolg bewertet werden. [116, Vgl. S.802]

Für den Erwartungswert $E(X)$ gilt:

$$E(X) = 1 + p + 0 * (1 - p) = p$$

Binomialverteilung

Die Binomialverteilung berücksichtigt im Gegensatz zur Bernoulli-Verteilung die Summe von vielen Zufallsvariablen, wobei die Zufallsvariable x die Anzahl der Erfolge angibt. Die Verteilung wird mit $Bin(n, p)$ angegeben, wobei n die Anzahl der hintereinander durchgeführten Experimente und p die Erfolgswahrscheinlichkeit für einen Wurf angibt. [Vgl. 117 S.30]:

$$X = \sum_i^n X_i$$

Intuitives Beispiel: Es wird der Münzwurf herangezogen. Es soll Kopf als Erfolg und Wappen als Misserfolg gewertet werden ($p = 0.5$). Die Binomialverteilung beantwortet bspw. die Frage, wie die Wahrscheinlichkeitsverteilung ist, bei 10 (n) maligem Wurf 7 (k) mal Kopf (als Erfolg) zu zählen.

Für die Wahrscheinlichkeit $P(X = k)$ gilt [Vgl. 118 S.356]:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Geometrische Verteilung

Die Geometrische Verteilung eignet sich zur Ermittlung der Wahrscheinlichkeitsverteilung über die Anzahl der Wiederholung eines Zufallsexperiments, bis zum ersten Mal ein Erfolg eintritt. [Vgl. 117 S.33]

Intuitives Beispiel: Es wird der Münzwurf herangezogen. Es soll Kopf als Erfolg und Wappen als Misserfolg gewertet werden ($p = 0.5$). Die Münze wird so oft (k) geworfen bis zum ersten Mal Kopf als Ergebnis eintritt.

$$P(X = k) = (1 - p)^{k-1} * p$$

Poissonverteilung

Die Poissonverteilung wird für die Wahrscheinlichkeitsverteilung über X interpretiert und als Anzahl von Ereignissen herangezogen, wenn Ereignisse zufällig auftreten, aber innerhalb eines Intervalls ein Durchschnittswert λ angegeben werden kann. Die Zufallsvariable x steht dann für Erfolge innerhalb eines Intervalls, in dem das Ereignis ebenfalls eintritt. Sie eignet sich zur Modellierung von Auftragseinkünften in Wartesystemen. Die (diskrete) Wahrscheinlichkeitsverteilung der Poissonfunktion ist wie folgt definiert:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (4)$$

- λ ist der Erwartungswert
- k bezeichnet die Anzahl der Ereignisse, wobei hierüber die Wahrscheinlichkeitsverteilung gebildet wird
- e ist die eulersche Zahl

Abbildung 25 zeigt eine beispielhafte Poissonverteilung bei der ein Ereignis im Mittel mit $\lambda = 5$ zu benennen ist. Somit sind Werte um $k = 5$ am wahrscheinlichsten, dennoch können Ereignisse innerhalb eines konstant großen Zeitraums sehr viel häufiger oder sehr viel seltener vorkommen. Die Wahrscheinlichkeit solcher Ereignisse ist dementsprechend gering.

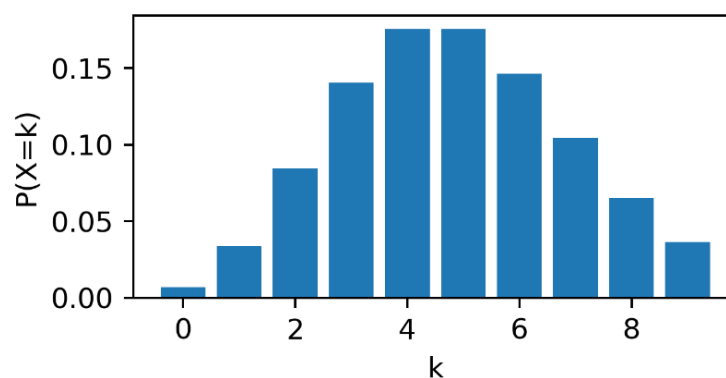


Abbildung 25: Poissonverteilung für eine durchschnittliche Ankunftsrate $\lambda = 5$ [Eigene Darstellung]

Intuitives Beispiel: In einem Restaurant werden im Schnitt λ Gäste pro Stunde erwartet. $P(X = k)$ gibt die Wahrscheinlichkeit an, dass genau k Gäste innerhalb einer Zeiteinheit bspw. einer Stunde eintreffen.

Erlangverteilung

Für die Erlangverteilung wird eine Dichtefunktion angegeben, denn die Verteilung wird für stetige Werte ermittelt. Sie wird über die stetige Zufallsvariable X mit den Parametern k und λ wie folgt berechnet:

$$P(x, k, \lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} \quad (5)$$

Dabei kann k als die Anzahl der Ereignisse und λ als die durchschnittliche Ereigniszahl pro Intervall interpretiert werden. [119 S.310]

Die Funktion gilt für $x \geq 0$. Der Parameter k ist der Formparameter und kann als Beschreibung der abzuwartenden Ereignisse interpretiert werden.

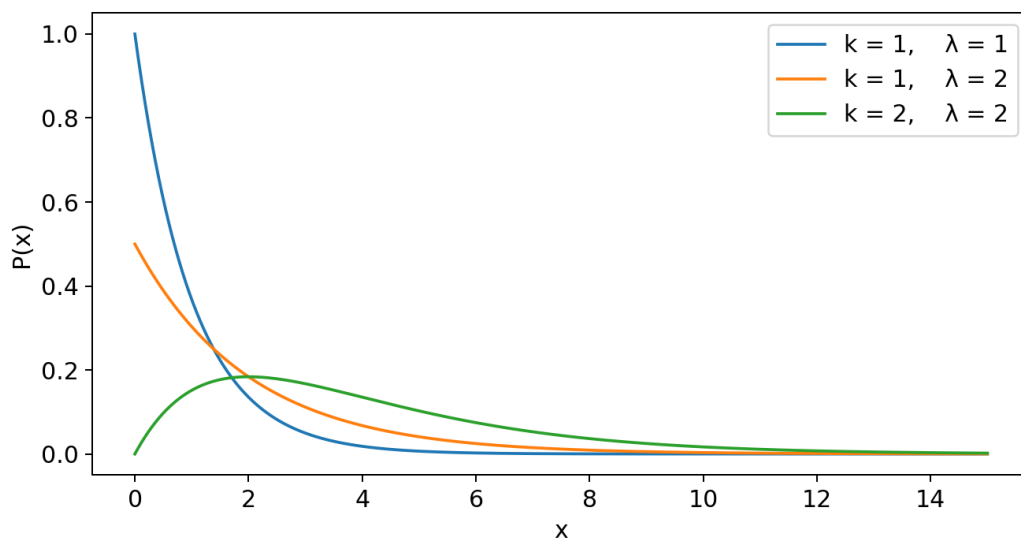


Abbildung 26: Erlangverteilung mit unterschiedlicher abzuwartender Ereignisanzahl [Eigene Darstellung]

Intuitives Beispiel: Treffen im Schnitt λ Ereignisse pro Zeitintervall ein, wird über x die Wahrscheinlichkeit ausgegeben, dass in der Zeitspanne x , n Ereignisse auftreten.

5 Anforderungen an den kognitiv assistierenden Nacharbeitsplatz

Das Ziel dieser Arbeit ist die Entwicklung eines kognitiv assistierenden Arbeitsplatzes für die Montage und Nacharbeit: ein Kognitives Assistenzsystem. Wichtig ist hierbei die Vorgabe, dass es sich bei dem betrachteten Montagebereich um einen Bereich mit einem hohen Umfang an manuellen Tätigkeiten handelt.

5.1 Ablauf in der Nacharbeit

Prinzipiell streben Unternehmen die Null-Fehler-Strategie an. Dennoch können Fehler im Produkt oder Prozess vorkommen. Da eine durchweg fehlerfreie Produktion erfahrungsgemäß unrealistisch ist, ist die Befassung mit fehlerhaften Produkten notwendig. [39 S.317]

Für ein Unternehmen bestehen im Wesentlichen drei Möglichkeiten, mit einem fehlerhaften Produkt umzugehen. [Vgl. 39 S.317]

1. Verkaufen z. B. zu einem günstigeren Preis
2. Entsorgung/Verschrottung des Produktes
3. Nacharbeiten am Produkt zur Wiederherstellung der spezifikationsgetreuen Funktions- und Verkaufsfähigkeit

Jede dieser Möglichkeiten ist mit weiteren Kosten verbunden.

Für die Nacharbeit existieren zwei Möglichkeiten. Zum einen kann die Nacharbeit in der Montagelinie bei unvollständigem Produkt vollzogen werden, bspw. indem in der Montage das unvollständige Produkt einer Zwischenprüfung unterzogen wird und bei einem erkannten Fehler ggf. ausgeschleust und nachbearbeitet wird. Zum anderen kann das Produkt am Bandende bzw. am Ende der Endmontage geprüft und ggf. in der Nacharbeit in einen verkaufsfähigen Zustand versetzt werden. [39 S.317] Die Nacharbeit wird in einem von der Endmontage losgelösten Bereich durchgeführt. Dieser wird als Nacharbeitsbereich bezeichnet. Im Rahmen dieser Arbeit wird der zweite Fall, also die Nacharbeit am endmontierten Produkt betrachtet.

Ein beispielhafter Anwendungsfall ist die Getriebeproduktion. Abbildung 27 zeigt den Grobablauf zwischen Serienmontage und Nacharbeit im Hinblick auf die Funktionsprüfung des Getriebes und die Handhabung des Getriebefehlers. Das Getriebe besteht aus hunderten Bauteilen bzw. dutzenden Baugruppen. Diese werden zum Teil von einem Zulieferer angeliefert, zum Teil in der Vormontage in Baugruppen zusammengebaut. In der sogenannten Endmontage wird aus allen vorgefertigten Bauteilen und vormontierten Baugruppen das Endprodukt montiert. Das fertige Produkt wandert in den Prüfstand, wo es diversen Prüfungen (z. B. Funktionsprüfung, Prüfung auf Leckage, akustische Emission, ungeplante Schwingungen etc.) unterzogen wird. Bei einer in-Ordnung Prüfung (i.O. Prüfung) wird das Getriebe in den Versand geschickt. Im

Falle eines erkannten Fehlers, wird das Getriebe samt Fehlerprotokoll in den Nacharbeitsbereich transportiert. Im Nacharbeitsbereich wird das Getriebe auf einer Bereitstellungsfläche abgestellt, wo es von einem Mitarbeiter der Nacharbeit als Auftrag entgegengenommen wird. In der Regel existiert keine systematisierte Auftragszuweisung. Dem Mitarbeiter obliegt die freie Wahl.

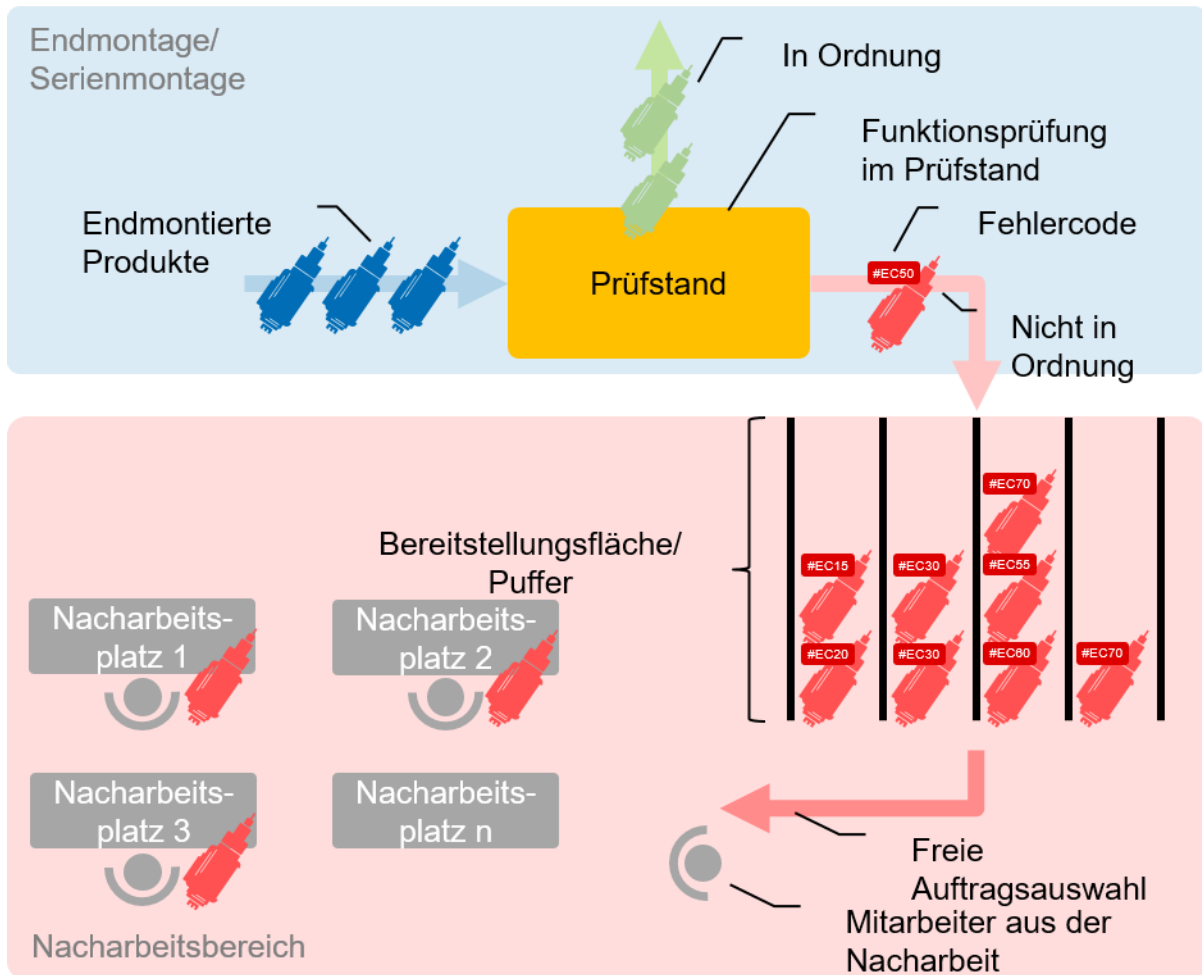


Abbildung 27: Zusammenhang zwischen Serienmontage und Nacharbeitsbereich [Eigene Darstellung]

Der Nacharbeitsbereich besteht aus mehreren Einzelarbeitsplätzen. Diese sind nicht variantengebunden. Der Mitarbeiter schaut sich das Fehlerprotokoll an, auf dem ein Fehlergrund benannt ist. Dieser beinhaltet das Fehlverhalten des Produktes wie zum Beispiel ein Druckverlust oder akustische Emissionen. Das Fehlerprotokoll kann Aufschluss über die mögliche Fehlerursache geben. Basierend auf dieser Information demontiert ein Mitarbeiter das Produkt bis zur Fehlerursache, führt alle Maßnahmen zur Fehlerbehebung durch und remontiert das Produkt. Da die Prozesssequenz von dem vorliegenden Fehlergrund abhängt, unterscheiden sich die Prozesssequenzen je nach Fehlerfall. Aufgrund der hohen Flexibilitätsanforderungen ist in dem Nacharbeitsbereich ein sehr geringer Automatisierungsgrad vorzufinden. Zusätzlich muss der Mitarbeiter einen hohen Erfahrungsgrad nachweisen. Daher kann bei Personalausfall nicht ohne Weiteres eine Ersatzperson einspringen. In vielerlei Hinsicht unterscheidet sich

die Nacharbeit von der (halb-)automatisierten Montage, was sich auch in der Produktivität im Nacharbeitsbereich niederschlägt. Daher werden im nächsten Kapitel die Anforderungen an das Assistenzsystem definiert.

5.2 Anforderungen an das Assistenzsystem im Kontext der Nacharbeit

Die Anforderungen an das Assistenzsystem leiten sich aus realen Problemstellungen bei produzierenden Unternehmen aus dem Maschinenbau ab. Dennoch sind die Anforderungen allgemein in ähnliche Domänen übertragbar. Im Fokus der Entwicklung liegt der Nacharbeitsbereich. Dabei wird die Entscheidung, dass die Nacharbeit am Produkt stattfindet, vorausgesetzt. Folgende Anforderungen werden an das Kognitive Assistenzsystem gestellt:

1. Das entwickelte Digitale Kognitive Assistenzsystem unterstützt den Mitarbeiter bei der Durchführung der Montage- bzw. den Fehlerbehebungsprozessen während der Nacharbeit.
2. Das Assistenzsystem unterstützt bei unterschiedlichen Fehlerfällen.
3. Das Assistenzsystem berücksichtigt die unterschiedlichen Erfahrungsgrade der Mitarbeiter.
4. Der Mitarbeiter hat die Möglichkeit, das Assistenzsystem zu überstimmen.
5. Das Assistenzsystem trifft Entscheidungen bei gegebenem Digital- und Realzustand.

Aufgrund unausgewogener Auftragsverteilung kommt es im System zu Verzögerungen, obwohl eine Auftragsverteilung unter Berücksichtigung der Mitarbeiterfähigkeiten und -befugnisse einen optimierten Durchlauf ermöglicht. Daher ist neben der Assistenz in der Nacharbeit auch der Aspekt geeigneter Auftragsverteilung zu berücksichtigen. Demzufolge wird in einem weiteren Schritt unter Berücksichtigung des Erfahrungsgrades sowie der Berechtigungen der Mitarbeiter die Auftragszuweisung optimiert.

6 Entwicklung des Digitalen Kognitiven Assistenzsystems für die Nacharbeit

Wie in der Einleitung der vorliegenden Arbeit bereits beschrieben, kann die Entwicklung des Assistenzsystems in vier Stufen eingeteilt werden. Jede Entwicklungsstufe erweitert die Einsatzmöglichkeiten des Kognitiven Assistenzsystems. In diesem Kapitel wird in vier Unterkapiteln die Entwicklung des Kognitiven Assistenzsystems beschrieben. Dabei wird sowohl auf den Stand der Technik eingegangen als auch auf die konkrete Umsetzung. Jede Entwicklungsstufe erweitert die vorherigen Entwicklungen punktuell.

6.1 Entwicklung eines herkömmlichen Assistenzsystems

In der ersten Entwicklungsstufe wird die Grundform des Kognitiven Assistenzsystems eingerichtet. In der Grundform unterscheidet sich das Assistenzsystem äußerlich unwesentlich von herkömmlichen Kognitiven Assistenzsystemen. Der Unterschied dieser Entwicklung liegt in der Bereitstellung der Daten und dem zugrunde liegenden Datenmodell. In diesem Kapitel werden der physische Aufbau des Kognitiven Assistenzsystems sowie die Steuerungsarchitektur behandelt.

6.1.1 Theoretische Umsetzung

Das entwickelte Digitale Kognitive Assistenzsystem unterstützt den Mitarbeiter während der Bearbeitung des Nacharbeitsauftrags. Hierzu werden dem Mitarbeiter **Informationen** zum aktuellen Nacharbeitsprozess angezeigt. Sind Betriebsmittel zur Durchführung des Montageprozesses notwendig, werden diese automatisiert freigegeben. Dies setzt voraus, dass das Betriebsmittel eine Datenschnittstelle zur elektronischen **Freigabe** bzw. Blockierung besitzt und ggf. auch situativ **parametriert** werden kann. Ist der Prozess durchgeführt, muss dieser durch die integrierte Sensorik des Assistenzsystems einer **Prüfung** bzw. Validierung unterzogen werden. Anschließend erfolgt die **Dokumentation** des Prüfergebnisses. Diese umfasst neben dem Ergebnis bspw. Produktdaten, Zeitstempel, Montagearbeitsplatz und ggf. die Mitarbeiteridentifikationsnummer. Die theoretische Umsetzung beschreibt die prinzipielle Gestaltung der Datenstruktur des Nacharbeitsplans des Assistenzsystems sowie das Steuerungskonzept. Ferner wird der physische Aufbau des Kognitiven Assistenzsystems beschrieben.

Datenstruktur des Assistenzsystems

Grundlage der Steuerung des Kognitiven Assistenzsystems bildet ein digitaler Nacharbeitsplan. Dieser beschreibt Schritt für Schritt alle Prozesse, die zur Behebung der mutmaßlichen Fehlerursache notwendig sind. Abbildung 28 zeigt den Aufbau des Nacharbeitsplans als Prozesskette. Der Nacharbeitsplan besteht aus einem Montageabschnitt, in dem das Produkt bis zur Zugänglichkeit der Fehlerursache zerlegt

wird, einem Fehlerbehebungsabschnitt sowie einem abschließenden Remontageabschnitt, in dem das Produkt wieder zusammengesetzt wird. Die Darstellung in Abbildung 28 verdeutlicht die Symmetrie des Nacharbeitsplans. Diese resultiert aus der zunächst vereinfachten Annahme, dass Demontageprozesse wieder mit Remontageprozessen rückgängig gemacht werden müssen. Wird ein fehlerfreies Bauteil herausgenommen, so muss es zumeist wieder eingesetzt werden. Sonderoperationen wie das Reinigen außerhalb der Fehlerbehebung werden nicht berücksichtigt.

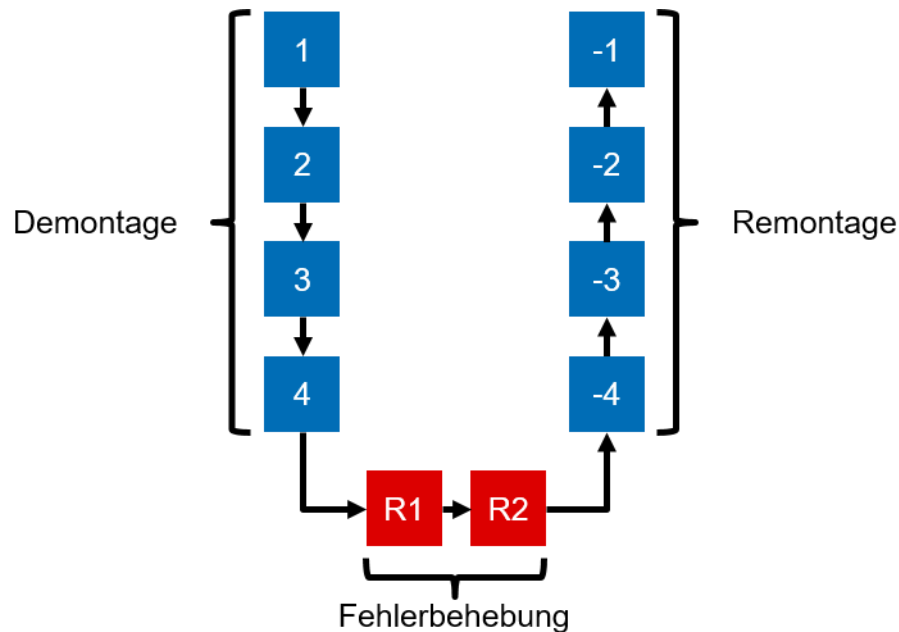


Abbildung 28: Schematische Darstellung eines Nacharbeitsplans mit Demontage-, Fehlerbehebungs- und Remontageprozessen [Eigene Darstellung]

Jedem Prozess wird eine maschinenlesbare Definition zugewiesen. Jede Prozessdefinition ist in vier Datenklassen unterteilt, die die Ziele der Assistenz widerspiegeln. Die Datenklassen lauten:

1. Prozessinformation
2. Prozessausführung
3. Prozessprüfung
4. Prozessdokumentation

Die maschinenlesbaren Daten werden im JSON-Format organisiert. Das JSON-Format hat den Vorteil, dass es weit verbreitet und übersichtlich strukturiert ist. Im Gegensatz zu XML kann die JSON-Notation einfach und überschaubar händisch vom Nutzer eingegeben werden. JSON erlaubt die Organisation der Daten im Name/Key-Value-Format. [120 S.6] Im Unterschied zu Arrays kann damit auf Daten wie in einem Wörterbuch durch Aufruf des Names bzw. des Keys zugegriffen werden. Das Grundgerüst der JSON-Datei ist wie folgt aufgebaut:

```
{
  "Prozessinformation":{
    "Adresse":"",
    "Parameter":{},
  },
  "Prozessausführung":{
    "Adresse":"",
    "Parameter":{}
  },
  "Prozessprüfung":{
    "Adresse":"",
    "Parameter":{}
  },
  "Prozessdokumentation":{
    "Adresse":"",
    "Parameter":{}
  }
}
```

Die Datei verfügt über vier JSON-Objekte, die den vier Datenklassen entsprechen. Jedes dieser JSON-Objekte besteht aus zwei weiteren JSON-Objekten, namentlich „Adresse“ und „Parameter“. „Adresse“ gibt das Betriebsmittel für diesen Nacharbeitsprozess an. Das JSON-Objekt mit dem Key-Value „Parameter“ listet die zugehörigen Parameter auf. Diese sind abhängig von den gewählten Betriebsmitteln. Beispielsweise soll im ersten Prozess überprüft werden, ob das Getriebe überhaupt positioniert ist. Das folgende JSON-Objekt zeigt einen Ausschnitt für die Prozessinformation, wobei die Prozessinformation durch das Mitarbeiterführungssystem realisiert wird. Das Mitarbeiterführungssystem benötigt Informationen zur Prozessbeschreibung in Kurz- und Langform, eine Prozessvisualisierung (Standbild oder Animation) sowie eine Visualisierung des Betriebsmittels.

```

{
  "Prozessinformation":{
    "Adresse":"Mitarbeiterführungssystem",
    "Parameter":{
      "ProcessShortDesc":"Getriebe Grundposition",
      "ProcessLongDesc":"Positionieren Sie das Getriebe unter die Kamera.",
      "ProcessImage":"C:\Users\ProduktDB\Process\Grundposition.gif",
      "RessourceImage":"C:\Users\ProduktDB\Betriebsmittel\Hand.png"
    },
  },
},
...

```

Abbildung 29 zeigt schematisch die Assoziation jeder JSON-Datei mit einem Prozessschritt. Somit sind die Parameter pro Nacharbeitsprozess ermittelbar. Ausnahme bildet das Mitarbeiterführungssystem. Da oftmals aus Gründen der Nutzerfreundlichkeit im Mitarbeiterführungssystem alle Montageprozesse bereits zu Beginn aufgelistet werden, müssen hierfür die JSON-Dateien auch zu Beginn ausgelesen werden. Diese Aufgabe wird allerdings nicht von der Assistenzsystemsteuerung übernommen, sondern vom Mitarbeiterführungssystem selbst.



Abbildung 29: Schema des Nacharbeitsplanes als Prozesskette samt maschinenlesbarer Prozessdefinition [Eigene Darstellung]

Modellierung eines Nacharbeitsplans zur qualifikationsbasierten Assistenz

Der Nacharbeitsplan listet die Nacharbeitsprozesse auf, die für die Wiederherstellung des verkaufsfähigen Zustands des Produktes notwendig sind. Allerdings muss jeder Auftrag einem Mitarbeiter der Nacharbeit zugewiesen werden. Dabei ist jedoch zu beachten, dass nicht jeder Mitarbeiter jeden Auftrag bearbeiten kann oder bearbeiten sollte. Gründe hierfür sind:

- Dem Mitarbeiter fehlen die Berechtigungen, um bestimmte Prozesse durchzuführen. Beispiel hierfür ist das Vermessen von Spiel unter Verwendung von Messmitteln.
- Der Mitarbeiter kann die ihm zur Verfügung gestellten Informationen nur vor dem Hintergrund seiner Erfahrung verstehen. Daher müssen Informationen personengerecht dargestellt werden.
- Die Präsentation und Darstellung der Information irritiert den Mitarbeiter oder löst beim Mitarbeiter Unmut durch Bevormundung aus.

Damit dem Mitarbeiter Informationen zum Montageprozess bedarfsgerecht präsentiert werden können, muss der Bedarf festgestellt werden. Hierzu ist die Kenntnis über zwei Sachverhalte notwendig. Zum einen betrifft dies die Qualifikation bzw. den Erfahrungsgrad des betreffenden Mitarbeiters, zum anderen die mutmaßliche Vertrautheit mit dem Montageprozess. Die Qualifikation des Mitarbeiters kann unabhängig von dessen Ausbildung geschehen. Es ist ratsam, die Qualifikation in Form einer Einordnung bzw. Klassifikation vorzunehmen. Hierdurch können unterschiedliche Qualifikationen vereinheitlicht werden. Die Einordnung der Mitarbeiter in eine Qualifikationsklasse geschieht manuell durch eine vorgesetzte Instanz mit entsprechenden Befugnissen wie beispielsweise einem Vorarbeiter oder einem Meister. Die Mutmaßung über die Vertrautheit mit dem Montageprozess kann auch über eine zeitliche Erfassung der Montageprozesse geschehen. Je länger die Prozessausführung zurückliegt, desto wahrscheinlicher kann ein Fehler im manuellen Prozess auftreten. Zudem kann wiederholtes Informieren über den Prozess für den Mitarbeiter lernförderlich sein.

Auf der anderen Seite müssen Montageprozesse ebenfalls einem Anforderungsniveau zugeordnet werden. Diese werden in schwierig bzw. einfach oder speziell untergliedert. Die Unterscheidung „schwieriger“ oder „einfacher“ Prozess sagt nichts über die Durchführbarkeit des Prozesses aus. Spezielle Prozesse hingegen bedürfen einer Autorisierung, die einmalig bzw. prinzipiell erteilt wird.

Zu Beginn eines Nacharbeitsauftrags muss dem Kognitiven Assistenzsystem der Mitarbeiter bekannt sein. Dies ist über eine Anmeldung an seinem Arbeitsplatz möglich. Durch die Eingabe des Fehlercodes wird der Nacharbeitsplan ermittelt. Da somit die Nacharbeitsprozesse bekannt sind, wird ein Fähigkeit/Befugnisse-Prozessanforderungsabgleich durchgeführt. Hierbei werden die Nacharbeitsanforderungen eingeholt und mit den Fähigkeiten des Mitarbeiters abgeglichen. Nach dem Abgleich werden bestimmte Montageprozesse prominenter dargestellt. Wichtig ist hierbei, dass im Mitarbeiterführungssystem alle Montageprozesse aufgelistet werden. Würden Informationen zum Montageprozess überhaupt nicht zur Verfügung gestellt werden, könnte dies ebenfalls zu Irritationen führen. Der Unterschied liegt allerdings in der Auswahl der Informationsanzeige, so muss bspw. nicht zusätzlich eine Projektion im Blickfeld des erfahrenen Mitarbeiters erzeugt werden.

6.1.2 Praktische Umsetzung der Assistenzfunktion

Die Umsetzung der entwickelten Assistenzfunktionen wird in einer Pilotumgebung vollzogen. Abbildung 30 zeigt den physischen Aufbau des Kognitiven Assistenzsystems.



Abbildung 30: Pilotarbeitsplatz als Assistenzsystem für dynamische Arbeitsumgebungen [Eigene Darstellung]

Das Assistenzsystem besteht aus zwei Arbeitsplätzen, wobei ein Arbeitsplatz mit einem Tisch ausgestattet ist und der andere lediglich den Werkstückträger mit dem eingespANNten Getriebe umfasst. Der Tisch dient zur Nacharbeit an Baugruppen sowie zur Ablage von demontierten Bauteilen. Der Werkstückträger dient der Aufnahme des Getriebes, sodass Nacharbeitsprozesse an diesem durchgeführt werden können. Der Tisch ist ausgestattet mit einem Bildprojektor, einer Waage sowie dem Mitarbeiterführungssystem. Der Werkstückträgerarbeitsplatz ist mit einer Kamera sowie einem Laserprojektor versehen. Weitere Betriebsmittel sind ein Elektronenschrauber sowie ein optisches Positionierungssystem. Die Betriebsmittel werden in den folgenden Abschnitten näher beschrieben.

Betriebsmittel zur Unterstützung der Mitarbeiter während der Nacharbeit

Der Kognitive-Assistenz-Arbeitsplatz ist mit diversen Betriebsmitteln zur Unterstützung des Mitarbeiters ausgestattet. Diese lassen sich den drei Kategorien Information, Ausführung und Überprüfung zuordnen. Für die Prozessdokumentation ist kein erwähnenswertes separates Betriebsmittel erforderlich. Es wird lediglich eine Datenbank zur Abspeicherung der Dokumentation benötigt.

6.1.2.1.1 Betriebsmittel zum Informationsaustausch

Betriebsmittel zum Informationsaustausch dienen der Übermittlung von Daten in beiden Richtungen: Vom Assistenzsystem zum Menschen, aber auch vom Menschen zum Assistenzsystem.

Mitarbeiterführungssystem als zentrale Mensch-Maschinen-Schnittstelle

Das Mitarbeiterführungssystem (oft auch Werkerführungssystem) nimmt in den meisten Digitalen Kognitiven Assistenzsystemen eine Schlüsselrolle ein, da es die zentrale Schnittstelle zur bidirektionalen Kommunikation zwischen Menschen und Maschine bzw. Assistenzsystem darstellt. Zum einen wird über das Mitarbeiterführungssystem dem Mitarbeiter eine Vielzahl an Informationen präsentiert, zum anderen wird dem Mitarbeiter die Möglichkeit eingeräumt, dem System Informationen zuzusenden. Im Rahmen der Arbeit wird ein Mitarbeiterführungssystem entwickelt. Es werden zunächst die Anforderungen definiert:

1. Das Mitarbeiterführungssystem präsentiert dem Nutzer Informationen zum aktuellen Prozess
 - a. Anzeige des kompletten Nacharbeitsplans beginnend bei der Demontage über die Fehlerbehebung bis zu Remontage
 - b. Textuelle Beschreibung der Demontage-, Fehlerbehebungs- sowie der Remontageprozesse
 - c. Visuelle Beschreibung der Demontage-, Fehlerbehebungs- sowie der Remontageprozesse
 - d. Information zu den zu verwendenden Betriebsmitteln
 - e. Visuelle und akustische Erfolgsmeldung nach erfolgter Prozessausführung
2. Das Mitarbeiterführungssystem ermöglicht die Eingabe von Rückmeldung seitens des Nutzers
 - a. Anmeldung des Nutzers
 - b. Eingabe des Fehlercodes des Prüfstands
 - c. Eingabe der Variante sowie der Auftragsnummer
 - d. Manuelle Bestätigung eines durchgeführten Prozesses
 - e. Eingabe einer vom Systemvorschlag abweichenden Fehlerursache
 - f. Hilfefunktion zur erweiterten Unterstützung seitens des Assistenzsystems

Die Ausgestaltung des Mitarbeiterführungssystems als Mensch-Maschinen-Schnittstelle besteht aus zwei Komponenten. Zum einen darf die grafische Gestaltung möglichst wenig die Aufmerksamkeit beeinträchtigen, wobei dennoch die wesentliche Information vermittelt werden soll. Zum anderen müssen benötigte Funktionen wie der Informationsaustausch sichergestellt sein. Diese Arbeit betrachtet nicht explizit die Gestaltung der Oberfläche des Mitarbeiterführungssystems. Da jedoch ein ansprechen-

des und funktionales Oberflächendesign wünschenswert ist, werden bisherige Konzepte aus dem Mensch-Maschinen-Interface-Design übernommen. Vergleicht man die Oberflächengestaltung verschiedener Anbieter von Mitarbeiterführungssystem, so fällt auf, dass der Grundaufbau oft sehr ähnlich ist. Abbildung 31 zeigt ein Beispiel eines (industriellen) Mitarbeiterführungssystems. Oftmals ist eine Prozessliste auf der linken Seite erkennbar, wobei mittig bzw. auf der rechten Seite eine visuelle Darstellung des Prozesses oder des zu erreichenden Produktzustands dargestellt ist. Zudem kommen diverse Bedienelemente und Menüpunkte hinzu.

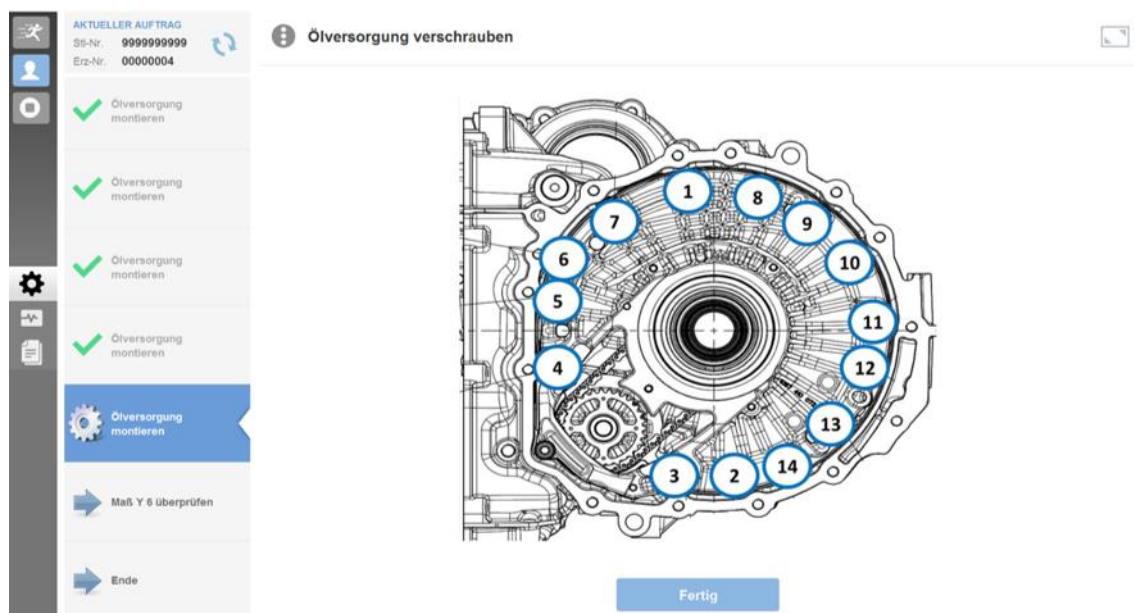


Abbildung 31: Beispiel eines industriell genutzten Mitarbeiterführungssystems [121 S.186]

Im Rahmen dieser Arbeit wird ein Mitarbeiterführungssystem entwickelt, das in der Oberflächengestaltung an den herkömmlichen Mitarbeiterführungssystemen angelehnt ist. Es ist in Abbildung 32 abgebildet. Dieses ist als Webapplikation umgesetzt. Das bietet den Vorteil, weitestgehend betriebssystemunabhängig zu sein und im Browser aufgerufen werden zu können. Das Mitarbeiterführungssystem erlaubt die Eingabe von Fehlercodes. Diese werden an das Assistenzsystem übermittelt, woraufhin auf der linken Seite des Mitarbeiterführungssystems der Nacharbeitsplan erscheint. Dabei wird der aktuell durchzuführende Montageprozess hervorgehoben, wobei die restliche Liste verblasst dargestellt wird. Wird ein Prozess erfolgreich abgeschlossen, so verblasst dieser und ein grüner Haken wird zur Kennzeichnung daneben gesetzt. Um dem Mitarbeiter das visuelle Überprüfen gesetzter Haken zu ersparen, wird ein gesetzter Haken durch ein akustisches Signal begleitet. Im mittleren Bereich der grafischen Oberfläche wird der Prozess visuell dargestellt. Hierzu eignet sich sowohl ein Bild als auch eine Animation im GIF-Format. Auf der rechten Seite werden Abbildungen und Kurzinformation zu den Betriebsmitteln angegeben. Im oberen Bereich ist eine Leiste zu sehen, in der Nutzerinformation sowie Produktinformationen angezeigt werden. Zudem ist ein dünner Fortschrittsbalken zu sehen, der den Fortschritt des Mitarbeiters

anzeigt. Unter der Prozessvisualisierung befindet sich eine ausführliche textuelle Beschreibung des Prozesses bzw. der Prozessdurchführung. Ein weiteres wichtiges Element ist ein Button zum manuellen Quittieren des Prozesses. Hierbei erhält der Nutzer die Möglichkeit, durch das Betätigen des virtuellen Knopfes den Prozess manuell als korrekt durchgeführt zu erklären. Im Falle eines Sensorversagens kann dies den nächsten Prozess freigeben und ein Stocken des Ablaufs verhindern. Zudem wird ein manuell quittierter Prozess mit einem orangefarbenen Haken gekennzeichnet.

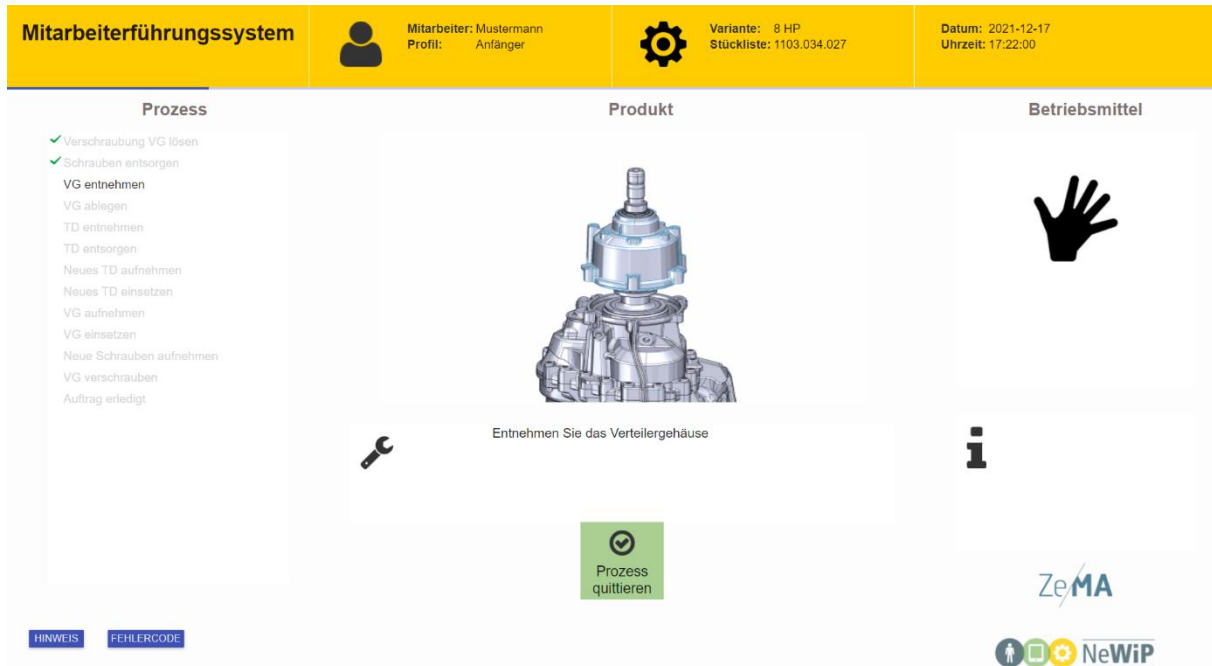


Abbildung 32: Grundansicht des entwickelten Mitarbeiterführungssystems [Eigene Darstellung]

Außerdem werden dem Nutzer weitere Interaktionsmöglichkeiten eingeräumt. Da diese Interaktionen seltener vorkommen, müssen sie über ein Menü aufgerufen werden. Das Menü eröffnet weitere Möglichkeiten. Zum einen hat der Mitarbeiter die Option über das Menü Hilfe anzufordern, zum anderen kann der Mitarbeiter erneut einen Fehlercode angeben, der einen neuen Nacharbeitsplan generiert. Das ist von Vorteil, wenn der Mitarbeiter einer anderen Fehlerursache nachgehen möchte als von System ursprünglich vorgeschlagen.

Zusätzlich bietet das Mitarbeiterführungssystem die Möglichkeit seitens Assistenzsystem Nachrichten in Form von Pop-ups darzustellen. Diese sind vom Mitarbeiter zu bestätigen. Dazu stehen ihm zwei Optionen in Form von Buttons zur Verfügung. Diese können bspw. mit „Ja“ oder „Nein“ belegt sein.

Da das Mitarbeiterführungssystem die Interaktionsschnittstelle von der Assistenzsystemsteuerung zum Mitarbeiter ist, existiert eine API, die das Aufrufen bestimmter Funktionen seitens des Assistenzsystems erlaubt. Abbildung 33 zeigt die Möglichkeiten des Nachrichtenaustauschs zwischen Assistenzsystem und Menschen mit zwischenge-

schaltetem Mitarbeiterführungssystem. Hierzu zählt das Einblenden des initialen Eingabefensters zur Erfassung des Mitarbeiters, den Mitarbeiterdaten sowie des Fehlercodes und das Setzen eines grünen bzw. orangefarbenen Hakens. Die eingegebenen Daten werden über eine TCP-Verbindung an das Assistenzsystem übergeben. Dieses ermittelt den zugehörigen Nacharbeitsplan und übermittelt diesen an das Mitarbeiterführungssystem. Wird ein Prozess vom Mitarbeiter spezifikationsgemäß durchgeführt und so vom Assistenzsystem erkannt, sendet das Assistenzsystem eine Nachricht zum Setzen eines grünen Hakens. Quittiert der Mitarbeiter einen Prozess manuell, wird eine Nachricht an das Assistenzsystem gesendet, woraufhin das Assistenzsystem auch hierfür eine Nachricht zum Setzen eines orangefarbenen Hakens übermittelt.

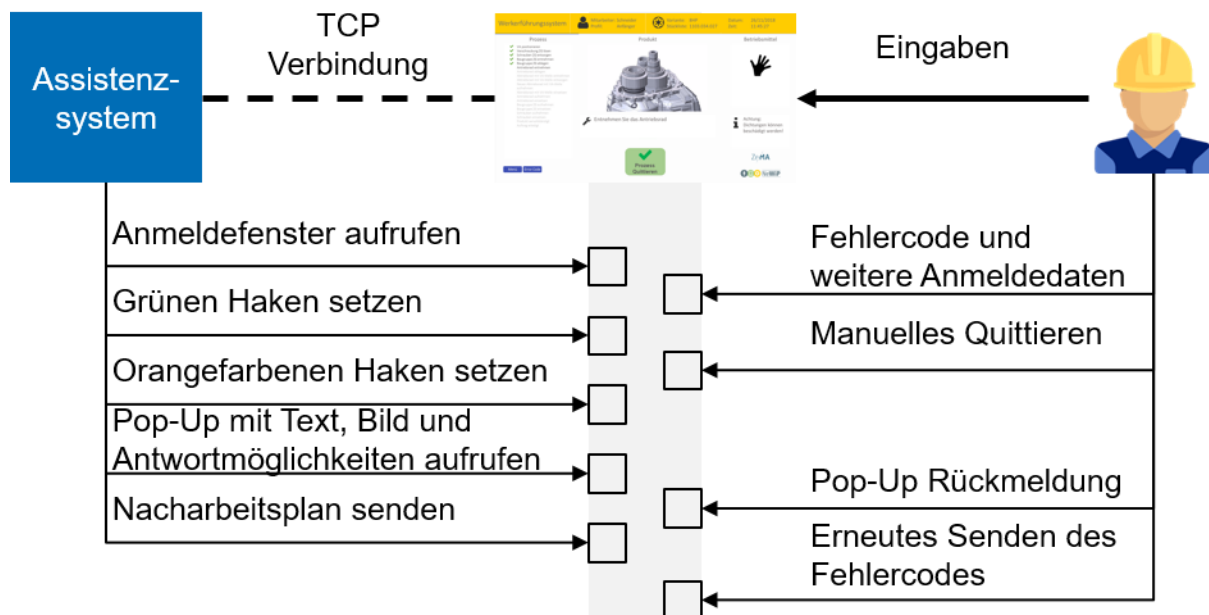


Abbildung 33: Nachrichtenaustausch zwischen Mitarbeiter und Assistenzsystem durch das Mitarbeiterführungssystem [Eigene Darstellung]

Das Mitarbeiterassistenzsystem ist ein wichtiges, aber nicht das einzige Mittel zum Informationsaustausch zwischen Mensch und Assistenzsystem. Zudem ist zu beachten, dass das Mitarbeiterassistenzsystem nicht stetig vom Mitarbeiter in Anspruch genommen werden muss. Der Mitarbeiter ist frei, die Information aufzunehmen oder in seinem Prozess fortzufahren, ohne seinen Blick auf den Monitor (und somit auf das Mitarbeiterführungssystem) zu richten.

Virtueller Taster

Im Mitarbeiterführungssystem wurde eine Quittiertaste vorgestellt, die dem Mitarbeiter das Voranschreiten im Prozess bei Versagen der Zustands- bzw. Prozesserkennung ermöglicht. Virtuelle Taster stellen eine Methode dar, ohne Kontakt zu einer elektronischen Hardware, Signale an eine Steuerung zu übermitteln. Oftmals wird dies mittels Kamera und Bilderkennung realisiert. Da im Validierungsszenario der Mitarbeiter einen Handschuh mit Positionssender trägt (siehe Kapitel 6.1.2.1.3), kann dieser in Kombination mit einem virtuellen und vordefinierten Raum als Signalgeber genutzt werden.

Das Betätigungsfeld kann frei am Arbeitsplatz gewählt werden, wobei die freie Sichtachse zur Kamera für die optische Positionserfassung gewährleistet sein muss.

Laserprojektor zur visuellen und ortsgebundenen Kennzeichnung der Bauteile oder Baugruppen

Ein Laserprojektor ist ein Lichtprojektor, der einen Laserstrahl auf eine Oberfläche projiziert. Der verwendete Projektor bewegt einen Laserpunkt mittels Spiegel in hoher Geschwindigkeit so, dass sich für den Menschen die Illusion einer Geometrie ergibt. Die Projektion schafft auch auf metallischen Oberflächen hohe Kontraste über größere Distanzen. Allerdings ist die Länge der Kontur beschränkt. Je länger die Kontur wird, desto weiter ist der Weg, den der Punkt zurücklegen muss. Hierdurch erscheint die Kontur in abgeblasster Form.

Bildprojektor (Beamer)

Des Weiteren kommt ein Bildprojektor zum Einsatz, der flächig Bilder in unterschiedlichen Farben auf eine Oberfläche projiziert. Der Vorteil gegenüber dem Laserprojektor liegt darin, dass mit dem Bildprojektor größere Flächen mit unterschiedlichen Farben abgedeckt werden. Zudem ist die Installation des Beamers meist mit keinem zusätzlichen softwaretechnischen Aufwand verbunden, da dieser von einem PC wie ein Monitor betrachtet wird. Abbildung 34 zeigt den schematischen Aufbau des Arbeitsplatzes mit einem über dem Arbeitstisch installierten Beamer. Zusätzlich sind in Abbildung 34 drei Koordinatensysteme abgebildet, die der Erläuterung zur Ermittlung der Transformationsmatrix vom Tisch-KS zum BPix-KS dienen.

Damit die Transformation vom Tischkoordinatensystem (Tisch-KS) auf das Pixelkoordinatensystem (BPixel-KS) gelingt, muss der Beamer zuerst eingemessen werden. Hierzu wird ein spezielles Einmessprogramm verwendet, das durch die Definition von 4 Punkt-Paaren auf der Projektionsfläche $^{Tisch}\mathbf{P} = ({}^{Tisch}P_1, {}^{Tisch}P_2, {}^{Tisch}P_3, {}^{Tisch}P_4)$ sowie im Pixelbereich $^{BPix}\mathbf{P} = ({}^{BPix}P_1, {}^{BPix}P_2, {}^{BPix}P_3, {}^{BPix}P_4)$ die Transformationsmatrix $^{BPix}\mathbf{T}_{Tisch}$ berechnet. Im Folgenden wird die $^{BPix}\mathbf{T}_{Tisch}$ hergeleitet.

$${}^{BPix}\mathbf{P} * {}^{Tisch}\mathbf{P}^{-1} = {}^{BPix}\mathbf{T}_{Tisch} \quad (6)$$

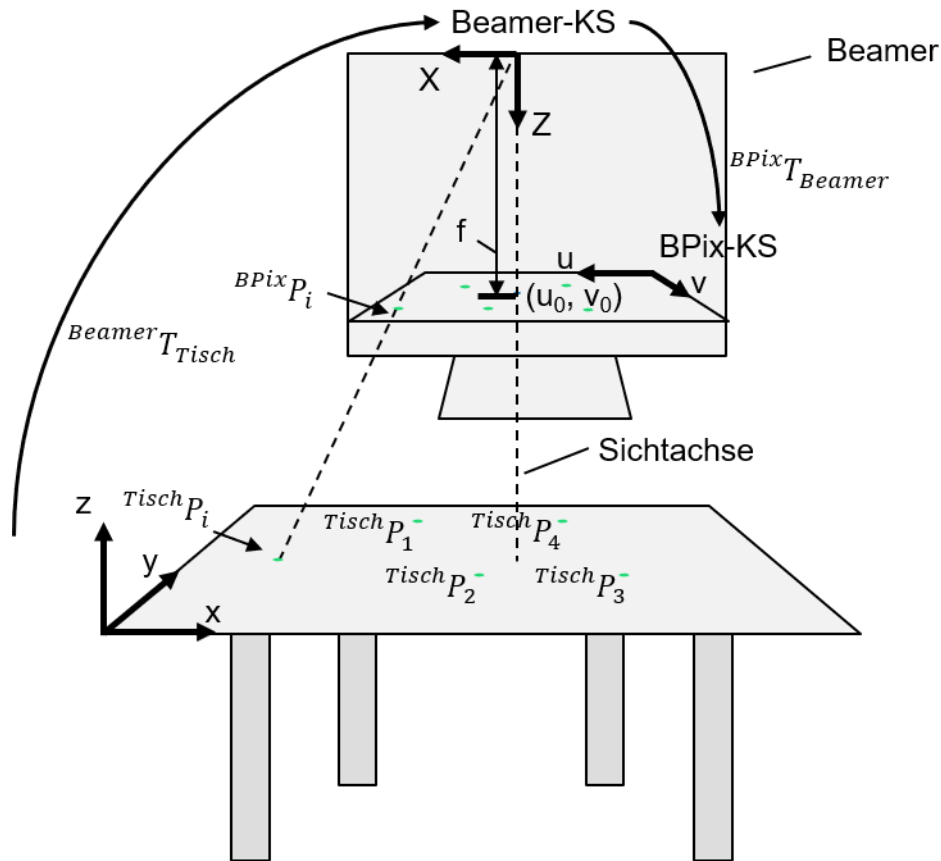


Abbildung 34: Transformation von Tischkoordinaten auf Beamerkoordinaten [Eigene Darstellung]

Grundlage für die Ableitung der Punkte aus einem Tischkoordinatensystem auf das Pixelkoordinatensystem bildet das Modell einer Lochkamera, wie in [122] beschrieben. Es wird ein bekannter Punkt $^{Tisch}P_i$ bezüglich eines Tischkoordinatensystems in einen Punkt $^{BPix}P_i$ bezüglich des Pixelkoordinatensystems des Beamers transformiert. Dabei ist eine Transformation vom Tisch zum Beamer $^{Beamer}T_{Tisch}$ und vom Beamer zum Pixelkoordinatensystem $^{BPix}T_{Beamer}$ durchzuführen, wie Gleichung (7) zeigt.

$$^{BPix}P = \frac{1}{^{Beamer}Z_{Tisch}} \cdot ^{BPix}T_{Beamer} \cdot ^{Beamer}T_{Tisch} \cdot ^{Tisch}P \quad (7)$$

Wobei die Transformationsmatrix $^{Beamer}T_{Tisch}$ aus den obersten drei Zeilenvektoren der homogenen 4×4 Transformationsmatrix $^{Beamer}T_{Tisch}$ (siehe Gleichung (8)) besteht und somit die Dimension 3×4 besitzt.

$$^{Beamer}T_{Tisch} = \begin{pmatrix} ^{Beamer}D_{Tisch} & ^{Beamer}r_{Tisch,Beamer} \\ \underline{0}^T & 1 \end{pmatrix} \quad (8)$$

$$^{Beamer}Tr_{Tisch} = \left(^{Beamer}D_{Tisch} \quad ^{Beamer}r_{Tisch,Beamer} \right) \quad (9)$$

Gleichung (10) zeigt die Faktoren der Transformationsmatrix ${}^{BPix}T_{Beamer}$, die die intrinsischen Parameter des Beamers beinhalten. Neben den Parametern s_x und s_y , die die Einheit Pixel/mm besitzen, beinhaltet die Transformationsmatrix ${}^{BPix}T_{Beamer}$ den Faktor f , der die „Brennweite“ angibt. Die Werte u_0 und v_0 geben den Punkt an, durch den die optische Achse in der Bildebene verläuft.[122]

$${}^{BPix}T_{Beamer} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

Die Ermittlung der Transformationsmatrix ${}^{BPix}T_{Tisch}$ geschieht durch die 4-Punktmethode. Dabei werden 4 Punktepaare gebildet, wobei die Koordinaten der Punkte sowohl im BPixel-KS als auch im Tisch-KS bekannt sind. Gleichung (11) zeigt die Berechnung der Transformationsmatrix. Zu beachten ist auch, dass sowohl die intrinsischen Parameter des Beamers als auch die extrinsischen Parameter, die nur von der Position und Orientierung des Tisches abhängen in der Transformationsmatrix ${}^{BPix}T_{Tisch}$ enthalten sind, wie Gleichung (11) verdeutlicht.

$${}^{BPix}T_{Tisch} = \frac{1}{{}^{Beamer}Z_{Tisch}} {}^{BPix}T_{Beamer} {}^{Beamer}T_{rTisch} \quad (11)$$

Im Rahmen dieser Arbeit wurde ein simplifiziertes Projektionsmodell ausgewählt. Die Methode ermöglicht eine schnelle und einfache Inbetriebnahme des Projektors, erfordert aber, dass die optische Achse senkrecht auf die Projektionsfläche auftrifft und somit konstant ist. Der Faktor $\frac{1}{{}^{Beamer}Z_{Tisch}}$ macht dies deutlich. Dadurch können Abweichungen aufgrund der manuellen Installation entstehen. Trotzdem genügt diese Betrachtung für die Anwendung im Assistenzsystem.

6.1.2.1.2 Betriebsmittel zur Prozessausführung

Die Nacharbeit setzt auf die Flexibilität der Mitarbeiter. Daher sind viele Prozesse zur manuellen Ausführung bestimmt. Allerdings kann die Einführung halbautomatisierter Systeme sinnvoll sein, wenn diese häufig vorkommende Prozesse betrifft. Im Folgenden werden Betriebsmittel des Kognitiven Assistenzsystems zur Ausführung von Prozessen vorgestellt.

Elektronikschrauber

Schraubprozesse sind eine häufige Form zur Verbindung zweier Bauteile. Da der Prozess darüber hinaus kaum variiert, aber recht zeitintensiv sein kann, kann eine Teilautomatisierung des Prozesses lohnenswert sein. Darüber hinaus können die Prozessparameter bei Automatikschraubern eingestellt werden. Hierzu zählen der Drehwinkel sowie das Endmoment.

6.1.2.1.3 Betriebsmittel zur Prozessprüfung

Ein wesentlicher Bestandteil der Mitarbeiterassistenz durch das Kognitive Assistenzsystem ist die Überprüfung der Prozessmerkmale. Im folgenden Kapitel werden die Betriebsmittel, denen die Aufgabe der Prozessprüfung zufällt, beschrieben.

2D- und 3D-Kamera

Da die meisten Nacharbeitsprozesse manuell durchgeführt werden, ist eine 2D- bzw. 3D-Kamera ebenfalls Bestandteil des Kognitiven Assistenzsystems. Der Zweck besteht im bildlichen Abgleich zwischen Soll-Zustand und Ist-Zustand. Der Soll-Zustand wird dabei mittels eines Referenzbildes definiert, indem zuvor bestimmte Merkmale des erfolgreichen Nacharbeitsprozesses bestimmt wurden. Beim Einlegen eines Zahnrades ist bspw. die Verzahnung sowie der Durchmesser des Zahnrades zu überprüfen. Werden im Realprozess unter ständiger Bildaufnahme (Filmen) des Nacharbeitsprozesses die Merkmale erkannt, wird dies als erfolgreiche Prozessausführung erkannt. Bei der Erstellung der Referenzbilder ist auf eine möglichst zustandsspezifische Auswahl von Bildmerkmalen zu achten.

Elektronikschrauber

Elektronikschrauber sind ein klassisches Betriebsmittel zur Durchführung von Schraubprozessen. Die integrierte Sensorik und Elektronik kann das aufgebrachte Drehmoment sowie den Drehwinkel erfassen und mit einer Soll-Vorgabe abgleichen. Die Übermittlung des Prüfergebnisses nach der Prüfung ist eine notwendige Anforderung, da diese die Beobachtung des Assistenzsystems darstellt. Zusammen mit dem System zur Positionserfassung, das weiter unten vorgestellt wird, erlaubt der Schrauber ebenfalls die Ermittlung seiner Position. Insbesondere bei der vorsorglichen Vermeidung von Fehlverschrauben in Form von Sequenzabweichungen bei Schraubenzug erweist sich dies als nützlich.

Waage

In der Nacharbeit muss die Entsorgung bestimmter Bauteile sichergestellt werden. Hierzu kann eine Waage verwendet werden. Mit Hilfe der Waage kann bis zu einem gewissen Grad sichergestellt werden, dass ein Bauteil nicht mehr im Produkt verbaut wurde. Hierzu wird das Bauteil auf der Waage abgelegt. Zudem wird das Gewicht des Bauteils erfasst. Hierzu muss das Gewicht des Bauteils bekannt sein, damit die Detektion fremder Bauteile vermieden wird. Weiterhin muss nach jeder Ablage die Toleranz durch Schmutzverschleppung berücksichtigt werden. Daher wird nicht das Gesamtgewicht erfasst, sondern lediglich das hinzukommende Gewicht. Dennoch erfasst die Waage die Entnahme von Bauteilen und stuft diese als qualitätsgefährdend ein, da die Möglichkeit einer Wiederverwertung besteht. Auch von der Waage muss eine Rückmeldung darüber erfolgen, ob das anvisierte Gewicht erzielt wurde oder nicht. Da die Waage im Gegensatz zum Elektronikschrauber über keine dezentrale Intelligenz verfügt, muss diese erst entwickelt werden. Das entsprechende Softwaremodul wurde in

Python realisiert und kann die Plandaten abgreifen und mit den Ist-Daten abgleichen. Basierend auf der Differenz wird eine Rückmeldung an das Assistenzsystem erstattet.



Abbildung 35: Waage zur Kontrolle der Bauteilentsorgung [Eigenes Bild]

Kommissionierungs-Box

Im Rahmen der Arbeit ist ebenfalls eine Kommissionier-Box entstanden. Diese kann entweder zur Absicherung der Entnahme oder zur Absicherung der Entsorgung genutzt werden.

Die Kommissionier-Box (siehe Abbildung 36) ist elektronisch aufgerüstet. Sie verfügt über eine Klappe mit Schließmechanismus. Dieser besteht aus einem Magneten und wird elektronisch bedient. Die Klappe öffnet sich automatisiert, muss aber händisch geschlossen werden, um Quetschungen zu vermeiden. Die Box kann verschiedene Gegenstände wie Schrauben und Dichtringe aufnehmen. Zudem ist die Box mit einer Waage ausgestattet. Die Waage dient zur Erfassung der Bauteile über die Masse der Bauteile. Als Box zur Entsorgungsabsicherung öffnet diese, sobald Bauteile zu entsorgen sind und eine Zielmasse nicht erreicht wird. Die Box lässt sich in dem Zeitraum nicht händisch schließen, da der Magnet elektronisch temporär entmagnetisiert wird. Auf der anderen Seite ertönt ein Warnton, wenn die Klappe ohne Anlass händisch geöffnet wird. Über einen eingebauten Mikrocomputer ist sie intelligent aufgerüstet und kann mit dem Assistenzsystem elektronisch verbunden werden.

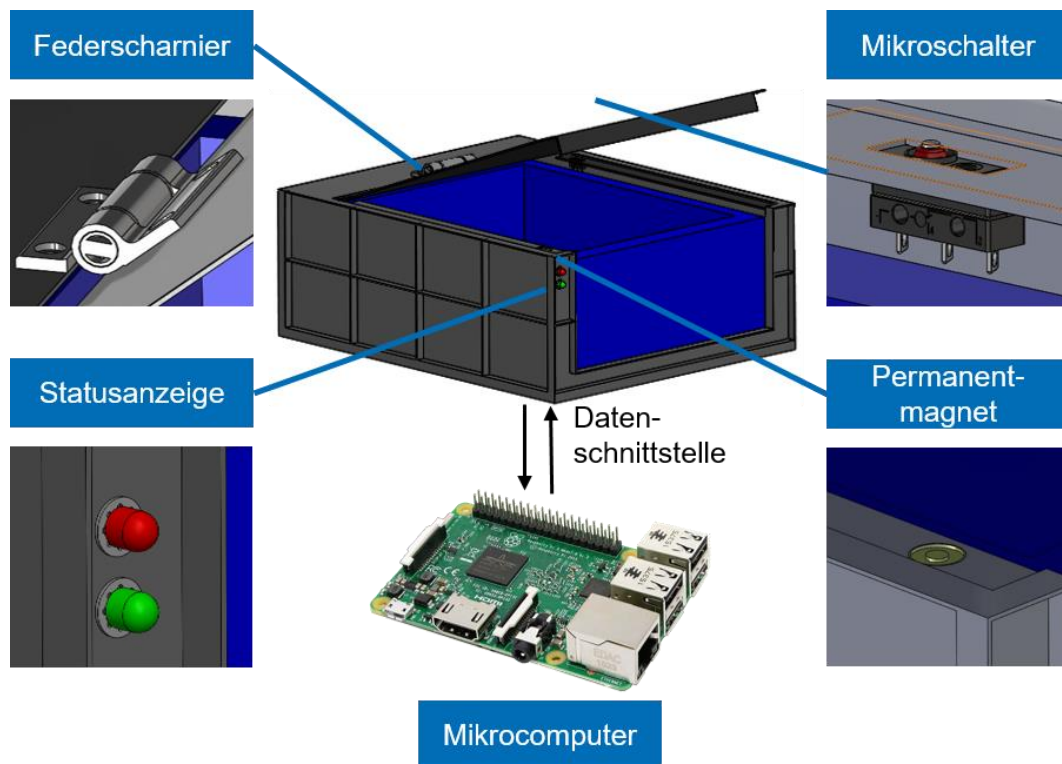


Abbildung 36: Intelligente Kommissionier-Box für Absicherung von Entnahme- oder Entsorgungsprozessen [Eigene Darstellung]

Positionserfassung

Zur Positionserfassung werden optische Systeme verwendet. Mittels Infrarotsendern zusammen mit einer Kamera errechnet ein marktreifes Produkt die Position des Senders. Allerdings muss der Arbeitsplatz vorher eingemessen werden.

Der Positionssender kann für verschiedene Zwecke verwendet werden. Zum einen wird er auf einen Schutzhandschuh positioniert, wodurch indirekt die Hand des Mitarbeiters erfasst werden kann. Hierdurch werden Eingriffskontrollen ermöglicht.

Des Weiteren können die Positionssender an den bereits beschriebenen Elektronikschrauben angebracht werden, wodurch Schraubprozesse auch hinsichtlich Einhaltung der Schraubreihenfolge überprüft werden können. So blockiert der Schrauber, wenn die Schraubreihenfolge nicht eingehalten wird. Dies ist erforderlich, da durch das Ignorieren der Schraubreihenfolge unter Umständen Qualitätseinbußen hingenommen werden müssen.

Des Weiteren werden durch das System virtuelle Räume kalibriert. Die Räume werden mit Hilfe eines Positionssenders markiert und eingelernt. Später können die Sender entfernt werden. Allerdings dürfen weder Infrarotkamera noch die mit den virtuellen Räumen behafteten Oberflächen relativ zueinander verschoben werden. Ein Vorteil dieser Methode ist die Überprüfung des Eingriffs (Eingriffskontrolle), wenn ein Mitarbeiter mit einem mittels Positionssender ausgerüsteten Handschuh in den virtuellen Raum eingreift.

Dieses Positionserfassungssystem eignet sich nicht für mobile virtuelle Räume. Diese müssen dauerhaft mit einem Positionssender ausgestattet sein. Ein virtueller Raum kann bezugnehmend auf einen Positionssender eingemessen sein. Dies ermöglicht Eingriffskontrollen auch an Systemen, die relativ zur Kamera beweglich sind.

Erstellung des Nacharbeitsplans

Der Nacharbeitsplan ist eine Prozessliste, die abgearbeitet werden muss. Die Liste ist in einer Steuerung hinterlegt, die in *Python* entwickelt wurde. Die dazugehörigen Prozessdaten werden in einer JSON-Datei beschrieben. Diese wird in Form eines Dokumentes in der Graphendatenbank ArangoDB angelegt. Die Graphendatenbank besteht aus mehreren Dokumenten, die die Nacharbeitsprozesse beschreiben. Ein Dokument beschreibt sowohl den Demontageschritt als auch den zugehörigen Remontageschritt. Das manuelle Anlegen des Graphen in der Datenbank ist nur in Richtung der Demontage erforderlich. Gegenläufige Prozesse, die für die Remontage notwendig sind, sind zwar ebenfalls zu definieren, ihre Sequenz erschließt sich allerdings automatisch über die Richtungsumkehr. Fehlerbehebungsprozesse werden gleich angelegt. Abbildung 37 zeigt die paarweise Generierung (Demontage, Remontage) der JSON-Datei in der Graphendatenbank ArangoDB.

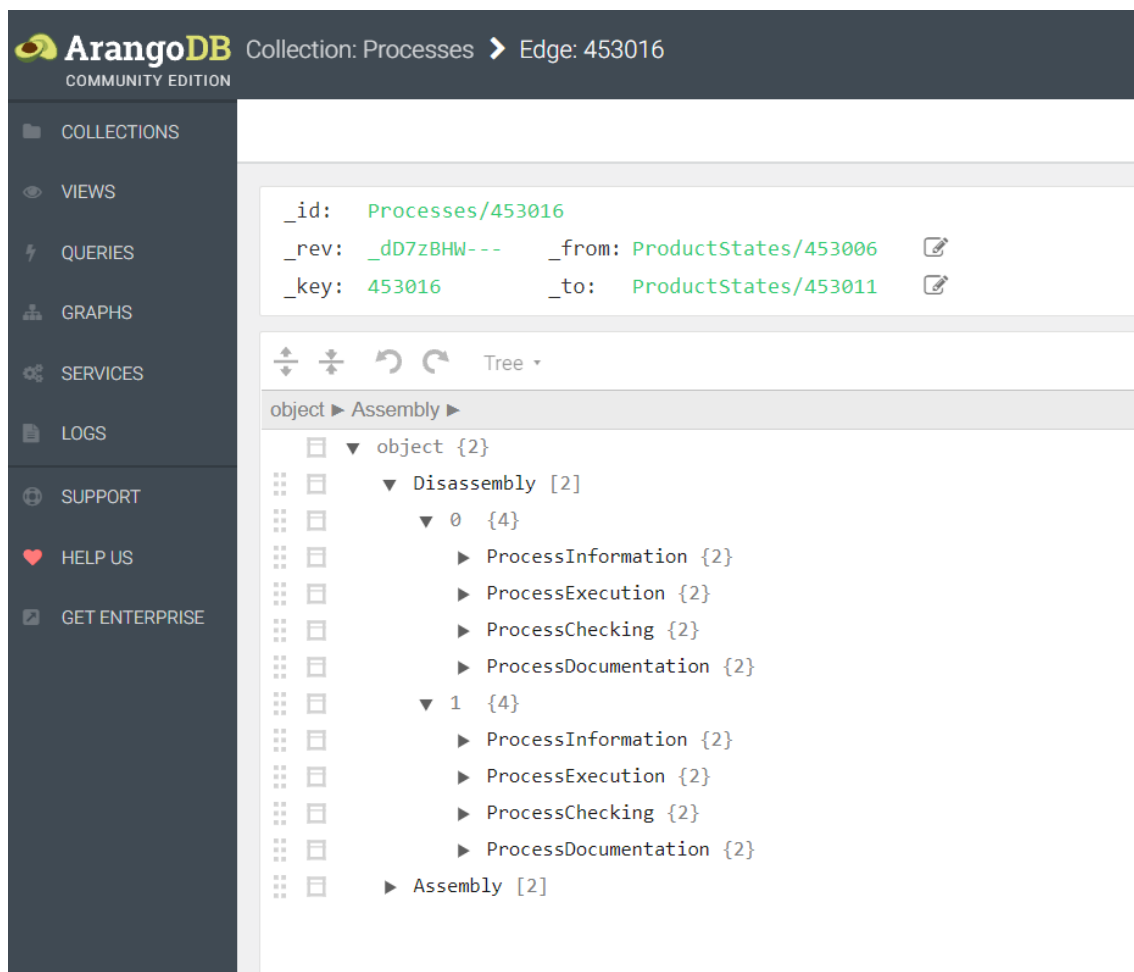


Abbildung 37: Paarweise Generierung der JSON-Datei [Eigene Darstellung]

In der Struktur ist die Unterteilung der Prozessdaten in die vier zuvor erwähnten Datenklassen Prozessinformation, Prozessausführung, Prozessprüfung und Prozessdokumentation ersichtlich.

Ausgabe des Nacharbeitsplans unter Berücksichtigung der Mitarbeiterqualifikation

Prozesse werden den Mitarbeitern gemäß ihrer Qualifikation angezeigt. Der Begriff „Qualifikation“ wird hierbei sehr diffus genutzt, da für die Einstufung der Qualifikation keine amtlichen Nachweise erforderlich sind. Die Einstufung liegt im Ermessen eines Vorgesetzten und wird ebenfalls in der Graphen- und Dokumentendatenbank vorgenommen. Dabei wird der Mitarbeiter in eine der drei Klassen „Anfänger“, „Fortgeschrittener“ und „Experte“ eingeordnet.

Für den Abgleich zwischen Mitarbeiter und Prozess ist eine Mitarbeiterdatenbank und eine Prozesseinstufung erforderlich. Die Mitarbeiterdatenbank wird in ArangoDB als weitere Collection angelegt. Diese enthält alle berücksichtigten Mitarbeiter als separates Dokument, das die Attribute „MitarbeiterID“, „Mitarbeitername“ und „Qualifikationslevel“ enthält, wie in Abbildung 38 zu sehen ist. Dabei werden die Ziffern 1, 2, 3 genutzt, um die 3 Qualifikationslevel zu beschreiben. Alternativ ist auch die Bewertung der Prozesse möglich. Hierbei wird für jede ProzessID die Ausführungszeit (in Sekunden) angegeben. Die letzte Ausführung sowie die Berechtigung („1“ berechtigt, „0“ unberechtigt) notiert.



```

1 {
2   "Name": "Mustermann",
3   "Level": 1,
4   "Prozesse": {
5     "[00000000], ['00000000', '10000000']": [
6       "00:00:04",
7       "2021-12-21",
8       "1"
9     ],
10    "[00000000],[00000000, 10000000]": [
11      "00:00:46",
12      "2021-12-21",
13      "1"
14    ],
15    "[00000000], [00000000, 10100000]": [
16      "00:00:07",
17      "2021-12-21",
18      "1"
19    ],
20    "[00000000],[10000000, 10000000]": [
21      "00:00:08",
22      "2021-12-21",
23      "1"

```

Abbildung 38: Mitarbeiterqualifikation in der Datenbank [Eigene Darstellung]

Dem gegenüber werden Nacharbeitsprozesse ebenfalls in drei Kategorien eingestuft. Bei der qualifikationsgerechten Assistenz gilt, dass Prozesse, die unterhalb des Qualifikationsniveaus liegen, nicht informatorisch präsentiert werden. Einzige Ausnahme bildet das Mitarbeiterführungssystem, das alle Nacharbeitsprozesse auflistet. Zudem werden alle Nacharbeitsprozesse nach wie vor abgeprüft und dokumentiert.

Diskussion

In der vorgestellten Entwicklungsstufe greift das Kognitive Assistenzsystem auf einen digitalen Nacharbeitsplan zurück. Die Prozessdaten werden in einer JSON-Datei hinterlegt. Der Nacharbeitsplan ist prozessweise mit JSON-Dateien verknüpft. Basierend auf den Plandaten werden vorbestimmte Betriebsmittel angesteuert. Zudem werden digitale Nacharbeitspläne durch einen Anforderungs-Fähigkeitsabgleich qualifikationsgerecht erstellt. In dieser Entwicklungsstufe hat das Assistenzsystem einen Nacharbeitsplan für einen Fehlerfall.

In der Nacharbeit sind allerdings mehrere fehlerabhängige Nacharbeitspläne erforderlich. Alle eventuellen Fehlerfälle über separate Nacharbeitspläne anzulegen ist sehr aufwendig. Daher wird im nächsten Kapitel eine Methode zur fallabhängigen Generierung von Nacharbeitsplänen beschrieben.

6.2 Weiterentwicklung des Kognitiven Assistenzsystems für dynamische Produktionsbereiche

Wie in der Einleitung bereits angekündigt, wird der dynamische Produktionsbereich am Beispiel des Nacharbeitsbereichs abgebildet. Die Dynamik im Nacharbeitsbereich kommt aufgrund der Vielzahl abwechslungsreicher Nacharbeitsaufträge zustande. Der Mitarbeiter sieht sich mit fehlerabhängigen Aufträgen konfrontiert. Herkömmliche Assistenzsysteme eignen sich kaum für die Verwendung im Nacharbeitsbereich, da ein statischer Nacharbeitsplan lediglich einen Auftrag abbildet. In dieser Form würde das für die Nacharbeit bedeuten, dass für jede Fehlermöglichkeit ein separater Nacharbeitsplan angelegt werden müsste. Aufgrund des damit verbundenen Pflegeaufwands wird daher auf eine Graphenstruktur zurückgegriffen. Dieser Graph ermöglicht unter Verwendung eines Pathfinding-Algorithmus die dynamische Erstellung des Nacharbeitsplans allein mit Hilfe des vorgegebenen Fehlercodes.

6.2.1 Stand der Technik

In diesem Kapitel werden einige erweiterte Methoden aus der Graphentheorie vorgestellt. Zum einen wird auf die Graphtraversierung eingegangen, die zur situativen Erstellung von Nacharbeitsplänen verwendet werden kann. Zum anderen werden ausgewählte Verfahren zur Beschreibung und Generierung von Verbaureihenfolgen eines Produktes vorgestellt.

Graphtraversierung und Pfade

Mithilfe von Graphen können bestimmte Modelle wie bspw. sequenzielle Systeme abgebildet werden. Diese Modelle dienen der Beschreibung und Analyse bestimmter Eigenschaften bzw. Möglichkeiten innerhalb eines Graphenmodells.

Zur Bestimmung von Wegen (Pfad) zwischen zwei Knoten werden Pathfinding-Algorithmen auf Graphen angewandt. Oftmals ist das Ziel die Ermittlung des kürzesten (günstigsten) Pfades.

6.2.1.1.1 Breitensuche

Die Breitensuche bezeichnet eine Methode zum Durchsuchen von Graphen. Dabei wird ein Startknoten ausgewählt, der die erste Ebene bildet. Alle Nachbarknoten werden in der zweiten Ebene gelistet. Dabei wird wieder bei einem der bereits gelisteten Knoten begonnen, wobei auch dessen Nachbarn notiert werden. Auf diese Weise ergibt sich eine Baumstruktur. Ist ein (Nachbar-)Knoten bereits in einer höheren Ebene gelistet, wird dieser nicht erneut berücksichtigt. [123 S. 225-226] [124 S. 99]

6.2.1.1.2 Tiefensuche

Bei der Tiefensuche wird ausgehend vom Startknoten der erste Nachbar n_0 besucht. Dort angelangt, wird der erste Nachbar n_1 des Knotens n_0 aufgesucht, wobei dieser Vorgang so lange wiederholt wird, bis ein Endknoten erreicht wird. Ist der letzte Knoten bzw. ein Knoten erreicht, der keine weiteren Nachbarn hat, wird der bisherige Pfad

rückwärts beschritten, bis der letzte Knoten mit mehreren Nachbarn erreicht wird, die noch nicht gelistet sind. Hier beginnt der Algorithmus von vorne. [123 S. 229-230]

6.2.1.1.3 Dijkstra Pathfinding-Algorithmus

Edsger W. Dijkstra beschrieb bereits 1959 einen Algorithmus zur Generierung kürzester Wege zwischen zwei beliebigen Knoten in einem Graphen. Im Folgenden wird der Algorithmus in der Art vorgestellt, wie er in der ursprünglichen Veröffentlichung „A note on two problems in connexion with graphs“ beschrieben wurde. Der Algorithmus unterscheidet die Menge an Knoten eines Graphen sowie die zugehörigen Kanten. Zusätzlich werden die Mengen M_A , M_B , M_C für Knoten und drei Mengen für die Knoten M_1 , M_2 , M_3 eingeführt [125 S.270].

- M_A beinhaltet die Knoten, zu denen der kürzeste Pfad bereits bekannt ist.
- M_B beinhaltet die Kandidatenknoten, die zur Menge M_A überführt werden können und zu mindestens einem Knoten aus M_A eine Verbindung haben.
- M_C ist die Menge aller übrigen Knoten.

Ähnlich verhält es sich mit den Mengen der Kanten.

- M_1 beinhaltet die Kanten, die Teil des Pfades sind.
- M_2 beinhaltet die Kandidatenkanten, die potenziell in die Menge M_1 überführt werden können.
- M_3 ist die Menge aller übrigen Kanten des Graphen.

Zu Beginn wird der Startknoten der Menge M_A zugewiesen. Damit wird der erste Knoten in den Pfad aufgenommen. Die restlichen Knoten werden der Menge M_C und alle Kanten der Menge M_3 zugeordnet. Damit ist der Initialzustand gesetzt. Als nächstes werden zwei Schritte iterativ abgearbeitet. [125 S.270]

1. Im ersten Schritt der Schleife werden all die Knoten betrachtet, die eine Verbindung zu dem Knoten haben, der zuletzt in die Menge M_A aufgenommen wurde und zugleich der Menge M_B oder M_C zugehörig sind. Dies sind somit die Nachbarknoten der in M_A aufgenommenen Knoten.
 - a. Ist der Nachbarknoten in der Menge M_B zu finden, so wird geprüft, ob zu diesem Nachbarknoten nun ein kürzerer Weg möglich ist. Ist ein kürzerer Weg möglich, wird dieser in die Menge M_2 aufgenommen.
 - b. Ist der Nachbarknoten in der Menge M_C zu finden, so wird dieser Knoten in die Menge M_B überführt. Die zugehörige Kante wird zu M_2 hinzugefügt.
2. Im zweiten Schritt werden alle in M_B gelisteten Knoten betrachtet. Diese besitzen alle eine Verbindung zum Startknoten, selbst wenn diese über mehrere Knoten verlaufen. Es wird jetzt der Knoten, der den minimalen Abstand vom Startknoten hat, aus der Menge M_B in die Menge M_A überführt, wobei die zugehörige Kante in M_1 gelistet wird.

Schritt 1 und 2 werden so lange wiederholt, bis der Endknoten in M_A übernommen worden ist.

Der Algorithmus wird am Beispiel aus Abbildung 39 genauer beschrieben. Hierbei soll der kürzeste Pfad von A nach F ermittelt werden.

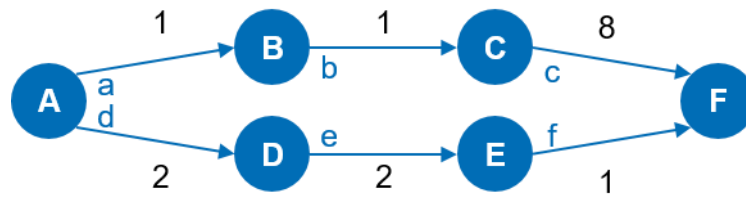

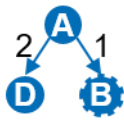


Abbildung 39: Pathfinding mit dem Dijkstra Algorithmus [Eigene Darstellung]

Tabelle 1 beschreibt beispielhaft den Algorithmus Schritt für Schritt in Anlehnung an Abbildung 39. Im ersten Schritt werden die sechs Mengen gebildet. Da A als Startknoten festgelegt wurde, kann dieser bereits in der Menge M_A gelistet werden, weil dieser gesichert Teil des kürzesten Pfades sein muss. Die restlichen Knoten und Kanten werden jeweils in M_C bzw. M_3 gezählt.

Tabelle 1: Initiale Mengen im Algorithmus nach Dijkstra

Erste Schleife		Beschreibung	
M_A	A	Da Knoten A der Startknoten ist, wird dieser Knoten direkt in die Menge M_A überführt.	
M_B			
M_C	B, C, D, E, F		
M_1			
M_2			
M_3	a, b, c, d, e, f		
Zweite Schleife			
M_A	A, B	Beginnend bei Knoten A werden die Nachbarknoten betrachtet. Da der Weg nach B kürzer ist, wird dieser Knoten in die Menge M_A aufgenommen. Damit werden ausgehend von B die Nachbarn ermittelt. Gleichzeitig wird die entsprechende Kante a in die Menge M_1 übernommen.	
M_B	D		
M_C	C, E, F		
M_1	a		
M_2	d		
M_3	b, c, e, f		
Dritte Schleife			
M_A	A, B, C	Die Distanz von A nach C ist mit der Distanz A nach D	
M_B	D		

M _C	E, F	gleichwertig. Da sich keine Verbesserung ergibt, wird Knoten C in die Menge M _A aufgenommen. Die entsprechende Kante wird in die Menge M ₁ aufgenommen.	
M ₁	a, b		
M ₂	d		
M ₃	c, e, f		
Vierte Schleife			
M _A	A, B, C, D	Der Nachbarknoten von C ist Knoten F. Da die Summe A-B-C-F größer als A-D ist, wird D aus M _B in M _A aufgenommen. Gleichzeitig wird die Kante d in die Menge M ₁ aufgenommen.	
M _B	F		
M _C	E		
M ₁	a, b		
M ₂	c, d		
M ₃	e, f		
Fünfte Schleife			
M _A	A, B, C, D, E	Der neue Knoten E wird in M _A aufgenommen. Ebenso die entsprechende Kante d in M ₁ .	
M _B	F		
M _C			
M ₁	a, b, d, e		
M ₂	c		
M ₃	f		
Sechste Schleife			
M _A	A, B, C, D, E, F	Auch Knoten E führt zu Knoten F. Allerdings ist dieser Weg kürzer, sodass A-D-E-F als kürzeste Strecke angenommen wird.	
M _B			
M _C			
M ₁	a, b, d, e, f		
M ₂	c		
M ₃			

Der Algorithmus nach Dijkstra kann aufgrund der kumulierten Betrachtung der Pfadkosten nicht in Graphen ohne Gewichtung oder mit negativer Gewichtung angewendet werden. [Vgl. 126 S.428] In solchen Fällen erweist sich bspw. der Algorithmus nach Bellman-Ford als die geeignetere Methode. [Vgl. 127 S.203]

Der Algorithmus nach Dijkstra wird aufgrund seiner Eigenschaft, den kürzesten Pfad anhand des nächsten aussichtsreichsten Knotens zu bestimmen, als „Greedy“ [dt.: gierig] bezeichnet. [126 S.428]

6.2.1.1.4 A*-Algorithmus

Obwohl erst später beschrieben, kann der A*-Algorithmus [Aussprache: „A-Stern-Algorithmus“] als Erweiterung des Pathfinding-Algorithmus nach Dijkstra verstanden werden. [128 S.538] Der Algorithmus bedient sich bei der Betrachtung der nächsten möglichen Kandidatenknoten einer Schätzfunktion. Die Schätzfunktion liefert einen Hinweis darauf, ob der aktuell aussichtsreichste Pfad auch zukünftig der günstigste sein wird bzw. wie weit schätzungsweise der Weg zum Endknoten ist [Vgl. 92 S.93]. Die Auswahl der Schätzfunktion obliegt dem Systementwickler. In vielen Fällen wie bspw. bei der Betrachtung geografischer Pfade wird der Graph auf ein Raster gelegt. Das Raster kann verwendet werden, um geografische Distanzen wie etwa die euklidische Distanz oder Manhattandistanz zu berechnen [129 S.5]. Diese Schätzfunktionen werden als Heuristiken bezeichnet. Die euklidische Distanz erlaubt die Berücksichtigung von diagonalen Strecken auf dem Raster. Die Manhattandistanz erlaubt lediglich Strecken entlang des Rasters.

6.2.1.1.5 Bellman-Ford-Algorithmus

Der Bellman-Ford-Algorithmus berücksichtigt auch negative Kantengewichtungen. Der Algorithmus durchläuft höchstens $|V| - 1$ Iterationsschleifen, wobei $|V|$ die Anzahl der Knoten bezeichnet [130] nach [131 S. 263]. Im ersten Iterationsschritt wird der Startknoten erfasst, wobei alle weiteren Knoten hinsichtlich Erreichbarkeit und Kostenaufwand analysiert werden. Es werden die Kosten (Gewichtungen) des Pfades notiert, die zum nachfolgenden Knoten führen. Ist kein Weg bekannt, wird der entsprechende Endknoten zunächst übersprungen. Im zweiten Schritt wird ein nächster Knoten fokussiert, wobei wieder alle weiteren Knoten auf Erreichbarkeit und kumuliertem Kostenaufwand überprüft werden. Ist der kumulierte Kostenaufwand geringer als der bisherige Wert, wird der Eintrag zugunsten des neuen Wertes abgeändert. [131 S.263, 132 S.213]

6.2.1.1.6 Floyd-Warshall-Algorithmus

Der Floyd-Warshall-Algorithmus berechnet zunächst die kürzesten Verbindungen zwischen jedem Knotenpaar und setzt zuletzt die Knotenpaare so zusammen, dass ein kürzester Pfad zwischen einem gewählten Start- und Endknoten ermittelt wird. Der Algorithmus wird anhand des Graphen in Abbildung 40a gezeigt bzw. anhand der zugehörigen Adjazenzmatrix aus Abbildung 40b beschrieben. Alle Werte a_{ij} , der Adjazenzmatrix A repräsentieren die Gewichtungen eines Knotenpaares. Dabei muss eine direkte Kante zwischen den Knoten nicht zwingend existieren. Die Gewichtung eines Knotenpaares (i, j) , das keine direkte Kante besitzt, wird als Kosten von Knoten i nach Knoten j verstanden. Allen Knotenpaaren deren Kosten bisher nicht bekannt sind, werden Kosten in Höhe von ∞ zugewiesen. [124 S.207]

Der Algorithmus betrachtet nacheinander die k -te Zeile sowie die k -te Spalte in der Adjazenzmatrix. In Abbildung 40 c-f) sind die k -ten Zeilen sowie die k -ten Spalten mittels blauer Balken hervorgehoben. Hierbei wird der Eintrag a_{ij} mit der Summe aus a_{ik} und a_{kj} verglichen, wobei für den Eintrag gilt: $a_{ij} = \min(a_{ij}, a_{ik} + a_{kj})$. Damit wird der geringste Kostenwert übernommen. [124 S.208, 127 S.208, 133 S.325-327]

Beispielsweise gilt in Abbildung 40b-c mit $k = 0$, $\min(a_{32} = \infty, a_{3k} + a_{k2} = 12) = 12$. Nachdem der Abgleich für alle Elemente durchgeführt wurde, wird k um je einen Wert erhöht, siehe Abbildung 40d-f. Falls der Abgleich zu einem neuen Eintrag führt, kann der Zwischenknoten notiert werden. Damit wird der Zwischenknoten, der zu der günstigsten Verbindung führt, gespeichert, damit der kürzeste Pfad ermittelt werden kann. Andernfalls wären nur die Kosten zwischen zwei Knoten bekannt, nicht aber der Zwischenknoten. In Abbildung 40c-f ist der Zwischenknoten in tiefgestellter Form neben den Neuberechneten Kosten angegeben.

An Abbildung 40c-f kann die intuitive Kernidee des Algorithmus abgelesen werden. Im Vordergrund steht die Berechnung von Teilpfaden. Wird in einem Iterationsschritt k festgelegt, wird für die übrigen Elemente der Pfad über den zu k analogen Knoten ermittelt. Der betreffende Brückenknoten ist in Abbildung 40c der Knoten A, also der Knoten, bei dem sich in der Abbildung die blauen Balken kreuzen. Mit $k = 0$ wurde somit für a_{32} der Weg $D \rightarrow C$ über A ermittelt: $D \rightarrow A \rightarrow C$. Zu beachten ist allerdings, dass diese Betrachtung nicht besagt, dass von D nach A bzw. von A nach D eine direkte Kante vorhanden sein muss. Diese Betrachtung nimmt lediglich an, dass der Teilweg von D nach A bzw. von A nach D unter Umständen bekannt ist.

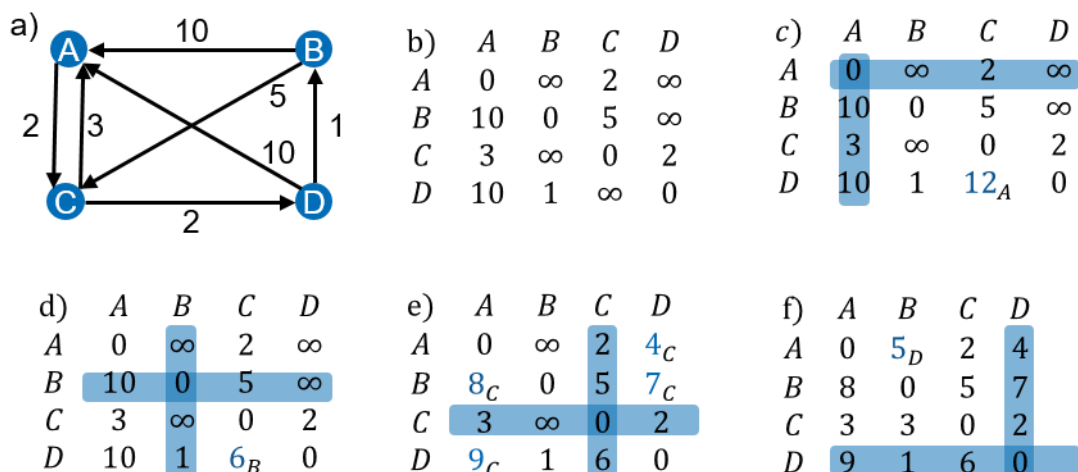


Abbildung 40: Floyd-Warshall-Algorithmus: a) Beispielgraph b) Adjazenzmatrix c) - f) Vorgehen beim Floyd-Warshall Algorithmus [Eigene Darstellung nach [124 S. 210]]

Pro Iterationsschritt ist k als Zwischenknoten für jede mögliche Verbindung zu verstehen [Vgl. 133 S.326]. Abbildung 41a zeigt einen Beispielgraphen und die zugehörige

Adjazenzmatrix in Abbildung 41b sowie die Darstellung des Zwischenknotens in Abbildung 41c. Direkte Verbindungen werden in dem Algorithmus zunächst nicht betrachtet. Bei der Zusammensetzung des kürzesten Pfades wird zuerst der Start- und Zielknoten angegeben. Diese Verbindung wird in Zwischenverbindungen zerlegt. Dabei entsteht eine Baumstruktur, wie in Abbildung 41d zu sehen ist. Aus den Einzelteilen wird der Pfad zusammengesetzt. [124S. 210]

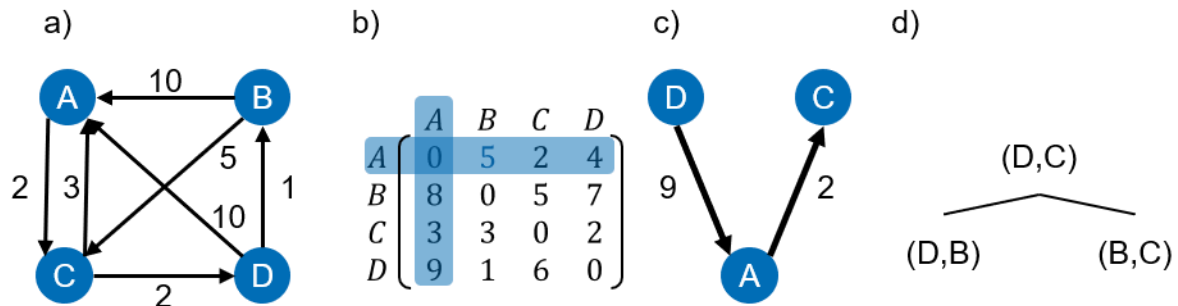


Abbildung 41: Zwischenknoten des Floyd-Warshall Algorithmus a) Beispielgraph b) Adjazenzmatrix mit markierten Zeilen und Spalten c) Algorithmische Betrachtung des Eintrags $e_{4,3}$ als grafische Darstellung d) Baumstruktur zur Zusammensetzung von Pfaden [Eigene Darstellung nach [124 S. 208-211]]

6.2.1.1.7 IDA*- Algorithmus

Der IDA*-Algorithmus kombiniert die iterative Tiefensuche mit dem A*-Algorithmus [92 S.99, 134 S.103]. Hierbei wird aus dem Graphen ein Baum konstruiert. Der Baum beinhaltet in den Verästelungen immer die benachbarten Knoten. Dies kann dazu führen, dass derselbe Knoten im Baum mehrmals vorzufinden ist. Der Baum wird zur Laufzeit des Algorithmus erweitert. Erweitert werden jedoch nur die Zweige, deren Kosten innerhalb einer gewissen variierenden Grenze liegen. Der Algorithmus wird so lange durchgeführt, bis der Zielknoten erreicht ist. [134S. 103]

6.2.2 Theoretische Umsetzung

Für die Generierung eines Nacharbeitsplans wird auf eine Graphenstruktur zurückgegriffen. In diesem Kapitel wird die Erstellung eines Graphen aus der Produktstruktur beschrieben. Anschließend wird mithilfe eines Pathfinding-Algorithmus ein Pfad ermittelt, aus dem der Nacharbeitsplan erstellt wird.

Halbautomatisierte Generierung von Produktzustandsgraphen

Bei der Assistenz des Mitarbeiters ist die Kenntnis über den Produktzustand eine wesentliche Voraussetzung. Der Produktzustand bezeichnet die Produktionsstufe, in der sich das Produkt im Laufe seiner Herstellung (Fertigung oder Montage) jeweils befindet. Basierend auf dem Produktzustand können situativ geeignete bzw. erforderliche Prozesse eingeleitet werden. Wird ein manueller Produktionsbereich betrachtet, der bspw. als Manufaktur organisiert ist, so muss berücksichtigt werden, dass dem Mitarbeiter sehr viele Möglichkeiten zur Zusammensetzung (oder Demontage) des Produktes zur Verfügung stehen. Die Möglichkeiten beinhalten neben alternativen Prozessen

auch die Freiheit, Prozesse rückgängig zu machen. Die Beziehung der Produktzustände wird in Graphen dargestellt. Hierbei repräsentiert jeder Knoten den Produktzustand und jede Kante den Montageprozess, der von einem Produktzustand in einen anderen Produktzustand führt. Da solch ein Graph sehr umfangreich werden kann, wird im Folgenden eine Methode zur halb-automatisierten Generierung des Produktzustandsgraphen vorgestellt. Ausgangspunkt ist eine durch den Planer generierte Verbaureihenfolge, die in Tabelle 2 in Form einer Adjazenzmatrix dargestellt wird. Die Verbaureihenfolge gibt unter Beachtung der Verbaubarkeit an, welche Bauteile/ Baugruppen nach einer bestimmten Baugruppe/ Bauteil montiert werden können. Für die Verbaubarkeit wird eine Matrix erstellt. Die Matrix listet sowohl in den Zeilen als auch in den Spalten alle Bauteile/ bzw. Baugruppen auf. Bei den Baugruppen gilt es jedoch zu bedenken, dass diese zunächst nicht samt ihrer Bauteile berücksichtigt werden. Jede Baugruppe muss separat in einer eigenständigen Adjazenzmatrix berücksichtigt werden.

Tabelle 2: Adjazenzmatrix eines Beispielproduktes

Vorgänger/ Nachfolger	A	B	C	D	E
A	0	0	1	1	1
B	0	0	1	1	1
C	0	0	0	0	1
D	0	0	0	0	1
E	0	0	0	0	0

Zur automatisierten Generierung eines Produktzustandsgraphen wird ein Algorithmus entwickelt. Dabei wird als Basis die Adjazenzmatrix wie beispielsweise aus Tabelle 2 herangezogen und kann interpretiert werden als die Auflistung der Vorbedingungen (Spalten) für ein zu montierendes Bauteil (Zeilen). Sie hat die Dimension $n \times n$, wobei n von der Anzahl betrachteter Bauteile/ Baugruppen abhängt. Die Knoten werden als Binärvektoren repräsentiert. Ein Binärvektor gestattet unter Berücksichtigung eines weiteren Ein-Schritt entfernten Binärvektors die Nachvollziehbarkeit über die bisher demontierten Bauteile. Dem Initialknoten wird der Nullvektor als Binärvektor zugewiesen. Die Dimension des Binärvektors entspricht $1 \times n$. Ausgehend vom Initialvektor wird unter Hinzunahme der Adjazenzmatrix geprüft, welche Knoten als Nachfolger gesetzt werden können. Es wird geprüft, ob ein Nachfolger existiert, dessen Vorbedingungen in Form von bereits bestehenden Knoten erfüllt sind. Dabei können auch mehrere Knoten die Bedingung erfüllen. Im Beispiel aus Tabelle 2 ist dies sowohl für Knoten A als auch für Knoten B der Fall. Daher werden beiden Knoten als Nachfolger des Initialknotens gesetzt. Die entsprechenden Zustandsknoten werden wie folgt dargestellt: $(1,0,0,0,0)$; $(0,1,0,0,0)$. Für beide Knoten wird erneut geprüft, welche weiteren Elemente jeweils als Nachbarn gesetzt werden können. Es kann bspw. nach dem Binärvektor

(1,0,0,0,0) unter Hinzunahme des Knoten B als nachfolgender Zustand (1,1,0,0,0) gesetzt werden. Auch für den Zustand (0,1,0,0,0) kann als Nachfolger (1,1,0,0,0) gesetzt werden. Für alle nicht abgehandelten Knoten wird in mehreren Iterationen geprüft, ob und welche Knoten als nächste Zustände gesetzt werden können. Der Vorgang wird so lange wiederholt, bis jede Zustandsabfolge in einem Einsvektor endet. Der Einsvektor zeigt an, dass alle Bauteile bzw. Baugruppen eingebaut wurden bzw. im Endprodukt enthalten sind. Das Ergebnis des Algorithmus für das Beispiel in Tabelle 2 wird in Abbildung 42 als Diagramm dargestellt.

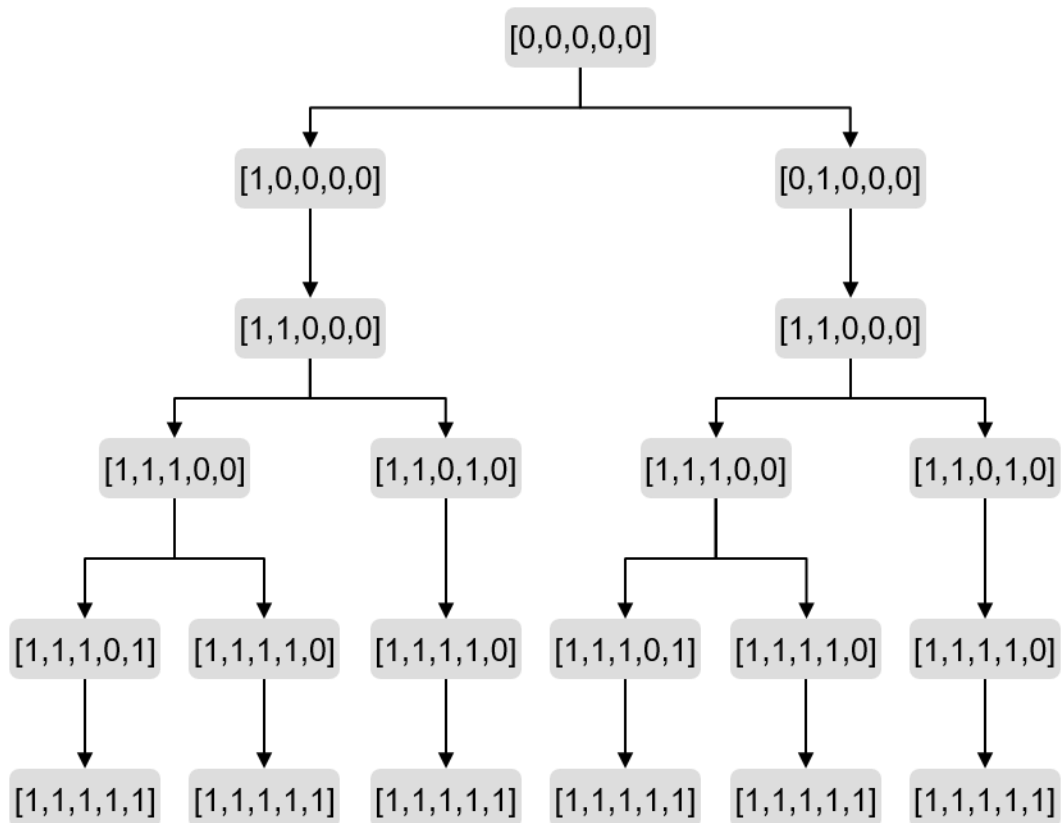


Abbildung 42: Redundanter Produktzustandsgraph als Ergebnis des Algorithmus [Eigene Darstellung]

Zugunsten der Übersichtlichkeit kann der Graph so reduziert werden, dass kein Zustandsknoten mehr als einmal auftaucht. Das Ergebnis ist in Abbildung 43 zu sehen.

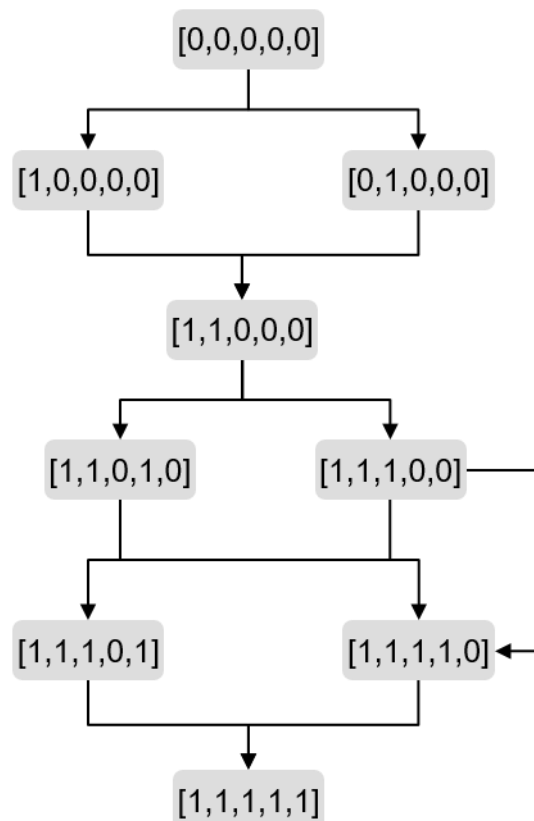


Abbildung 43: Zusammenführung der Knoten zur Vereinfachung des bisherigen Produktzustandsgraphs [Eigene Darstellung]

Gibt die Produktkonstruktion wenig Montagerestriktionen vor, kann die Größe des Zustandsraums selten wirtschaftlich sinnvoll gehandhabt werden. Daher ist es unter Umständen ratsam, produktunabhängige Restriktionen aufzustellen und umzusetzen, bspw. indem Betriebsmittel vorenthalten werden oder nur unter bestimmten Voraussetzungen freigegeben werden. Zudem können Basisbauteile wie eine (Grund-)Platte ausgewählt werden, die als erstes Bauteil bestimmt wird und alle weiteren Bauteile/Baugruppen aufnimmt.

Der Zustandsgraph gibt lediglich die erreichbaren Zustände wieder. Der zugehörige Prozessgraph, der die Nacharbeitsprozesse beschreibt, lässt sich durch den entsprechenden Kantengraphen darstellen, der in Abbildung 44 gezeigt wird. Der Prozessgraph unterscheidet sich von einem Vorranggraph in drei Punkten:

1. Den Prozessen sind (noch) keine Betriebsmittel zugewiesen
2. Die Prozesse sind nicht eingetaktet
3. Die Verbauezeitpunkte sowie eindeutige Sequenzen sind nicht festgelegt

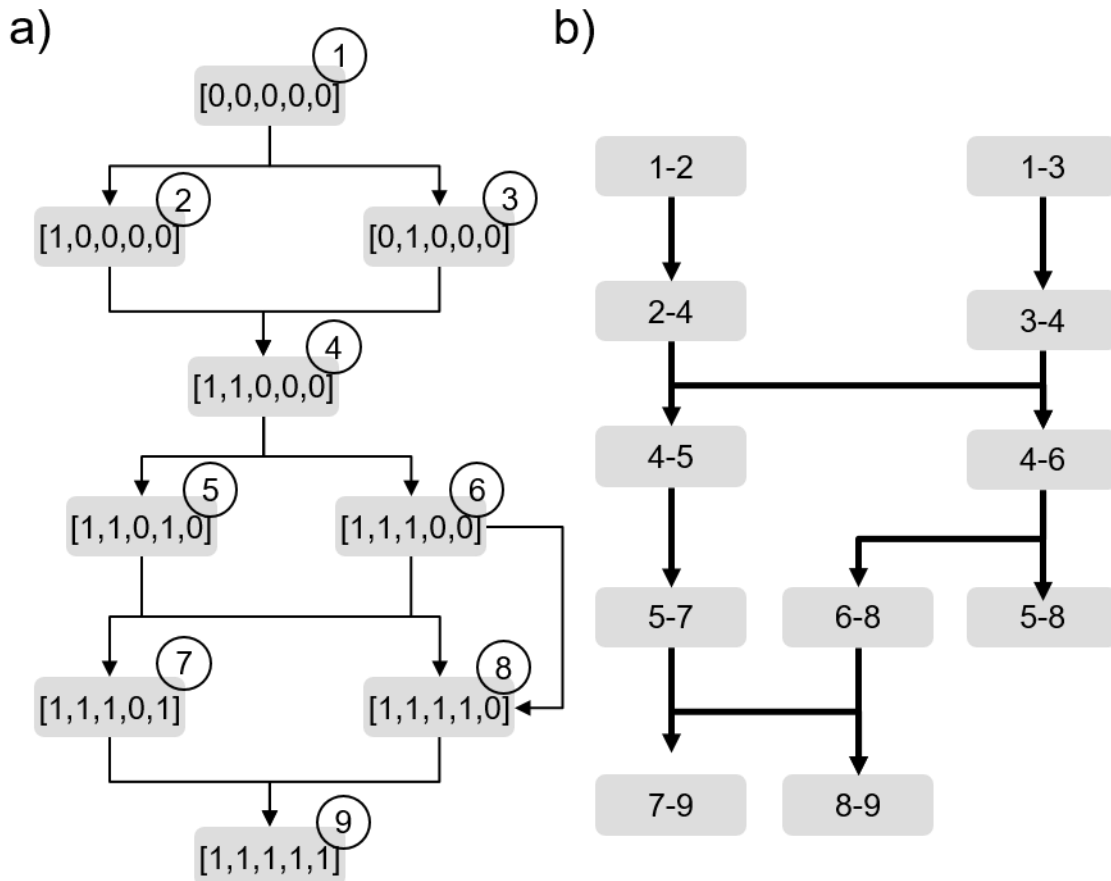


Abbildung 44: a) Zeigt den Produktzustandsgraph b) zugehöriger Kantengraph [Eigene Darstellung]

Die Pfeile im Produktzustandsgraph in Abbildung 44a zeigen in die Demontagegerichtung. Allerdings können die gleichen Zustände auch im Remontageprozess erreicht werden. Der Unterschied liegt im Montageprozess. So kann ein Produktzustand bspw. durch das Entnehmen eines Bauteils erzielt werden, aber auch durch das wiederholte Einsetzen desselben Bauteils. Die Remontagegerichtung wird durch den transponierten Graphen dargestellt. Dieser weist dieselbe Menge an Knoten auf, wobei die Pfeile in Bezug zum ursprünglichen Graph in die entgegengesetzte Richtung zeigen. Zugunsten der Überschaubarkeit können zwei Prozessgraphen (Demontageprozess und Remontageprozess) erstellt werden.

Beide Prozessgraphen (Kantengraph) müssen im nächsten Schritt mit prozessspezifischen Daten angereichert werden. Vor dem Hintergrund des Assistenzsystems werden alle vier Datenklassen (Information, Prozessausführung, Prozessprüfung und Dokumentation) berücksichtigt. Des Weiteren ist es möglich, weitere Prozesse in einem Knoten zu definieren, wenn diese zur Erreichung des nächsten Produktzustands erforderlich sind.

Ausgaben von digitalen Nacharbeitsplänen zur Assistenz des Mitarbeiters

Da der Nacharbeitsauftrag vom vorliegenden Fehlerfall abhängt, ergibt sich eine hohe Zahl potenzieller digitaler Nacharbeitspläne. Daher müssen die Nacharbeitspläne flexibel erstellt werden. Der herkömmliche Ansatz besteht aus der Erstellung aller Prozesssequenzen. Dieser Ansatz birgt in der Nacharbeit den Nachteil, dass neben der hohen Anzahl der Nacharbeitspläne auch die Pflege dieser mit einem ebenso hohen Aufwand betrieben werden muss. Zur flexiblen Erstellung des fehlerabhängigen Nacharbeitsplans wird auf einen Graphen zurückgegriffen. Der Graph leitet sich von der Produktstruktur ab und wird als Produktzustandsgraph ausgelegt. Die Knoten des Produktzustandsgraphen geben den Produktzustand an, wobei die Kanten den Übergang von einem Produktzustand in den nächsten Zustand symbolisieren und somit als Prozesse zu verstehen sind. Der Produktzustandswechsel wird somit von einem Montageprozess vollzogen.

Der Produktzustandsgraph stellt zunächst lediglich eine Struktur zur Abbildung der Vorgänger-Nachfolger-Beziehungen dar. Dieser muss weiter mit Prozessdaten angereichert werden. Prozessdaten umfassen Daten zur **Prozessinformation** zum Montageprozess, zur **Prozessausführung**, zur **Prozessprüfung** sowie zur **Prozessdokumentation**. Um die Prozesse mit Daten anzureichern, wird der Produktzustandsgraph in seinen Kantengraph umgewandelt. Diese Umwandlung dient lediglich der Hervorhebung der Prozesssequenzen. In der Realität erlauben unterschiedliche Programme die Anreicherung der Kanten mit entsprechenden Daten. Nach der Umwandlung steht ein Graph zur Verfügung, in dem Prozesse als Knoten abgebildet sind. Dies erleichtert die Handhabung der Prozessdaten sowie deren Pflege. Der Übergang von einem Prozesszustand zum Nächsten kann mehrere Prozesse beanspruchen. Daher ist es möglich, den Vorranggraphen um weitere Knoten zu erweitern. Dem Prozessknoten werden im nächsten Schritt Prozessparameter in den vier Datenklassen zugewiesen.

Parameter zur Information umfassen Angaben zur Prozessdurchführung. Diese können multimodal bereitgestellt werden. Eine gängige Methode ist die Informationsbereitstellung über einen Monitor in Form eines Werkerführungssystems [Vgl. 135, 136]. Hierbei wird dem Mitarbeiter meist ein Text eingeblendet und der Prozess visuell dargestellt. Die Informationsbereitstellung kann als Mensch-Technik-Interaktion betrachtet werden. Diese findet in Form eines (eingeschränkten) Dialogs statt. Einerseits stellt das Assistenzsystem Informationen bereit und führt weitere Aktionen durch, andererseits muss auch die Möglichkeit seitens des Mitarbeiters bestehen, der Maschine neue Informationen zu melden. Für die Auswahl geeigneter Kanäle zur Übermittlung der Informationen an den Mitarbeiter werden die folgenden Punkte berücksichtigt:

- Welchen Montageprozess führt der Mitarbeiter aus?
- Welche Information benötigt der Mitarbeiter?
 - Welche Notwendigkeit hat die Information?
 - Wie schwer ist die Information zu verstehen?

Im Folgenden wird die Praktikabilität verschiedener Medien hinsichtlich der Informationsbereitstellung bewertet (siehe Tabelle 3). Dabei werden Medien berücksichtigt, die den Mitarbeiter visuell, auditiv oder haptisch ansprechen. Kinästhetische und olfaktorische Medien werden aufgrund ihrer Irrelevanz nicht berücksichtigt. Die einzelnen Arten werden hinsichtlich Implementierungsaufwand, Informationsgehalt und dem zeitlichen Aufwand zur Informationsaufnahme im Vergleich zu anderen Medien bewertet. Die Bewertung reicht von 1 bis 4 und wird in Form von Harvey Balls dargestellt. Hierbei steht eine 1 für „sehr gering“ und eine 4 für „sehr hoch“. Zudem wird betrachtet, ob in der Regel eine Unterbrechung des Montageprozesses zwingend erforderlich ist oder nicht.

Tabelle 3: Praktikabilität verschiedener Informationskanäle zur Übermittlung von Informationen an einen Mitarbeiter

Informationskanal		Implementierungsaufwand	Informationsgehalt	Zeitlicher Aufwand zur Informationsaufnahme	Prozessunterbrechung notwendig
visuell	Text	●	●	●	●
	Standbild	●	●	●	●
	Bewegtbild	●	●	●	●
	Lichtsignal	●	●	●	●
	Projektion	●	●	●	○
auditiv	Sprachausgabe	●	●	●	○
	Akustisches Signal	●	●	●	○
haptisch	Vibration	●	●	●	●
	Kraftfeedback	●	●	●	○

Informationsübermittlung auf **textueller** Basis ist mit äußerst geringem Implementierungsaufwand realisierbar. Viele Systeme bieten diese Möglichkeit an. Zudem sind außer einem Rechner und einem Bildschirm keine weiteren Ressourcen notwendig, um Informationen in textueller Form darzustellen. Darüber hinaus können Informationen äußerst dicht verpackt werden. Allerdings ist auch die Informationsaufnahme für

den Mitarbeiter aufwendiger. **Abbildungen** können ebenfalls zur Informationsvermittlung genutzt werden. Zur Erstellung von Stand oder Bewegungsbildern bedarf es zusätzlicher Mittel, wodurch diese Form etwas aufwändiger in der Pflege ist. Allerdings kann durch ein Standbild in der Regel weniger Information transportiert werden. Bewegungsbilder wie Animationen oder sonstige Videoclips erlauben einen höheren Informationsgehalt. Beide Inhalte können einfach aufgenommen werden. Ein weiteres Beispiel für die visuelle Informationsbereitstellung ist das **Lichtsignal**, das durch entsprechende Möglichkeiten bspw. über eine Lampen- oder Ampelsteuerung eingerichtet werden muss. Allerdings können hierdurch sehr geringe Informationsmengen vermittelt werden. Meistens beschränken sich diese auf Hinweise hinsichtlich einer vorliegenden Störung oder eines planmäßigen, korrekten Zustands. Ein weiteres Mittel sind **Projektionen**, die lediglich in Kombination mit der Projektionsfläche auftreten und den kompletten Informationsgehalt wiedergeben. Somit ist nicht etwa die Projektion einer grafischen Nutzeroberfläche als Alternative zu einem Bildschirm gemeint. Vielmehr geht es darum, Oberflächen anzustrahlen, die selbst Teil der Montageaufgabe sind, beispielsweise, wenn auf diesen ein Gegenstand abgelegt werden soll. Das Einrichten dieser Information ist sehr aufwendig, da zunächst ein Projektor eingemessen werden muss und die Gegebenheiten vor Ort bekannt sein müssen. Zudem sind hierfür spezielle Programme notwendig, derer es unter Umständen einer Schulung bedarf. Allerdings sind die Informationen bei bekanntem Kontext, aber auch intuitiv vom Mitarbeiter zu interpretieren. Zudem muss zur Informationsaufnahme der Arbeitsfluss nicht zwangsläufig unterbrochen werden.

Mitarbeiter können ebenfalls auditiv angesprochen werden. Eine Methode hierfür sind **Sprachanweisungen**. Allerdings müssen diese vorher ggf. aufgenommen werden. Alternativ sind Text-to-Speech-Systeme nutzbar. In beiden Fällen sind geeignete Systeme bzw. Ausrüstung notwendig. Ähnlich wie beim Text ist eine hohe Informationsdichte möglich. Im Gegensatz zum Text können die Sprachanweisungen bei akustisch ungünstigen Umgebungsbedingungen nur erschwert aufgenommen werden. Ein **akustisches Signal** transportiert wenig Information, kann aber ebenfalls einfach vom Mitarbeiter interpretiert werden.

Modernere Technologien auch in Form von Wearables ermöglichen haptisches Feedback. Eine bekannte Form ist das Feedback über **Vibrationen**. Existiert keine geeignete Software, muss ein Vibrationssignal ggf. in eine Applikation programmiert werden. Allerdings können Vibrationen einfach vom Mitarbeiter interpretiert werden, wobei ebenfalls lediglich geringe Informationsmengen transportiert werden können. **Kraftfeedbacks** bringen vor allem bei Steuerungen und Bewegungen eine Gegenkraft auf.

Die einzelnen Kanäle können auch in Kombination genutzt werden. Hierbei wird aber die adäquate zeitgleiche Beanspruchung mehrerer mentaler Ressourcen nach Wickens berücksichtigt. So werden häufig Informationen sowohl textuell als auch bildlich aufbereitet, sodass unterschiedliche mentale Ressourcen angesprochen werden. Zudem ertönt beim Abschließen des Prozesses ein Ton, sodass auch hier durch ein akustisches Signal der Erfolg des Prozesses mitgeteilt wird, ohne dass der Mitarbeiter

wesentlich in seiner Prozessausführung beschränkt wird. Beim Einrichten des Assistenzsystems ist zudem zu beachten, wie kritisch die Information ist. Einerseits soll der Prozess möglichst nicht ins Stocken geraten, andererseits muss der Einrichtungsaufwand zur Informationsvermittlung vertretbar sein. Bei der Entscheidung, in welchem Umfang Informationen zu einem Montageprozess bereitgestellt werden, muss der Montageprozess betrachtet werden. Zum einem muss die Kritizität des Fehlers berücksichtigt werden bspw. durch eine FMEA (Fehlermöglichkeits- und Einflussanalyse), zum anderen, welche Produkt-, Prozess-, und Ressourcenparameter betroffen sind. So sind bspw. bei einem Schraubprozess neben den Schraubparametern auch die Anzahl der Schrauben sowie die Verschraubpunkte (ggf. die Schraubsequenz) wichtige Parameter. Bei einem halbautomatisierten Prozess muss dem Mitarbeiter bspw. die Anzahl der Schrauben sowie der Verschraubpunkt bekannt sein. In diesem Fall eignet sich ein Projektionssystem zum visuellen Hervorheben der Verschraubpunkte. Hörauf gibt einen Überblick über Montageprozesse sowie zugehöriger Eigenschaften und Parameter [137]. Dieser legt den Schwerpunkt auf Prozesse aus dem Maschinenbau. Jedoch handelt es sich um eine erweiterbare Liste, sowohl was die Art der Prozesse betrifft als auch deren Parameter.

Unterschiedliche Fehlerfälle erfordern entsprechende Maßnahmen zu ihrer Behebung. Diese Fehlerfälle werden in Form von Fehlerverhaltenscodes angegeben, wie in Kapitel 5.1 beschrieben. Der Fehlerverhaltenscode weist dem Fehlerverhalten des Produktes bspw. Leckage oder akustische Emission eine Beschreibung zu. Über die tatsächliche Ursache können Vermutungen angestellt werden. Im Rahmen dieser Arbeit wird zunächst von dieser Annahme abgewichen, da auch in der Praxis versucht wird, anhand des beobachteten Fehlerverhaltens die Ursache abzuleiten. Daher wird ein Fehlercode eingeführt, der die Fehlerursache angeben soll. Der Fehlercode lässt Schlussfolgerungen zum Fehler sowie den Fehlerort im Produkt zu und verweist auf die erforderlichen Maßnahmen zur Behebung des Fehlers. Der Fehlercode wird im Produktzustandsgraphen an einem Prozess angebracht.

Geht man davon aus, dass insbesondere in der Nacharbeit der Initialzustand eines zu demontierenden Produktes das vollständige Produkt ist (d. h. das Produkt wurde vollständig montiert und fiel im Prüfstand als fehlerhaft auf), so wird dieser Zustand als Startzustand in einem Graphen verwendet. Der Nacharbeitsplan kann ermittelt werden, indem der Pfad zum Zustand mit dem Fehlercode ermittelt wird. Hierzu eignen sich unterschiedliche Graphtraversierungsalgorithmen.

Eine einfache Methode der Graphtraversierung ist der Algorithmus nach Dijkstra. Dieser garantiert die Ermittlung des kürzesten Pfades bei gewichteten Kanten. Abbildung 28 zeigt einen Pfad, der ausgehend vom komplettierten Produkt zunächst den Demontagepfad zum defekten Bauteil oder Baugruppe anzeigt. Im Anschluss folgt die Fehlerbehebung. Schlussendlich muss der Remontagepfad beschriftet werden, um das Produkt wieder zusammenzubauen und somit in den komplettierten (und verkaufsfähigen) Zustand zu versetzen. Zur Verwendung des Pathfinding-Algorithmus nach Dijkstra gilt es zu beachten, dass die Gewichtung einer Kante nicht negativ ausfallen darf. Bei der

Anwendung des Pathfinding-Algorithmus zur Erstellung des Nacharbeitsplans wird ein gerichteter Graph (alternativ auch Multigraph) verwendet. Die Richtung ist notwendig, weil die Montageprozesse sequenzbedingten Restriktionen unterliegen. Gerichtete Graphen garantieren die Einhaltung dieser Restriktionen, vor allem wenn in einem Graphen Zyklen vorkommen. Es existieren unterschiedliche Methoden, die Gewichtung einer Kante zu bestimmen. Wiesbeck verwendet bspw. bei der Gewichtung von Kanten, die Prozesse abbilden, eine Methode, die auf Systemen vorbestimmter Zeiten (ähnlich zu MTM) basiert [Vgl. 22 S.88ff.]. Eine etwas simplere Methode ist die Gewichtung durch die Berücksichtigung der Anzahl der Schritte zum Beheben des Fehlers. Hierzu sind die Demontage sowie die Remontage zu berücksichtigen, da diese für die Fehlerbehebung zunächst nicht maßgeblich sind, bspw. ist der Demontepfad zum fehlerbehafteten Bauteil für die Fehlerbehebung irrelevant. Hingegen muss die Fehlerbehebung nicht berücksichtigt werden, da die Umfänge im Vergleich zur De-/Remontage gering sind, meist komplett durchgeführt werden müssen und in dieser Betrachtung kein Optimierungspotenzial bieten. Diese Methode erlaubt die gleiche Gewichtung aller Prozesse. In diesem Fall ist der kürzeste Pfad derjenige, der die kürzeste Anzahl an Demontageprozessen zur Behebung des Fehlers enthält. Dies führt dazu, dass Pfade, die den Ausbau von Baugruppen berücksichtigen, den Vorzug erhalten und nicht Pfade, die die grundlose Demontage und Zerlegung der Baugruppe enthalten. Hier gilt es zu berücksichtigen, dass die entsprechenden Ressourcen bzw. Betriebsmittel für den Prozess vorhanden sein müssen. Ein Mitarbeiter kann somit nicht eine Baugruppe herausnehmen ohne entsprechende Betriebsmittel zur Verfügung stehen zu haben. Selbst die manuelle Bewältigung des Prozesses steht außer Frage, da der Montageprozess bestimmte Betriebsmittel benötigt oder der Prozess ohne Hilfsmittel unergonomisch sein könnte.

6.2.3 Praktische Umsetzung der Assistenzfunktion

In der praktischen Umsetzung wird die Erstellung sowie die Ausgabe des Nacharbeitsplans nach Eingabe eines Fehlercodes beschrieben. Als Beispiel wird der Bereich des Vorderachsantrieb des Getriebes analysiert. Dieser Bereich des Getriebes besteht sowohl aus Baugruppen als auch aus Bauteilen, die teilweise seriell und teilweise parallel montiert werden müssen. Ausgehend von einer Strukturstückliste wird eine Vorgänger-Nachfolger-Beziehung erstellt. Der betrachtete Produktausschnitt besteht aus den folgenden Bauteilen:

1. 6 Aluminiumschrauben
2. Verteilergehäuse
3. Torsendifferential
4. Zwischengehäuse
5. 12 Aluminiumschrauben
6. Antriebsrad
7. Abtriebsrad
8. Abtriebsradwelle

Abbildung 45 zeigt den zur Adjazenzmatrix aus Tabelle 4 zugehörigen Produktzustandsgraphen. Der oberste Knoten zeigt das komplettierte Produkt. Der letzte Knoten (unten im Diagramm) zeigt den letzten betrachteten Zustand, in dem das Produkt komplett zerlegt wurde.

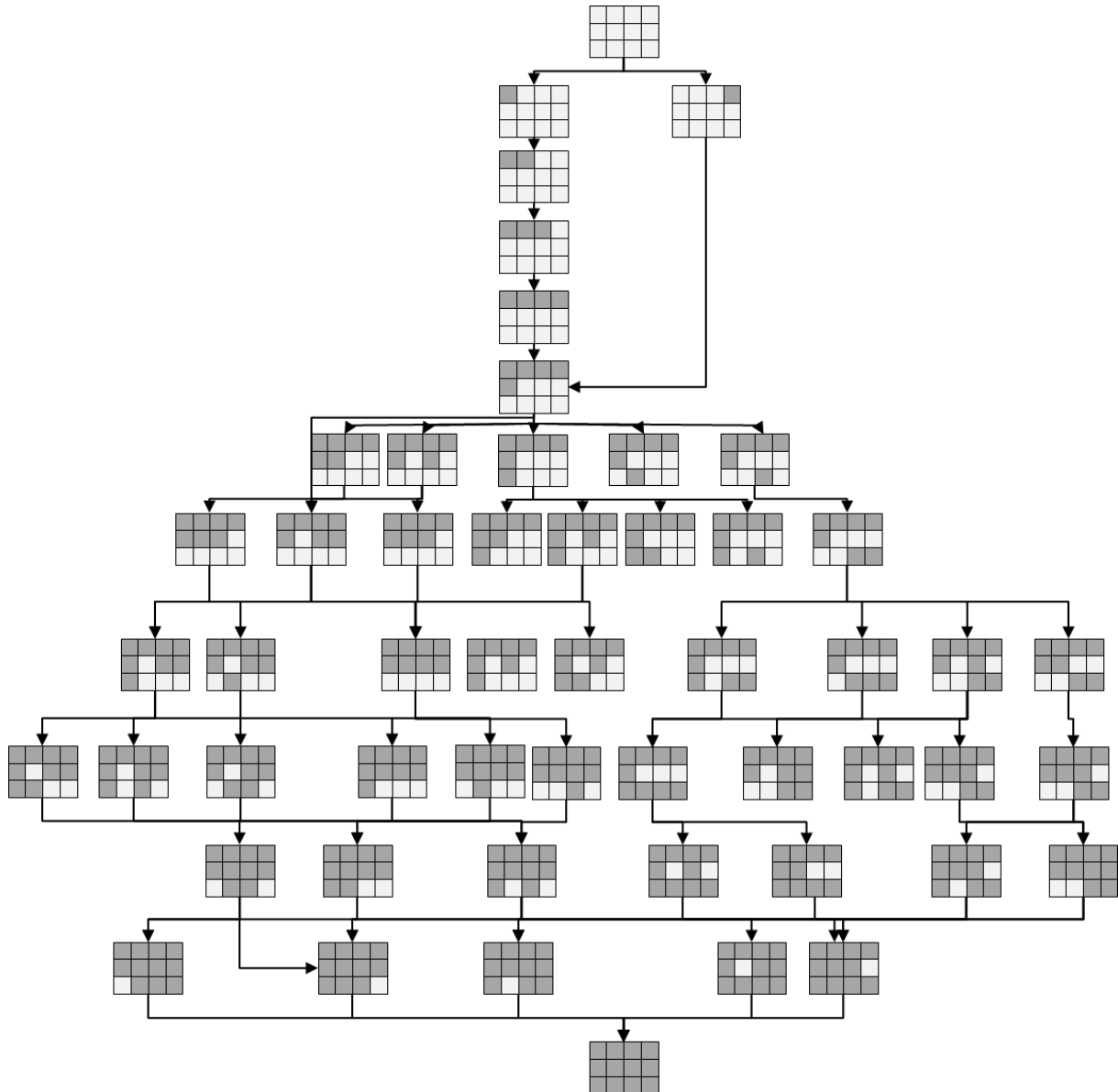


Abbildung 45: Produktzustandsgraph des Vorderachsantriebs des Getriebes [Eigene Darstellung]

Abbildung 46 zeigt die Umsetzung der Adjazenzmatrix aus der Montagesicht in einem Tabellenkalkulationsprogramm. Hierbei gibt der Nutzer ein, welches Bauteil in der Zeile vor dem Bauteil in der Spalte montiert sein muss.

	A	B	C	D	E	F	G	H	I	J	K	L	M
		6 Schrauben am Verteilergehäuse	Verteilergehäuse	Torsendiferential	12 Schrauben am Zwischengehäuse	Zwischengehäuse	Abtriebsrad	Abtriebswelle	Antriebsrad	Magnet	Dichtring	2 Stahlschrauben	Seitenwelle
1													
2	6 Schrauben am Verteilergehäuse	0	0	0	0	0	0	0	0	0	0	0	0
3	Verteilergehäuse	1	0	0	0	0	0	0	0	0	0	0	0
4	Torsendiferential	1	1	0	0	0	0	0	0	0	0	0	0
5	12 Schrauben am Zwischengehäuse	0	0	0	0	0	0	0	0	0	0	0	0
6	Zwischengehäuse	0	0	0	1	0	0	0	0	0	0	0	0
7	Abtriebsrad	0	0	0	1	0	0	0	0	0	0	0	0
8	Abtriebswelle	0	0	0	1	1	0	0	0	0	0	0	0
9	Antriebsrad	0	0	0	1	1	0	1	0	0	0	0	0
10	Magnet	0	0	0	1	1	0	0	0	0	0	0	0
11	Dichtring	0	0	0	1	1	0	0	0	0	0	0	0
12	2 Stahlschrauben	0	0	0	1	1	0	0	0	0	0	0	1
13	Seitenwelle	0	0	0	0	0	0	0	0	0	0	0	0
14													

Abbildung 46: Erstellung der Adjazenzmatrix im Tabellenkalkulationsprogramm [Eigene Darstellung]

Mit dieser Tabelle wird ein *Python*-Script ausgeführt, das den Algorithmus zur Erstellung des Produktzustandsgraphen implementiert hat. Nach Ausführung des Scripts wird in einer Dokumenten- und Graphendatenbank der Produktzustandsgraph abgespeichert. Abbildung 47 zeigt den Produktzustandsgraphen in der Graphendatenbank. Zu erkennen sind die Binärvektoren, die als Labels an den Knoten des Graphen zu sehen sind.

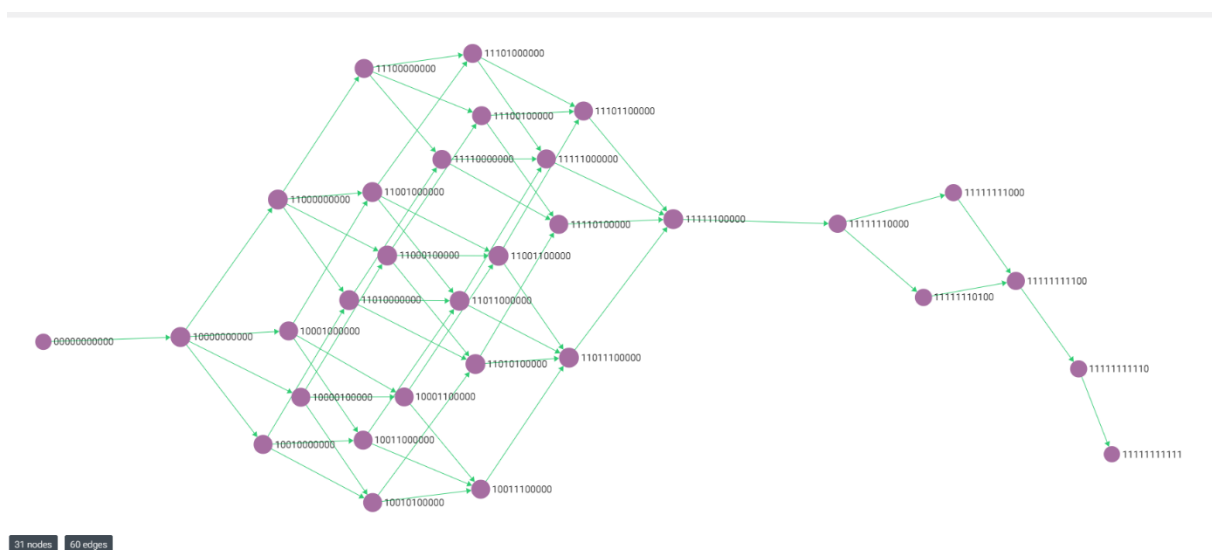


Abbildung 47: Halbautomatisiert generierter Produktzustandsgraph [Eigene Darstellung]

Zur Anreicherung des Graphen mit Prozessdaten werden die Kanten ausgewählt. Eine Umwandlung des Produktzustandsgraphen in den Kantengraphen ist praktisch nicht erforderlich. Die Kanten sind mit einem Dokument verknüpft. Mehrere Kanten können mit dem gleichen Dokument verknüpft sein. Dies ist dann der Fall, wenn durch mehrere Kanten unter bestimmten Voraussetzungen exakt das gleiche Bit im Binärvektor verändert wird. Hierdurch muss dieser Prozess lediglich einmal angelegt werden. Die Zuweisung der Dokumente an eine Kante geschieht ebenfalls automatisiert bei der Ausführung des *Python*-Scripts sowie der Erstellung des Produktzustandsgraphen. Danach erfolgt die Definition der Nacharbeitsprozesse, wie in Kapitel 6.1 beschrieben.

Im nächsten Schritt werden an den Prozessen Fehlercodes definiert. Dabei wird die JSON-Datei um einen weiteren Eintrag erweitert. Der Grund für die Definition des Fehlercodes in den Prozessen und nicht in den Produktzuständen ist, dass eine Kante mehrere Prozesse enthalten kann. Einige Prozesse sind aber unter Umständen nicht durchzuführen. Beispielsweise muss ein Bauteil, das in der Nacharbeit entsorgt wird, nicht vorher sorgfältig abgestellt werden. Allerdings wird jeder Fehlercode für eine Kante erstellt.

Weiterhin werden Fehlerbehebungsprozesse definiert. Diese sind analog zu den Kanten aufgebaut. Allerdings sind diese nicht Teil des Produktzustandsgraphen. Abbildung 48 zeigt alle Dokumente und Kantendateien, die angelegt werden. Die Datei „De_Re_Montageprozesse“ beinhaltet die Prozessdefinitionen. Das Dokument „Fehlerbehebungsprozesse“ beinhaltet alle Daten zu den Fehlerbehebungsprozessen. Das Dokument „GenerischeFehlerbehebungsprozesse“ beschreibt die Fehlerprozesse allgemein. Gängige Fehler sind: fehlende Bauteile, fehlerhafte bzw. beschädigte Bauteile und verschmutzte Bauteile. Das Dokument „Produktzustaende“ beinhaltet die Knoten des Produktzustandsgraphen.

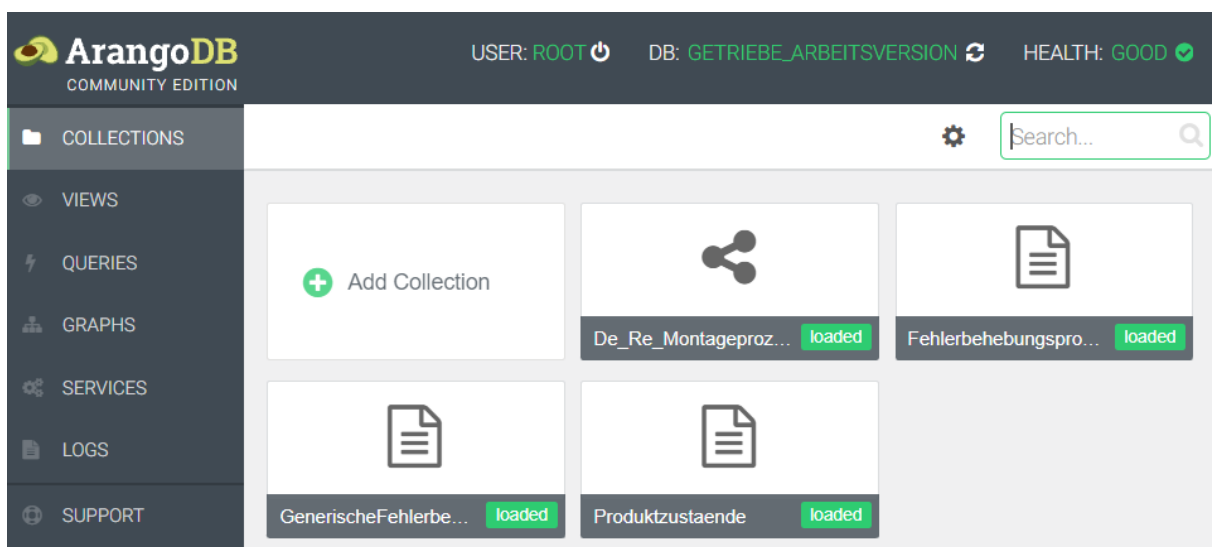


Abbildung 48: Dateien als Bestandteil des Produktzustandsgraphen [Eigene Darstellung]

In der Graphendatenbank wird aus den Dokumenten „Produktzustaende“ und „De_Re_Montageprozesse“ der Produktzustandsgraph erstellt, wie Abbildung 49 zeigt.

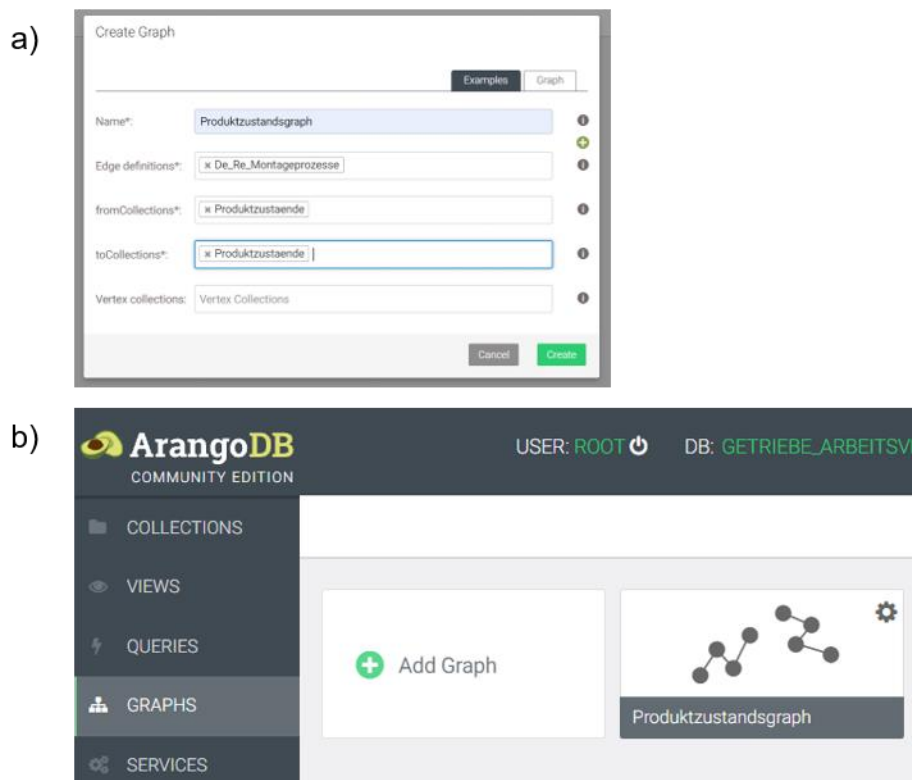


Abbildung 49: Anlegen des Produktzustandsgraphen (a) Zusammensetzung des Produktzustandsgraphen b) Erstellter Produktzustandsgraph. [Eigene Darstellung]

Mit der Fertigstellung des Produktzustandsgraphen samt zugehöriger Prozessdaten und den Fehlerbehebungsprozessen ist die Datenbasis für die zweite Entwicklungsstufe vollständig. Im nächsten Schritt muss der Nacharbeitsplan aus dem Graphen erstellt werden.

Hierzu wird über die Eingabemaske des Mitarbeiterführungssystems der Fehlercode eingegeben. Das Mitarbeiterführungssystem ist über eine TCP-Schnittstelle mit dem Steuerungsprogramm des Kognitiven Assistenzsystem verbunden. Bei Eingang eines neuen Fehlercodes samt Produktbezeichnung und Mitarbeiternamen wird der Produktzustandsgraph aus der Graphendatenbank abgerufen und unter Anwendung des Pathfinding-Algorithmus nach Dijkstra der kürzeste Pfad vom ersten Nacharbeitsprozess des komplettierten Getriebes zum Nacharbeitsprozess mit dem Fehlercode ermittelt. Der Pfad enthält sowohl Demontageanweisungen als auch Remontageanweisungen. Mithilfe des Fehlercodes wird zwischen beiden Pfaden der Fehlerbehebungsprozess eingefügt. Der finale Pfad samt zugehöriger JSON-Dateien stellt den digitalen Nacharbeitsplan dar. Somit ist Schritt für Schritt vorgegeben, welche Betriebsmittel angesprochen werden und welche Rückmeldungen im Sinne einer Prüfung von den Betriebsmitteln zu erwarten sind.

Diskussion

In diesem Kapitel wurde die Entwicklung eines Kognitiven Assistenzsystems beschrieben, das für die Bearbeitung von Nacharbeitsaufträgen geeignet ist. Als digitales Rohgerüst dient ein Produktzustands- bzw. Prozessgraph, der Informationen zu den jeweiligen Prozessen enthält. Durch die Anwendung eines Pathfinding-Algorithmus wird automatisiert ein fehlerspezifischer digitaler Nacharbeitsplan ermittelt, der, wie in Kapitel 6.1 beschreiben, den Ablauf für das Kognitive Assistenzsystem bereitstellt.

Sowohl in Kapitel 6.1 als auch in Kapitel 6.2 wird für den erfolgreichen Einsatz des Assistenzsystems die Annahme zugrunde gelegt, dass ein Fehlercode auf die Ursache eines Produktfehlerverhaltens verweist. Vor allem in der Nacharbeit muss von dieser Annahme abgerückt werden, da dem Produktfehlerverhalten unterschiedliche Ursachen zugrunde liegen können. Beispielsweise kann Leckage an einem Produkt aufgrund eines fehlerhaften Dichtrings oder aufgrund inkorrekt verschraubung auftreten. Daher muss dem Mitarbeiter die Möglichkeit eingeräumt werden, von der vorgegebenen bzw. mutmaßlichen Fehlerursache abzuweichen und seinem Verdacht nachzugehen. Kapitel 6.3 beschreibt die Erweiterung des Kognitiven Assistenzsystems hinsichtlich eines veränderbaren Nacharbeitsplans.

6.3 Weiterentwicklung des Kognitiven Assistenzsystems zur Neubewertung der Fehlerursache (Routing)

Der Fehlercode, der einem Nacharbeitsauftrag anhaftet, beruht prinzipiell auf einer Mutmaßung, weswegen die Möglichkeit einer fehlerhaften Ursachenbewertung bestehen kann. Hierzu hat der Mitarbeiter die Option, über die Eingabe eines neuen Fehlercodes in einem Mitarbeiterführungssystem einen neuen Nacharbeitsplan ausgeben zu lassen. Da der Mitarbeiter jederzeit die Möglichkeit hat, seinem Verdacht nachzugehen, muss der neuermittelte Nacharbeitsplan im Startpunkt variabel sein. Die Neuermittlung des Nacharbeitsplans wird als „Routing“ bezeichnet.

An dieser Stelle könnte der Einwand eingebracht werden, dass die Eingabe des Fehlercodes obsolet sei, da der Mitarbeiter nunmehr die bessere Kenntnis über den Prozess besitzt, das Assistenzsystem ohnehin überstimmen kann und demnach dem Montagassistenzsystem überlegen ist. Daher sei keine Assistenz notwendig. Allerdings ist die Fehlercodeeingabe erforderlich, da das Assistenzsystem nicht allein Unterstützungsaufgaben wahrnimmt, sondern zeitgleich auch kontextsensitiv die Betriebsmittel freigibt, die ausgeführten Montageprozesse auf ihre Vollständigkeit und Richtigkeit überprüft und die ausgeführten Nacharbeitsprozesse dokumentiert.

6.3.1 Theoretische Umsetzung

Routing bei Verdachtsfällen

Die Neuausrichtung eines Arbeitsplans unterscheidet sich in zwei wesentlichen Punkten vom ursprünglichen Arbeitsplan. Zum einen gilt als Ausgangszustand nicht das

komplettierte Getriebe, sondern der Produktzustand, ab dem die Neuermittlung des Arbeitsplans angefordert wird. Zum anderen ist der Arbeitsplan nicht mehr symmetrisch, da er zum komplettierten Produkt führen muss. In Abbildung 50 ist zu sehen, dass ausgehend von einem Produktzustandsgraph der Nacharbeitsplan zum Prozess „8-R“ mit dem Fehlercode ECx ermittelt wird. Zur Ermittlung des aktualisierten Nacharbeitsplans wird nach der Eingabe des neuen Fehlercodes überprüft, ob der neue Pfad durch Weiterführen (Erweitern) des bisherigen Arbeitsplans erreicht werden kann. Ist dies der Fall, wird der Arbeitsplan um die neuen Prozesse erweitert, wobei Demontageprozesse, Fehlerbehebungsprozesse sowie die Remontageprozesse angepasst werden. Abbildung 50 zeigt die Pfadermittlung für den Fall, dass der bisher beschrittene Pfad nicht Teilpfad des aktualisierten Nacharbeitsplans sein kann. Dann müssen die Prozesse mittels Remontageprozessen rückgängig gemacht werden. Die Remontageprozesse stehen zu Beginn des neuermittelten Nacharbeitsplans. Ist ein Produktzustand gefunden, von dem ausgehend der relevante Produktzustand erreicht werden kann, wird wieder der Pathfinding-Algorithmus angewendet. Der ermittelte Pfad wird erneut zum Arbeitsplan hinter den Remontageprozessen hinzugefügt. Anschließend werden Fehlerbehebungsprozesse durchgeführt. Danach werden die Prozesse wieder zum zweiten Ausgangspunkt zurückgeführt. Zum Schluss werden von diesem Punkt ausgehend die Remontageprozesse ausgeführt, die bereits Teil des ersten ursprünglichen Arbeitsplans waren.

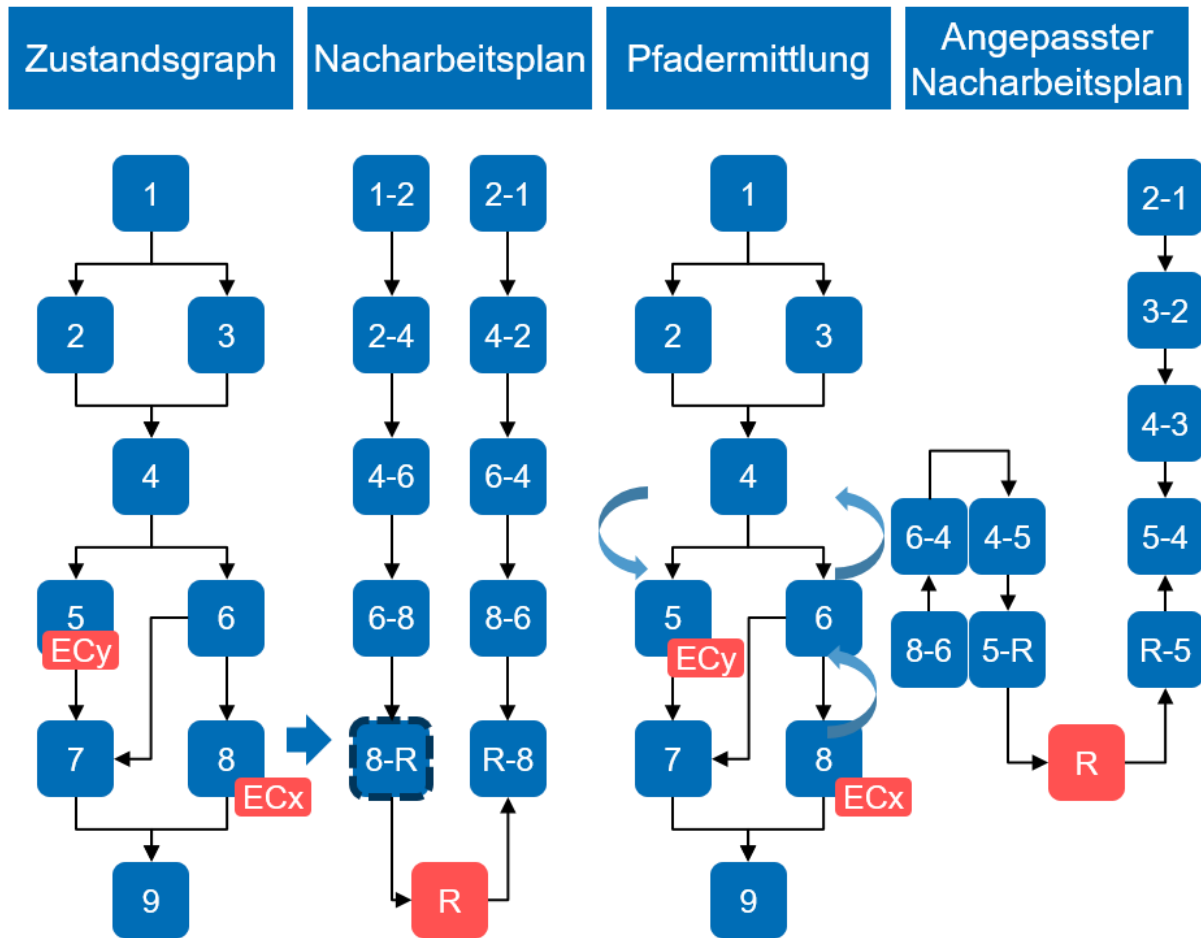


Abbildung 50: Konstruktion des Nacharbeitsplans bei neuer Fehlerursacheneinschätzung [Eigene Darstellung]

6.3.2 Praktische Umsetzung der Assistenzfunktion

Die praktische Umsetzung wird in einem *Python*-Script realisiert. Über die Eingabemaske des Mitarbeiterführungssystems hat der Mitarbeiter die Möglichkeit, in jedem Produktzustand Demontage oder Remontage einen neuen Fehlercode einzugeben und dem Verdacht nachzugehen. Daraufhin verwirft das Steuerungsprogramm den weiteren Nacharbeitsablauf und blendet den neuen Nacharbeitsplan ein. Eine Ausnahme stellt die neue Fehlereingabe im Fehlerbehebungsprozess dar. Hierbei muss zuerst der Fehlerbehebungsprozess abgeschlossen werden, bevor dem neuen Verdacht nachgegangen werden kann.

6.3.3 Diskussion

Das entwickelte Kognitive Assistenzsystem in dieser Entwicklungsstufe erlaubt einem Mitarbeiter einen Fehlercode einzugeben und diesen bei einem Verdacht abzuändern. Hierdurch wird ihm ausgehend von seinem unvollendeten Produktzustand der neue Nacharbeitsplan ausgegeben. Der Nacharbeitsplan führt in der Remontagephase zum komplettierten Produkt.

Alle bisherigen Entwicklungsstufen setzen die strikte Befolgung des Nacharbeitsplans voraus. Weicht ein Mitarbeiter vom vorgegebenen Nacharbeitsplan ab, so verharrt das Kognitive Assistenzsystem im letzten mit dem Nacharbeitsplan übereinstimmenden Prozess. Daher wird in der nächsten Entwicklungsstufe das Assistenzsystem so erweitert, dass der physische und digitale Produktzustand übereinstimmen.

6.4 Weiterentwicklung des Kognitiven Assistenzsystems zur Realisierung der Zustandssynchronisation

Herkömmliche Digitale Kognitive Assistenzsysteme unterstützen den Mitarbeiter durch die kontextbezogene Bereitstellung von Informationen. Dies setzt die Synchronität des digitalen Produktzustands mit dem realen Produktzustand voraus. Andernfalls können prozessfremde Informationen eingeblendet werden und somit zur Verwirrung oder sogar zu Montagefehlern beitragen. Abbildung 51 stellt die schematische Funktionsweise eines herkömmlichen Kognitiven Assistenzsystems dar. Die Phasen Informieren, Prozessdurchführung, Prozessprüfung und Prozessdokumentation wurden bereits in den vorangegangenen Kapiteln erläutert. Durch die Eingabe des Fehlercodes und der darauffolgenden Ermittlung der Prozesskette mittels Pathfinding-Algorithmus ist dem Kognitiven Assistenzsystem der Nacharbeitsplan bereits im Voraus bekannt. Daher gibt das Assistenzsystem einen Prozess vor und nimmt gegenüber dem Mitarbeiter eine Erwartungshaltung ein. Mit der Umsetzung des Montageprozesses erfüllt der Montagemitarbeiter die Vorgaben des Assistenzsystems. Durch die Prüfung des Montageprozesses erkennt das Assistenzsystem die sachgemäße Ausführung des Prozesses und schließt daraus auf den realen Produktzustand. Die zu überprüfenden Produkt- bzw. Prozessmerkmale sind ebenfalls im Vorfeld bekannt. Nachdem der Prozess erfolgreich abgeprüft wurde, wird nach der Dokumentation der nächste Nacharbeitsprozess angezeigt, wobei zeitgleich die Erfüllung der Montageprozessanforderung erwartet wird. Dieser Ablauf wird so lange wiederholt, bis ein Abbruchkriterium, in der Regel der Abschluss eines Auftrags z. B. in Form der Produktvervollständigung vorliegt.

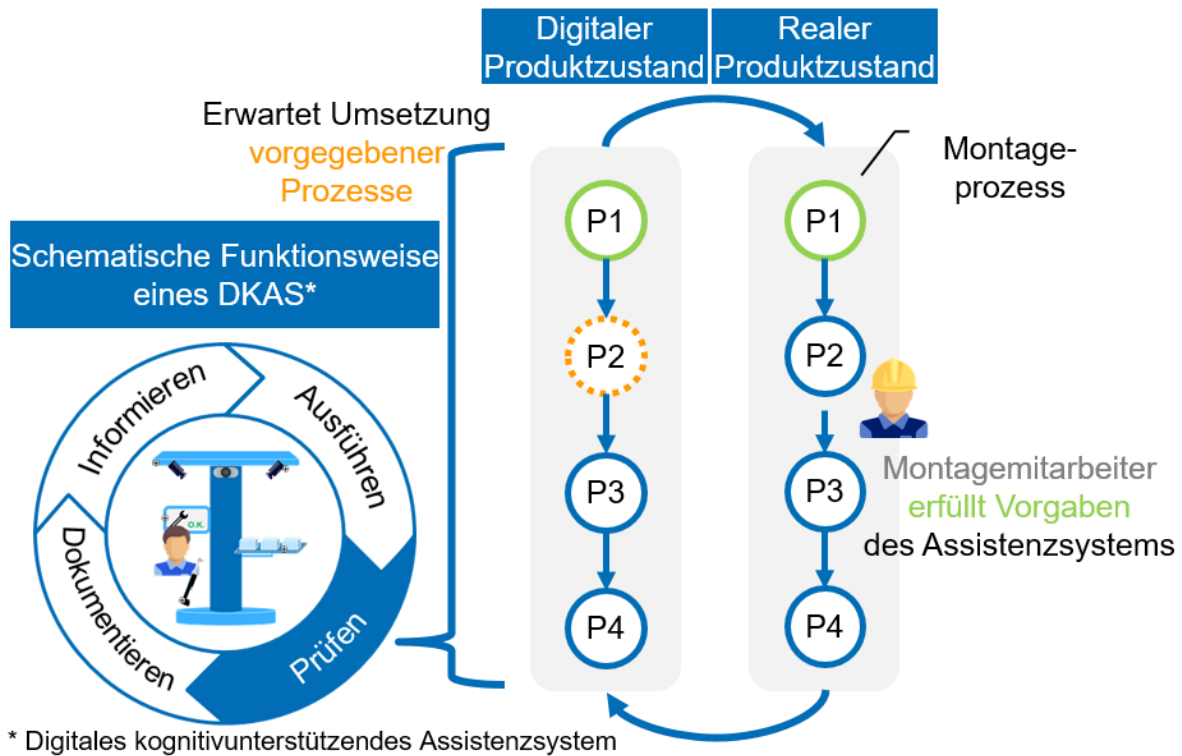


Abbildung 51: Schematische Funktionsweise herkömmlicher Digitaler Kognitiver Assistenzsysteme [Eigene Darstellung]

Ein Defizit herkömmlicher Kognitiver Assistenzsysteme ist die unzulängliche Synchronisationssicherheit zwischen realem und digitalem Produktzustand. Da Assistenzsysteme oftmals in der Serienmontage eingesetzt werden, fällt dieses Problem nicht stark ins Gewicht, da der Mitarbeiter ohnehin in seinen montagebezogenen Freiheiten eingeschränkt ist. Dies wird beispielsweise umgesetzt, indem ihm die entsprechenden Betriebsmittel erst gar nicht zur Verfügung stehen oder indem ein Werkstückträger die Handhabung des Produktes eingrenzt. Anders verhält es sich an einem manuellen Arbeitsplatz, der wie eine Manufaktur organisiert ist. Prinzipiell stehen dem Mitarbeiter alle Betriebsmittel zur Verfügung, die dieser zur Zerlegung oder zum Zusammenbau des Produktes benötigt. Dies bedeutet aber auch, dass der Mitarbeiter alle Montageprozesse zu jeder Zeit durchführen kann, vorausgesetzt, dass das Produkt den Montageprozess zulässt. Ein Problem ergibt sich bspw., wenn ein Mitarbeiter den verlangten Montageprozess durchführt, das Assistenzsystem den Prozess erkennt und als abgeschlossen markiert, der Mitarbeiter den Montageprozess aber wieder rückgängig macht, das Assistenzsystem dies jedoch nicht abprüft oder erkennt. In diesem Fall würde das Assistenzsystem die Ausführung des übernächsten Montageprozesses verlangen, ohne den Mitarbeiter darauf hinzuweisen, dass er den letzten Prozess (der jetzt rückgängig gemacht wurde) wiederholen muss. In dem Moment sind der reale Produktzustand und der digitale Produktzustand nicht mehr synchron. Zudem erlaubt ein Produkt mehrere Wege der Montage sowie der Demontage. Auch hierbei können

der digitale und reale Produktzustand asynchron werden. Zudem sind manche Sensoren im Vergleich zu anderen Sensoren weniger aussagekräftig, da diese nur wenige Merkmale überprüfen können.

Da die Nacharbeit einen hohen Anteil an manuellen Prozessen aufweist, kann der Mitarbeiter vom vorgesehenen Nacharbeitsplan abweichen. Zu diesem Zeitpunkt stimmt der tatsächliche, physische Produktzustand nicht mit dem angenommenen, digitalen Produktzustand überein. Daher wird in dieser Entwicklungsstufe das Assistenzsystem um die Fähigkeit ergänzt, aufgrund von Beobachtungen den physischen Produktzustand einschätzen zu können. Hierfür ist die erneute Erweiterung des dem Assistenzsystem zugrundeliegenden Graphen notwendig.

6.4.1 Stand der Technik

In diesem Kapitel werden Methoden zur Einschätzung des Zustandes auf Basis von Markov-Modellen vorgestellt. Markov-Modelle eignen sich zur Abbildung von wahrscheinlichkeitsverteilten Zustandsänderungen. Da das entwickelte Assistenzsystem ebenfalls unter Unsicherheiten agiert, stellen sie ein wichtiges Werkzeug dar und werden im Folgenden grundlegend beschrieben.

Markov-Eigenschaft

Markov-Modelle eignen sich zur Abbildung stochastischer Prozesse, die die Markov-Eigenschaft besitzen. Die Markov-Eigenschaft wird als Gedächtnislosigkeit eines stochastischen Prozesses definiert. Hierbei wird davon ausgegangen, dass die Wahrscheinlichkeit für einen Zustandswechsel nicht von der Geschichte bisheriger Zustände abhängt. [92 S.568, 108 S.265, 138 S.89, 139 S.42, 140 S.11f.]

Ein anschauliches Beispiel ist der *Wurf einer Münze*: Ungeachtet der Anzahl der Zufallsexperimente ist jeder Wurf von den vorherigen Ergebnissen unabhängig. Hingegen weist der Prozess: *Ziehung von Kugeln aus einer Urne ohne zurücklegen* keine Markov-Eigenschaft auf, da sich aus der Kenntnis bisheriger Ziehungen und der Kenntnis über die Kugel insgesamt die Wahrscheinlichkeiten für die Ziehung eines Ergebnisses ändern.

Markov-Modelle

Markov-Modelle eignen sich zur Modellierung von Zustandsmodellen mit stochastischen Übergängen [141 S.219]. Hierzu existiert eine Vielzahl von Modellen. Die häufigsten Modelle sind die Markov-Kette (MC: Markov Chain), das Hidden-Markov-Modell (HMM: Hidden Markov Model), das Markov-Entscheidungsproblem (MDP: Markov Decision Problem/ Process) und teilweise beobachtbare Markov-Entscheidungsprobleme (POMDP: Partially Observable Markov Decision Problem/ Process). [142 S.1]

Abhängig von den gegebenen Umständen erweist sich ein Modell als geeigneter als ein Anderes. Tabelle 5 stellt den Zusammenhang häufig verwendeter Markov-Modelle

nach Michael Littman dar. [143 S.352] Im Wesentlichen werden zwei Eigenschaften betrachtet:

1. Erlaubt das zu modellierende System die direkte Beobachtung der Zustände?
2. Erlaubt das Modell die Steuerung der Übergänge zwischen zwei Zuständen?

Tabelle 5: Häufig verwendete Markov-Modelle sowie deren Eigenschaften hinsichtlich Beobachtbarkeit und Steuerbarkeit angelehnt an Michael Littman [144]

Markov-Modelle		Können Transitionen beeinflusst werden?	
		Nein	Ja
Sind alle Zustände vollständig beobachtbar?	Ja	Markov Chain	MDP
	Nein	HMM	POMDP

Die einfachste Form der Markov-Modelle bilden die Markov-Ketten. Diese können am deutlichsten durch einen Graphen dargestellt werden. Der Übergang zwischen zwei Knoten (von s nach s') ist nicht durch eine Steuerung (Agent) beeinflussbar. Vielmehr existieren Übergangswahrscheinlichkeiten $P(s, s')$.

Ein MDP wird ähnlich wie eine Markov-Kette modelliert, jedoch werden die Übergänge durch die Auswahl einer Aktion durch einen Agenten mitbestimmt. Diesem Agenten steht hierbei eine Menge an Aktionen $a \in A$ zur Verfügung. Der Entscheidungsprozess besteht in der Ermittlung der auszuwählenden Aktion, die durch unterschiedliche Algorithmen erfolgen kann. Zwar führen Aktionen zu einem Übergang, allerdings nur mit einer bestimmten Wahrscheinlichkeit $P(s'|s, a)$. s' bezeichnet hier einen nachfolgenden Zustand. Zudem enthält das Modell eine Rewardfunktion $R(s)$ bzw. $R(s, a, s')$, die zur Ermittlung der Aktion herangezogen werden. [92 S.647, 145 S.14]

Zur Erfassung eines Zustands ist ein Sensor notwendig, dessen Zuverlässigkeit variieren kann. Soll dieser Umstand berücksichtigt werden, eignet sich die Modellierung durch Partially Observable Markov Decision Processes (POMDP). Dieses Modell besteht ähnlich wie das MDP aus einer Menge an Zuständen $s \in S$ sowie einer Menge von Aktionen $a \in A$. Desweiteren gibt es Übergangswahrscheinlichkeiten $T(s'|s, a)$ sowie eine Beobachtungswahrscheinlichkeit $O(z|s', a)$, wobei $z \in Z$ eine konkrete Beobachtung aus einer Menge von Beobachtungen darstellt. Eine konkrete Beobachtung kann sein: „Die Ampel ist grün“, „Das Bauteil ist vorhanden“, „Der Prozess ist o.k.“. Auch im POMDP-Modell werden Rewards in der Form $R(s')$ oder $R(s, a, s')$ vergeben. Da die Rewards keine Wahrscheinlichkeit ausdrücken, werden diese nicht wie die bedingten Wahrscheinlichkeiten notiert, sondern in Form einer Funktion. Da im POMDP die Zustände nicht direkt beobachtet werden (können), wird lediglich auf eine gewisse Aufenthaltswahrscheinlichkeit geschlossen. Der Belief State B beschreibt die Aufenthaltswahrscheinlichkeitsverteilung über alle Zustände. Dieser wird oftmals in der Form

$B^{1 \times |S|}$ angegeben. Hauptsächlich wird beim POMDP der Belief State aktualisiert. Darauf basierend können die nächsten Aktionen abgeleitet werden.

Tabelle 6 zeigt eine Übersicht der Parameter im POMDP. [Vgl. 146 S.119] [Vgl. 92 S.658ff.]

Tabelle 6: Parameter eines POMDP-Modells

Parameter	Symbol	Erläuterung
Zustand (States)	$s \in S$	Alle Zustände des Markov-Modells
Nachfolgezustand	$s' \in S$	Bei der Betrachtung eines Nachfolgezustands genutzte Notation
Übergangswahrscheinlichkeit (Transition Probability)	$T = P(s' s, a)$	Wahrscheinlichkeit, dass s' eingenommen wird, gegeben, dass s der Ausgangszustand war und die Aktion a durchgeführt wurde
Beobachtungswahrscheinlichkeit (Observation Probability)	$O = P(z s, a)$	Die Wahrscheinlichkeit, eine bestimmte Beobachtung zu tätigen, wenn eine Aktion ausgeführt wurde und der Ausgangszustand mit einer gewissen Wahrscheinlichkeit bekannt war
Reward	$R(s, a, s')$	Modellparameter, der die Begehrlichkeit eines Zustands beziffert
Beobachtung (Observation)	$z \in Z$	Die Beobachtung eines Sensors
Belief State	B	Wahrscheinlichkeitsverteilung über alle Zustände

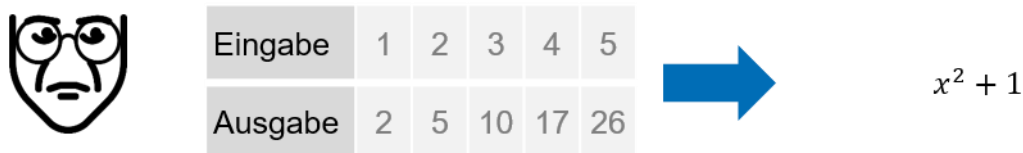
Lernende Systeme

Das zu entwickelnde Assistenzsystem besitzt ebenfalls eine lernende Komponente. Auf der „Plattform Lernende Systeme - Die Plattform für Künstliche Intelligenz“ werden Lernende Systeme wie folgt beschrieben: „Lernende Systeme sind Maschinen, Roboter und Softwaresysteme, die abstrakt beschriebene Aufgaben auf Basis von Daten, die ihnen als Lerngrundlage dienen, selbstständig erledigen, ohne dass jeder Schritt spezifisch programmiert wird.“ [147]

Beschränkt man sich bei der Definition von Lernenden Systemen auf Systeme aus dem Bereich der Künstlichen Intelligenz, wie die Plattform Lernenden Systeme suggeriert, fallen hierunter insbesondere Systeme aus dem Bereich des Maschinellen Lernens. Verfahren aus dem Bereich des Maschinellen Lernens lassen sich meist in drei

Kategorien einordnen: Überwachtes Lernen (Supervised Learning), Unüberwachtes Lernen (Unsupervised Learning) und Bestärkendes Lernen (Reinforcement Learning). Abbildung 52 stellt die drei Klassen aus dem Bereich des Maschinellen Lernens intuitiv dar. Im Überwachten Lernen bestehen die Trainingsdaten aus Eingabe-Ausgabe Paaren (input- output pairs) [92 S.695], wobei die Algorithmen einen Zusammenhang zwischen den Eingaben und Ausgaben herstellen. Im gegebenen Beispiel aus Abbildung 52a werden die Eingabedaten und Ausgabedaten dargestellt. Der Leser kann durch gedankliche Versuche den Zusammenhang ableiten. Beim Unüberwachten Lernen bestehen die Trainingsdaten aus einer Menge von Daten, in denen Zusammenhänge hergestellt werden. Allerdings stehen hierfür keine Ausgabedaten zur Verfügung, siehe Abbildung 52b. Das Bestärkende Lernen agiert innerhalb eines bekannten oder unbekanntes Systems. Dabei werden Aktionen durchgeführt, deren jeweiliges Ergebnis bewertet wird. In Abbildung 52c wird bspw. eine Zielscheibe anvisiert, wobei die Einschussstelle einer Bewertung unterzogen wird. Je nach Ergebnis wird (in einer gleichen oder ähnlichen Situation) die Aktion korrigiert.

a) Überwachtes Lernen



b) Unüberwachtes Lernen



c) Bestärkendes Lernen



Abbildung 52: Intuitive Betrachtung der drei Probleme und Algorithmen im Maschinellen Lernen [Eigene Darstellung]

In der Regel kann Überwachtes Lernen im Vergleich zum Unüberwachten Lernen als kostenintensiver betrachtet werden. Grund hierfür ist die Bereitstellung der Labels für die Trainingsdaten. Labels sind das bereits bekannte Ergebnis zu den Eingangsdaten. Dazu müssen die Eingangsdaten einer Vorabbewertung unterzogen werden. Hingegen reichen für das Unüberwachte Lernen die Eingangsdaten aus. Diese liegen relativ häufig vor und bedürfen abgesehen von einigen Bereinigungsprozessen keiner weiteren Evaluierung.

Lernende Assistenzsysteme mit Markov-Modellen

Für die Entwicklung eines Lernfähigen Assistenzsystems werden Markov-Modelle verwendet, die bereits einführend beschrieben wurden. Allerdings wurden diese für Entscheidungsprozesse genutzt. Da hierfür ein grafisches Wahrscheinlichkeitsmodell notwendig ist, müssen die zugehörigen Modellparameter ermittelt werden. Hierzu können Algorithmen verwendet werden, die insbesondere für Hidden-Markov-Modelle (HMM) genutzt werden. Hidden-Markov Modelle beschreiben Systeme mit verborgenen Zuständen (engl.: Hidden States). Allein aufgrund der Beobachtung sowie der Beobachtungssequenzen wird auf die Zustände geschlossen. Hierzu sind Übergangs- bzw. Transitionsmatrizen/-wahrscheinlichkeiten notwendig. Zudem müssen auch Wahrscheinlichkeiten über die Beobachtungen in bestimmten Zuständen bekannt sein. Abbildung 53 zeigt ein oft gewähltes Beispiel mit dem Wetter als Hidden-State (Sonnig (S), Bewölkt (B), Regnerisch (R)) und das Gemüt als Beobachtung (Fröhlich (F), Traurig (T)). [148, Vgl. 149 S.88] Eine Erläuterung hierzu lautet: Ein Gefangener kann das Wetter nicht unmittelbar sehen, hingegen aber den Gemütszustand des Wärters beobachten. Daher versucht er anhand dieser Beobachtung auf das Wetter zu schließen. Das in Abbildung 53 beschriebene Modell zeigt sowohl die Wahrscheinlichkeiten für den Übergang von einem (Wetter-)Zustand in den nächsten (Wetter-)Zustand als auch die Beobachtungswahrscheinlichkeit eines bestimmten Gemüts in einem (Wetter-)Zustand.

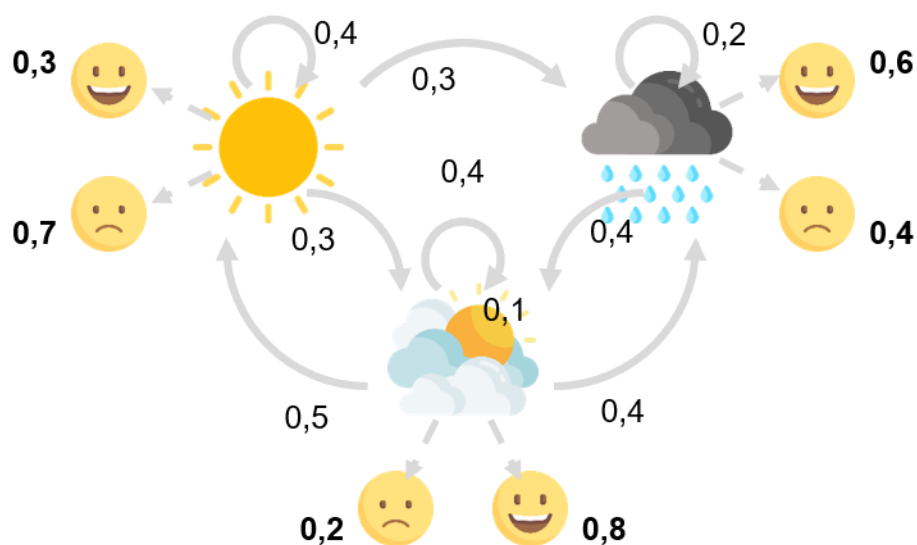


Abbildung 53: Beispiel eines Hidden-Markov-Modells mit Übergangswahrscheinlichkeit für das Wetter und wahrscheinlicher Gemütszustände als Beobachtung. [Eigene Darstellung nach gängigen Beispielen]

Die folgende Beschreibung des HMM sowie der zugehörigen Probleme und Algorithmen in den Kapiteln 6.4.1 und 6.4.1.1.2 beruht auf der Arbeit „A tutorial on hidden Markov models and selected applications in speech recognition“ von Rabiner [150]. Rabiner nennt für das HMM drei fundamentale Probleme [150 S.261]:

1. Wie hoch ist die Wahrscheinlichkeit für eine Sequenz von Beobachtungen $O = o_1, o_2, \dots, o_T$, wenn die Beobachtungssequenz sowie die Modellparameter λ des HMM gegeben sind?

$$P(O|\lambda)?$$

2. Wie lautet die wahrscheinlichste Zustandssequenz Q , wenn die Beobachtungssequenz O sowie die Modellparameter λ des HMM bekannt sind? Bzw. welche Sequenz von Zuständen erklärt am besten die Sequenz der Beobachtungen innerhalb des HMM?

$$Q = q_1, q_2, \dots, q_T?$$

3. Welche Modellparameter $\lambda = (A, B, \pi)$ beschreiben mit einer hohen Genauigkeit die Beobachtungen?

O bezeichnet eine Folge von Beobachtungen mit den (Beobachtungs-) Elementen (in der Folge) $O = (o_1, o_2, \dots, o_T)$, wie diese beispielhaft in Abbildung 54 zu sehen sind.

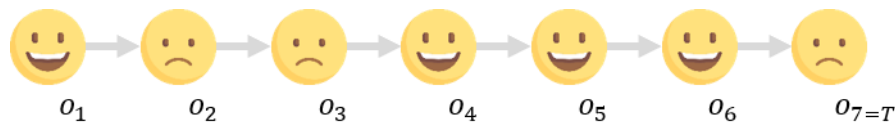


Abbildung 54: Beispiel einer Folge von Beobachtungen [Eigene Darstellung]

Q bezeichnet eine Folge von Zuständen mit den Elementen $Q = (q_1, q_2, \dots, q_T)$, die in Abbildung 55 beispielhaft dargestellt werden.

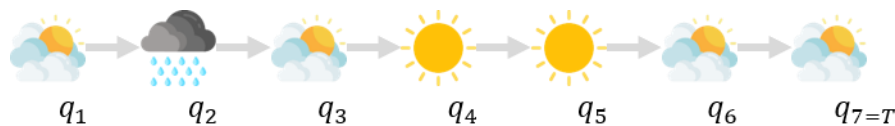


Abbildung 55: Beispiel einer Folge von Hidden States (versteckten Zuständen) [Eigene Darstellung]

λ bezeichnet die Menge der Modellparameter des HMM, bestehend aus der Transitionsmatrix $A \in \mathbb{R}^{|S| \times |S|}$, der Beobachtungsmatrix $B \in \mathbb{R}^{|S| \times |V|}$ sowie der Initialverteilung bzw. -wahrscheinlichkeit π . S ist die Menge der Zustände. $|S| = N$ ist die Anzahl der Elemente in S , also die Anzahl der Zustände. V ist die Menge der Beobachtungen. An dieser Stelle sei angemerkt, dass sich in der Beschreibung unterschiedlicher Markov-Modelle für gleiche Parameter unterschiedliche Abkürzungen durchgesetzt haben. Diese sind zum Teil auf bekannte Schriften zurückzuführen, bspw. werden für POMDPs als Symbol für Transitionsmatrizen T und für die Beobachtungsmatrix O verwendet, wie bspw. in [146] zu sehen ist.

Im Folgenden werden bezugnehmend auf die formulierten Probleme 1 und 3 unterschiedliche Methoden zu deren Lösung präsentiert. Der Bezug zum Assistenzsystem besteht in der Fähigkeit, dessen Beobachtungen einzuschätzen und andererseits die Modellparameter aus den Beobachtungen erlernen zu können. Dabei wird die Annahme getroffen, dass ein Zustand lediglich eine Beobachtung zulässt und somit die Sequenzen der Hidden States und der Beobachtungen die gleiche Länge aufweisen.

6.4.1.1.1 Forward-Algorithmus

Zur Klärung der Notwendigkeit des Forward-Algorithmus wird zur Ermittlung der Wahrscheinlichkeit für eine Beobachtungssequenz $P(O|\lambda)$ zunächst ein naiver Ansatz verfolgt.

Gleichung (12) zeigt die Ermittlung der Beobachtungswahrscheinlichkeit aus $P(O|\lambda)$, also der Wahrscheinlichkeit einer Beobachtungssequenz bei gegebenen Parametern des HMM. Dabei wird für jede Permutation einer Zustandsabfolge die Wahrscheinlichkeit für eine konkrete Beobachtung mit der jeweiligen Wahrscheinlichkeit der Zustandsabfolge multipliziert und über diese Zustandspermutationen summiert. Allerdings erweist sich dieser Ansatz als unpraktisch, da die Summe über Q , N^T Summanden in Betracht zieht. Dabei ist N die Anzahl der Zustände, die das HMM beinhaltet und T die Länge einer Sequenz. Im Beispiel aus Abbildung 54 bzw. Abbildung 55 wären das bereits 2187 Summanden und bei einer Folge von 8 Elementen bereits 6.561 Summanden bzw. bei 4 Zuständen bereits 16.384 Summanden.

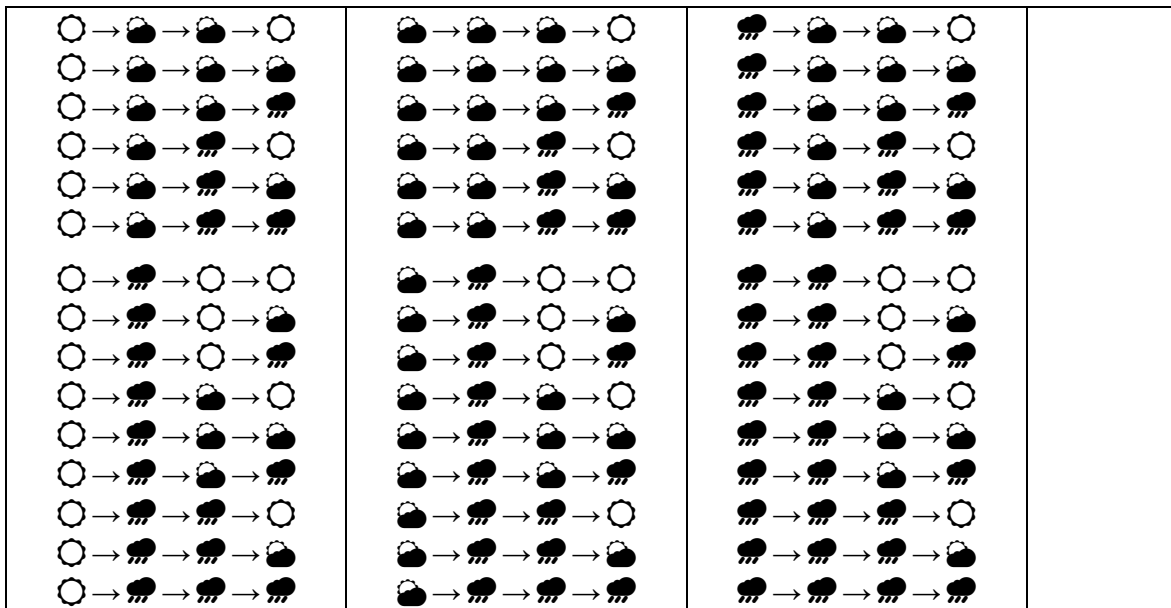
$$P(O|\lambda) = \sum_{\text{für alle } Q} P(O|Q, \lambda) \times P(Q|\lambda) \tag{12}$$

Ein Beispiel für die Anwendung von Gleichung (12) wird anhand von Tabelle 7 demonstriert. Hierbei steht eine Sequenz aus vier Beobachtungen (Fröhlich, Fröhlich, Fröhlich, Traurig) zur Verfügung. Bei einer Sequenzlänge von 4 und drei verfügbaren Hidden States (Sonnig, Bewölkt, Regnerisch) sind $3^4 = 81$ Zustandssequenzen möglich. Diese Sequenzen können alle mit einer gewissen Wahrscheinlichkeit zur benannten Beobachtung führen. Beispielsweise erklärt die Abfolge (Sonnig, Sonnig, Sonnig, Sonnig) die Beobachtung (Fröhlich, Fröhlich, Fröhlich, Traurig) mit der Wahrscheinlichkeit:

$$P(F, F, F, T|S, S, S, S) = P(F|S) \times P(F|S) \times P(F|S) \times P(T|S)$$

Tabelle 7: Bei fixer Beobachtungssequenz sind bei vier Hidden States 81 Permutationen der Zustandssequenz möglich

→ → →			O
→ → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → →	→ → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → →	→ → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → → →	Q



Die Gesamtwahrscheinlichkeit setzt sich somit aus 81 Summanden zusammen.

$$P(O) = P(F, F, F, T | S, S, S, S) + P(F, F, F, T | S, S, S, B) + \dots + P(F, F, F, T | R, R, R, R)$$

Die Berechnung der Wahrscheinlichkeit $P(O)$ kann durch einen dynamisch-programmierbaren Algorithmus im Aufwand reduziert werden. Der Forward-Algorithmus ist ein dynamisch-programmierbares Verfahren zur Ermittlung der Wahrscheinlichkeit $P(O|\lambda)$. Abbildung 56 zeigt die Möglichkeiten der Zusammenstellung der Sequenz für die Hidden States. Beispielsweise ist die Wahrscheinlichkeit zum Zeitpunkt $t = T = 4$ im Hidden State „sonnig“ befindlich zu sein, die Summe aus den Wahrscheinlichkeiten zum Zeitpunkt $t = 3$ in den Zuständen „sonnig“, „bewölkt“ oder „regnerisch“ zu sein und der jeweiligen Transitionswahrscheinlichkeit zum Zustand „sonnig“. Diese Zustände setzen sich wiederum aus den jeweiligen Summen der vorhergehenden Zustandswahrscheinlichkeiten zusammen. Dabei wird zu jeder Zeit auch die zugehörige Beobachtung (fröhlich, traurig) berücksichtigt. Diese Betrachtungsweise reduziert die Komplexität von N^T auf N^2T , wie an der Anzahl der Pfeile in Abbildung 56 zu erkennen ist.

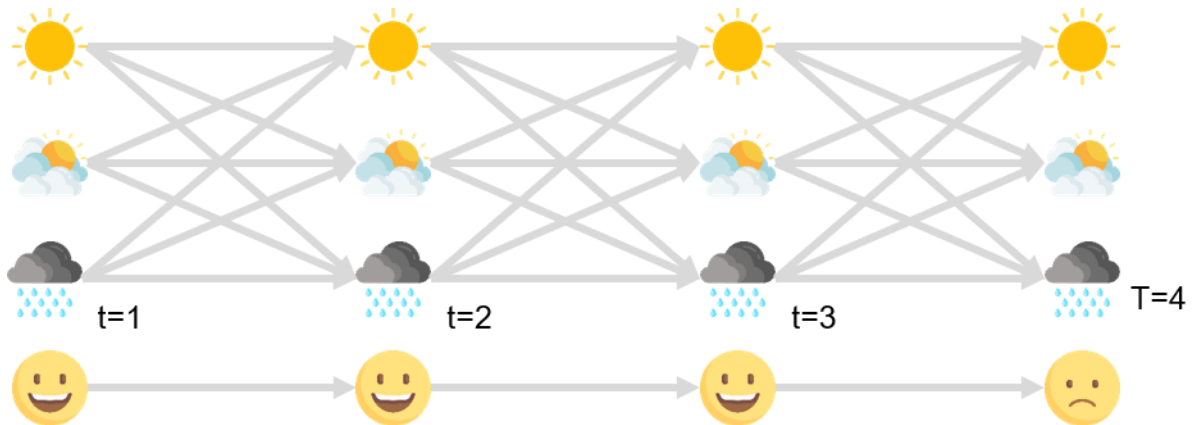


Abbildung 56: Permutative Zusammenstellungsmöglichkeiten der Hidden States bei einer Sequenzlänge von T=4. [Eigene Darstellung entlehnt aus [150 S. 262]]

Gleichung (13) zeigt die Definition von $\alpha_t(j)$, die die Wahrscheinlichkeit einer Beobachtungssequenz mit Hidden State j zum Zeitpunkt t bei gegebenen Modellparametern λ angibt. Dabei bezeichnet t die Zeit, bis zu der die Beobachtung gemacht wurde und j den t -ten Zustand. [151 S.93] [150 S.262]

$$\alpha_t(j) = P(o_1, o_2 \dots o_t, q_t = S_j | \lambda) \quad (13)$$

Die Wahrscheinlichkeit kann als Summe über alle (N) Hidden States gebildet werden. Der Faktor $\alpha_{t-1}(i)$ in Gleichung (14) ist die Zustandswahrscheinlichkeit für den Vorzustand samt Beobachtungssequenz bis zum Zeitpunkt $t - 1$. Hieraus wird ersichtlich, dass die Gleichung eine rekursive Funktion darstellt, die für die dynamische Programmierung ausgelegt ist. Der Faktor a_{ij} gibt die Wahrscheinlichkeit für den Zustandswechsel i nach j an. Dabei ist i nicht fix, sondern ebenfalls die Laufvariable. Letztendlich werden hierdurch alle Zustände, die in den Zustand j münden berücksichtigt. $b_j(o_t)$ ist die Beobachtungswahrscheinlichkeit von o_t im Zustand j .

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t) \quad (14)$$

Auf das Beispiel aus Abbildung 56 angewendet, berechnen sich die Anteile der Gleichung wie folgt:

$$\alpha_4(S) = \alpha_3(S) a_{SS} b_S(T) + \alpha_3(B) a_{BS} b_S(T) + \alpha_3(R) a_{RS} b_S(T)$$

$$\alpha_4(B) = \alpha_3(S) a_{SB} b_B(T) + \alpha_3(B) a_{BB} b_B(T) + \alpha_3(R) a_{RB} b_B(T)$$

$$\alpha_4(R) = \alpha_3(S) a_{SR} b_R(T) + \alpha_3(B) a_{BR} b_R(T) + \alpha_3(R) a_{RR} b_R(T)$$

$$\alpha_3(S) = \alpha_2(S) a_{SS} b_S(F) + \alpha_2(B) a_{BS} b_S(F) + \alpha_2(R) a_{RS} b_S(F)$$

$$\alpha_3(B) = \alpha_2(S) a_{SB} b_B(F) + \alpha_2(B) a_{BB} b_B(F) + \alpha_2(R) a_{RB} b_B(F)$$

$$\alpha_3(R) = \alpha_2(S) a_{SR} b_R(F) + \alpha_2(B) a_{BR} b_R(F) + \alpha_2(R) a_{RR} b_R(F)$$

$$\alpha_2(S) = \alpha_1(S)a_{SS}b_S(F) + \alpha_1(B)a_{BS}b_S(F) + \alpha_1(R)a_{RS}b_S(F)$$

$$\alpha_2(B) = \alpha_1(S)a_{SB}b_B(F) + \alpha_1(B)a_{BB}b_B(F) + \alpha_1(R)a_{RB}b_B(F)$$

$$\alpha_2(R) = \alpha_1(S)a_{SR}b_S(F) + \alpha_1(B)a_{BR}b_S(F) + \alpha_1(R)a_{RR}b_S(F)$$

$$\alpha_1(S) = P(S|Start)P(F|S)$$

$$\alpha_1(B) = P(B|Start)P(F|B)$$

$$\alpha_1(R) = P(R|Start)P(F|R)$$

Bestimmte Faktoren finden hierbei rekursive Verwendung, was den Rechenaufwand reduziert.

Abbildung 57 zeigt die Matrizen A und B zur Interpretation der Gleichungen aus dem Beispiel. A bezeichnet hierbei eine Matrix mit den Übergangswahrscheinlichkeiten von einem Zustand in den nächsten Zustand. B ist die Wahrscheinlichkeit einer Beobachtung in einem Zustand.









	A			B	
	 (S)	 (B)	 (R)	 (F)	 (T)
	$P(S S)$	$P(B S)$	$P(R S)$	$P(F S)$	$P(T S)$
	$P(S B)$	$P(B B)$	$P(R B)$	$P(F B)$	$P(T B)$
	$P(S R)$	$P(B R)$	$P(R R)$	$P(F R)$	$P(T R)$

Abbildung 57: Wahrscheinlichkeiten für Zustandswechsel (A) und Beobachtungswahrscheinlichkeiten (B). [Eigene Darstellung]

Bisher wurde lediglich die Wahrscheinlichkeit einer Beobachtung bestimmt. Diese Vorarbeit dient zur Ermittlung und Schätzung der Parameter λ des Markov-Modells mittels Baum-Welch-Algorithmus.

6.4.1.1.2 Forward-Backward-Algorithmus

Mithilfe des Forward-Backward-Algorithmus wird die Wahrscheinlichkeit für eine Beobachtungssequenz ermittelt, wobei die Parameter des Markov-Modells $\lambda = \pi, A, B$ bekannt sein müssen. [Vgl. 152 S.114]

Der Rückwärtsanteil (Backward) β des Forward-Backward-Algorithmus zieht die Wahrscheinlichkeit einer bestimmten Beobachtungssequenz aus dem Zustand S_i heraus zum Zeitpunkt t in Betracht. Er gibt die Wahrscheinlichkeit an, im Zustand S_i zu sein, wenn die nachfolgenden Beobachtungen bekannt sind. [150 S.263, 151 S.93] Anders formuliert gibt β an, wie wahrscheinlich aus einem Zustand heraus die nachfolgende Beobachtungssequenz ist.

β berechnet sich mit der folgenden Gleichung:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = S_i, \lambda) \quad (15)$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad (16)$$

α gibt für eine (vorausgehende) Beobachtungssequenz die Wahrscheinlichkeit an, in einem gegebenen Zustand zu landen. β hingegen gibt für eine (nachfolgende) Beobachtungssequenz die Wahrscheinlichkeit an, von einem gegebenen Zustand (zum Zeitpunkt t) gestartet zu sein.

Bei gegebener Beobachtungssequenz O sowie dem HMM-Modellparameter λ wird die Übergangswahrscheinlichkeit von $i \rightarrow j$ mit $\xi(i, j)$ angegeben.

Es gilt:

$$\xi(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda) \quad (17)$$

$$\xi(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \quad (18)$$

ξ bezeichnet die zu erwartende Wahrscheinlichkeit, aus dem Zustand i in einen anderen/nächsten Zustand überzugehen.

γ bezeichnet die Wahrscheinlichkeit für einen Zustand zum Zeitpunkt t bei gegebener Beobachtungssequenz sowie den gegebenen Parametern λ .

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) \quad (19)$$

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (20)$$

Zur Ermittlung der Transitionsmatrix A sowie der Beobachtungsmatrix B wird der Expectation-Maximization-Algorithmus genutzt.

6.4.1.1.3 Expectation Maximization

Der Expectation-Maximization-Algorithmus (EM-Algorithmus) ist ein iteratives Verfahren zur Neuermittlung der HMM-Parameter A, B, π . Dieses zweistufige Verfahren (E-Step/ M-Step) berechnet die Elemente der Matrizen A und B iterativ (bis eine Konvergenz erkannt wird). Das hier beschriebene Verfahren ist speziell auf die Behandlung von HMM ausgelegt und bedient sich des Forward-Backward-Algorithmus. Er ist unter der Bezeichnung Baum-Welch-Algorithmus bekannt [Vgl. 153 S.114, 124].

Der **E-Step** besteht in diesem Fall aus der Ermittlung von $\xi(i, j)$ und $\gamma_t(j)$ nach Gleichung (18) und (20).

Der **M-Step** errechnet die neuen Werte \hat{a}_{ij} und $\hat{b}_j(v_k)$. Gleichung (21) zeigt die Definition der Übergangswahrscheinlichkeit. Hierbei wird der aktualisierte Wert der Übergangswahrscheinlichkeit \hat{a}_{ij} durch das Verhältnis zwischen der Anzahl einer bestimmten Transition i nach j mit der Anzahl aller Transitionen aus i heraus ins Verhältnis gesetzt. Für die aktualisierte Übergangswahrscheinlichkeit von $i \rightarrow j$ gilt:

$$\hat{a}_{ij} = \frac{\text{Erwartete Anzahl der Transitionen } i \rightarrow j}{\text{Erwartete Anzahl von Transitionen aus } i}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi(i, j)}{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi(i, k)} \quad (21)$$

Für die aktualisierte Beobachtungswahrscheinlichkeit gilt:

$$\hat{b}_j(v_k) = \frac{\sum_{t=1}^T o_{t=v_k} \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (22)$$

Da im vorgestellten Verfahren lediglich lokale Optima ermittelt werden können, ist die Wahl geeigneter Startwerte wichtig. [150 S.265, 154 S.2, 155 S.360]

6.4.2 Theoretische Umsetzung

In diesem Kapitel wird die Umsetzung der Zustandssynchronisation mittels des Belief State eines Markov-Modells beschrieben.

Ausgabe bestmöglicher Handlungsalternativen bei Unsicherheiten im Synchronisationsprozess

Zur Zustandssynchronisation werden Prozessprüfungen genutzt. Im vorgestellten Ansatz werden die Überwachung zur Zustandssynchronisation sowie die Prozessprüfung voneinander getrennt. Zur Prozess- und Produktzustandsprüfung werden die Sensoren herangezogen, die zur Merkmalsbestimmung am besten geeignet sind. Zur Zustandssynchronisation wird ein Modell entwickelt, das Rückschluss auf den realen Produktzustand erlaubt. Hierzu wird ein auf Markov basierter Ansatz gewählt. Das ausgewählte Markov-Modell nutzt die Belief-State-Funktion des **Partially Observable Markov Decision Process**. Die Komponente „Partielle Beobachtbarkeit“ ist dadurch gegeben, dass nicht jeder Sensor des Assistenzsystems zuverlässige Angaben über den Produktzustand machen kann. Beispielsweise kann eine Lichtschranke zur Eingriffskontrolle versehentlich verletzt werden, ohne dass hieraus die Annahme folgen muss, dass ein Bauteil aufgenommen wurde. Daraus ergibt sich, dass lediglich mit einer relativen Sicherheit auf den realen Produktzustand geschlossen werden kann. Im Falle von Markov-Modellen wird der Belief State in Form eines Vektors wiedergegeben. Dieser drückt die Sicherheit aus, dass ein bestimmter Zustand (wahrscheinlich) vorliegt. Das Markov-Modell kann als Graph beschrieben werden, dessen Knoten als Zustände s bezeichnet werden. Im vorliegenden Fall besteht die Grundstruktur des Graphen aus

dem in den vorangegangenen Kapiteln erarbeiteten Produktzustandsgraphen. Die Übergänge von einem Zustand in den nachfolgenden Zustand werden durch Aktionen verursacht, wobei die Aktionen u. a. Montageprozesse beinhalten. Zur Beurteilung des Zustands ist die Beobachtung z des Zustands notwendig. Die Beobachtungswahrscheinlichkeit z gibt die Wahrscheinlichkeit an, eine Beobachtung in einem bestimmten Zustand zu machen. Die Beobachtung wird durch Sensoren durchgeführt.

Zur Berechnung des Belief States ist neben dem Zustand s , der Aktion a sowie der Beobachtung ebenfalls die Übergangswahrscheinlichkeit $T = P(s'|s, a) = T(s'|s, a)$ notwendig. Diese gibt die Wahrscheinlichkeit an, von einem Produktzustand zum nächsten Produktzustand überzugehen. Zudem ist die Wahrscheinlichkeit z notwendig. Der Belief State ist ein Vektor, dessen Elemente mit je einem Zustand assoziiert sind. In der Summe ergeben die Einträge 1. Der Belief State lässt sich für einen Zustand wie folgt ermitteln [146 S.121].

$$b'(s') = \sum_{s \in S} \frac{b(s)T(s'|s, a) * O(z|a, s')}{p_z} \quad (23)$$

Dabei steht b' für den nächsten Belief State, der berechnet wird, nachdem eine Aktion durchgeführt und eine Beobachtung gemacht wurde. Die Wahrscheinlichkeit $b'(s')$ im Zustand s befindlich zu sein, ist abhängig von der vorherigen Aufenthaltswahrscheinlichkeit sowie der Wahrscheinlichkeit, von einem bestimmten Zustand s in einen bestimmten Nachfolgezustand s' überzugehen. Zudem ist die aktualisierte Aufenthaltswahrscheinlichkeit abhängig von der Wahrscheinlichkeit O , überhaupt die Beobachtung z zu machen. Tabelle 8 gibt einen Überblick über die Parameter des POMDP.

Tabelle 8: Markov-Parameter und ihre Übertragung in den Montageassistentenkontext

Markov Parameter	Formelzeichen	Bedeutung im Montage- und Nacharbeitskontext
Zustand (State)	S	Produktzustand und virtuelle Zustände
Aktion (Action)	A	Aktionen des Agenten, wie Fortsetzung des Montageprozesses oder Sonderaktionen
Beobachtung (Observation)	Z	Beobachtung/ Prüfung
Beobachtungswahrscheinlichkeit (Observation probability)	O	---
Übergangswahrscheinlichkeit (Transition probability)	T	Wahrscheinlichkeit, dass auf einen Produktzustand oder Prozess ein bestimmter Produktzustand oder Prozess folgt
Belief State	B	Wahrscheinlichkeiten des Produktzustands
Reward	R	---

Basierend auf den Belief State kann in der Entscheidungsfindung ein gewisses Risiko eingegangen werden. Der Reward-Parameter bestimmt, inwiefern das Risiko eingegangen werden kann.

Der Reward wird im vorliegenden Fall mit dem Belief State verknüpft und orientiert sich an der Streuung der Wahrscheinlichkeitsverteilung.

Zur Realisierung der 4. Entwicklungsstufe des Kognitiven Assistenzsystems sind die Aufnahme von Nacharbeitsprozessdaten sowie das Einlernen der Transitions- und Beobachtungswahrscheinlichkeiten notwendig. Daher muss die Befähigung zur 4. Entwicklungsstufe in zwei Phasen betrachtet werden. Die erste Phase ist die Lernphase und die zweite Phase ist die Betriebsphase. In der Lernphase wird das Assistenzsystem in der 1-3. Entwicklungsstufe betrieben. Dabei werden die Prozesse über die Prüfung aufgenommen. In der Betriebsphase werden die eingelernten Wahrscheinlichkeiten genutzt, um Aussagen über den Realzustand zu treffen und danach zu handeln. Die Beschreibung in diesem Kapitel folgt erneut dem Bottom-Up-Prinzip, d. h. es wird zunächst geschildert, wie das Assistenzsystem in der Betriebsphase funktioniert und danach wird das Lernen erläutert.

Grundlage für die Zustandssynchronisation ist ein Markov-Modell, das auf einen erweiterten Produktzustandsgraphen zurückgreift. Bei der Erweiterung des Produktzustandsgraphen werden virtuelle Zustände hinzugefügt. Grund hierfür ist, dass durch bestimmte Prozesse am Produkt keine Veränderungen eintreten. Beispielsweise wird bei den Prozessen „Herausnehmen des Antriebsrades“ das Antriebsrad aus dem Referenz-Produktzustand entnommen, aber mit dem nachfolgenden Nacharbeitsprozess „Antriebsrad Ablegen“ wird am Produktzustand keine Veränderung bewirkt. Eine Veränderung ist erst dann bemerkbar, wenn die gesamte Umgebung berücksichtigt wird, somit auch der Umstand, dass ein Zahnrad aufgenommen und danach abgelegt wurde. Zudem ist ein Zustand nur dann relevant, wenn dieser auch sensorisch erfasst werden kann. Im aufgeführten Beispiel ist die Entnahme des Zahnrades durch einen kamerabasierten Abgleich erfassbar. Der nachfolgende Zustand ist durch Positionstracking zu erfassen. Die virtuellen Zustände ergeben sich automatisch aus den definierten Nacharbeitsprozessen in den Kanten und werden automatisiert erstellt. Insofern gibt der Zustand an, welcher Nacharbeitsprozess gerade ausgeführt wurde. Somit folgt nach jedem Prozessschritt ein Knoten, der einen Zustand repräsentiert. Die diskrete Wahrscheinlichkeitsverteilung wird somit für jeden Zustandsknoten ausgegeben. Die Knoten werden zwar binär repräsentiert, aber zur besseren Nachvollziehbarkeit werden sie in Anlehnung an den zuvor ausgeführten Prozessen in der Zustandsform bezeichnet. In Anlehnung an die Prozessanweisung „Zahnrad entnehmen“ wird bspw. die Zustandsbeschreibung „Zahnrad entnommen“ gewählt. Abbildung 58 zeigt den Zusammenhang zwischen dem bisherigen Produktzustandsgraphen, der in den Kanten auch mehrere sequenzielle Prozessanweisungen enthalten kann und dem erweiterten Produktzustandsgraphen der auch Zustände erfasst, die nach einem Prozess folgen, der keine Änderung am Produkt bringt. Ein solcher Prozess ist bspw. die Ablage eines Bauteils, nach dem es aus dem Produkt demontiert wurde.

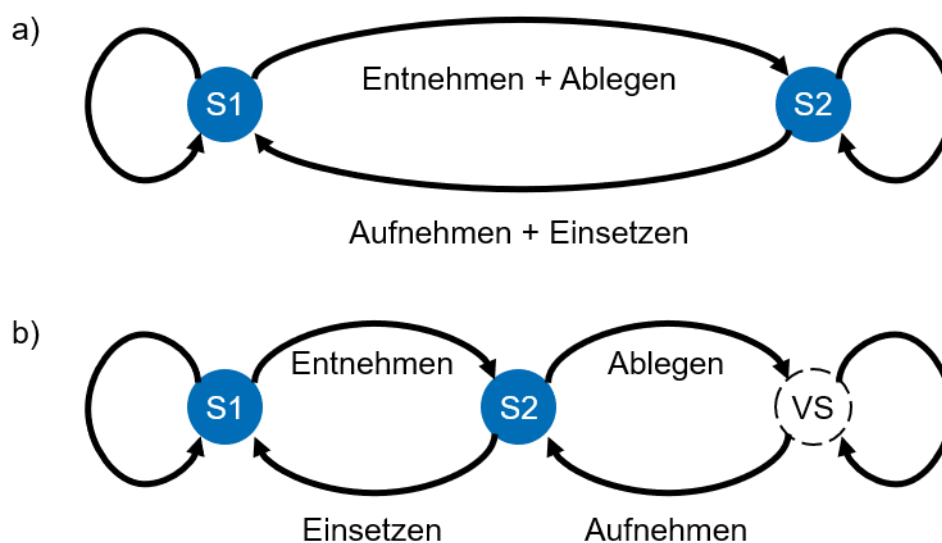


Abbildung 58: a) Produktzustandsgraph b) Erweiterter Produktzustandsgraph. [Eigene Darstellung]

Zur Erstellung des erweiterten Produktzustandsgraphen greift die Steuerung des Kognitiven Assistenzsystems auf die Graphendatenbank „ArangoDB“ zu und lädt den kompletten Produktzustandsgraphen. Hierbei wird jede einzelne Kante ausgelesen und für jeden weiteren Prozess in der Kante (nach dem ersten Prozess) wird ein virtueller Zustand angelegt. Zudem werden im Assistenzsystem die Transitionswahrscheinlichkeiten und die Beobachtungswahrscheinlichkeiten in Form einer Liste angelegt. Die Transitionswahrscheinlichkeiten ergeben sich zunächst nahezu gleichverteilt über alle Nachbarzustände. Die Beobachtungswahrscheinlichkeiten werden händisch angelegt. Das Kognitive Assistenzsystem greift während der Betriebsphase auf die Transitionswahrscheinlichkeiten und den Beobachtungswahrscheinlichkeiten zurück, die in den Kanten pro Prozess notiert sind. Das Kognitive Assistenzsystem wird um die POMDP-Funktionalität erweitert. Dabei werden Beobachtungen ausgewertet und ein Belief State wird ausgegeben. Der Belief State repräsentiert die normierte und diskrete Wahrscheinlichkeitsverteilung über alle Zustände. In dem entwickelten Assistenzsystem werden lediglich zwei Arten von Aktionen definiert. Zum einen werden alle Nacharbeitungsprozesse unter „Montage“ aggregiert, zum anderen wird die Aktion „Abfrage“ definiert, die das Kognitive Assistenzsystem zur Abfrage des Zustands über das Mitarbeiterführungssystem veranlasst.

Nach jeder Beobachtung wird der Belief State neu ermittelt. Der Belief State wird herangezogen, um die nächste bestmögliche Entscheidung zu fällen. Im vorliegenden Fall wird entschieden, ob die Montage weiterverfolgt wird oder ob eine Abfrage im Mitarbeiterführungssystem durchgeführt wird. Die Auswahl der Aktion erfolgt über die Berechnung der Standardabweichung über den Belief State.

Bevor die vierte Stufe des Kognitiven Assistenzsystem in den Produktivbetrieb gehen kann, müssen die Observations- und Transitionsmatrix trainiert werden. Dabei wird die Transitionsmatrix pro Fehlercode trainiert. Das Training geschieht mittels Anwendung des Baum-Welch-Algorithmus in der Trainingsphase. Hierfür werden über das Assistenzsystem bzw. den Betriebsmitteln die Beobachtungen aufgenommen. Jede Beobachtungssequenz wird zum Trainieren der Beobachtung und Transitionswahrscheinlichkeit genutzt. Eine Abschätzung über den angemessenen Trainingsgrad wird zum einen über die besuchten Kanten des erweiterten Produktzustandsgraphen ermittelt, zum anderen mittels Vorhersagen über den nächsten Zustand. Die Anzahl besuchter Knoten und Kanten im erweiterten Produktzustandsgraphen wird durch Dokumentation und Abspeicherung der Kanten und Knoten und Abgleich mit dem erweiterten Produktzustandsgraph ermittelt. Die Vorhersage der nächsten Kanten entspricht dem nächsten wahrscheinlichsten Zustand, der eintreten müsste und wird nach Ausgabe der Zustandsliste mittels Viterbi-Algorithmus über die Transitionsmatrix ermittelt. Erst wenn sowohl der Erweiterte Produktzustandsgraph in ausreichender Häufigkeit und Abdeckung beschrritten wurde und die Anzahl der Vorhersagen sich in der relativen Häufigkeit mit den Angaben in der Transitionsmatrix decken, wird die vierte Entwicklungsstufe freigegeben. Hierbei muss allerdings beachtet werden, dass beide Grö-

ßen lediglich eine Schätzung über den ausreichenden Trainingszustand zulassen. Zudem wird zur Vermeidung der Codierung der Häufigkeit von Fehlercodes in den Wahrscheinlichkeitsmatrizen jede Transitionsmatrix fehlercodespezifisch angelegt. Dennoch gibt es Überschneidungen in den Wahrscheinlichkeitsmatrizen für die Fehlercodes, da teilweise Fehler und zugehörige Nacharbeitspläne in großen Teilen die gleichen Prozesse und somit die gleichen Transitionen haben. Daher greifen auch Fehlerfälle, die noch nie zuvor trainiert wurden, auf Teile bereits bestehender Wahrscheinlichkeitsmatrizen zurück.

6.4.3 Praktische Umsetzung der Assistenzfunktion

Zur Realisierung der 4. Entwicklungsstufe greift das Assistenzsystem auf den Produktzustandsgraphen in ArangoDB zurück. Ein *Python*-Script erweitert den Produktzustandsgraphen um virtuelle Zustände. Nach diesem Schritt besteht ebenfalls die Möglichkeit, den Produktzustandsgraphen punktuell manuell zu erweitern. Nachdem der erweiterte Produktzustandsgraph vorliegt, wird dieser in einem weiteren Programm in einen Markovgraphen umgewandelt. Die zugrunde liegenden Wahrscheinlichkeiten stammen ebenfalls aus der Graphendatenbank.

Das Assistenzsystem unterscheidet die Aktionen „Abfrage“ und „Montage“. Beide Aktionen können vom Agenten sehr gut unterschieden werden, da das Assistenzsystem die Aktion Abfrage im Mitarbeiterführungssystem selbst initiiert. Nach jeder Beobachtung wird der Belief State ausgewertet. Auf Basis des Belief States wird über die nächste Handlungsanweisung entschieden, dabei wird die Standardabweichung des Belief States herangezogen. Ein Schwellwert kann selbst bestimmt werden. Liegt die Standardabweichung innerhalb des Schwellwertes, wird die Aktion „Montage“ vorgeschlagen. Dies besagt, dass der Produktzustand relativ sicher erkannt wurde. Dennoch muss die Steuerung noch immer prüfen, ob der richtige Schritt gewählt wurde. Ist dies der Fall, wird wie bisher fortgefahren, weicht der Schritt ab und ist mit dem Reparaturplan nicht mehr vereinbar, werden Prozessschritte angezeigt, die die Korrektur ermöglichen sollen. Liegt aber die Standardabweichung oberhalb des Schwellwertes, wird im Mitarbeiterführungssystem die Frage eingeblendet, ob der aktuelle Produktzustand dem aktuell wahrscheinlichsten Zustand entspricht. Der Mitarbeiter kann mit „Ja“ oder „Nein“ antworten. Dabei wird die Beobachtung „Ja“ oder „Nein“ gemacht, was wiederum den Belief State ändert. Hierbei wird aber explizit davon ausgegangen, dass der Mitarbeiter korrekt antwortet.

6.5 Zusammenfassung der digitalen Modellierung eines Assistenzsystems zur Unterstützung in der Nacharbeit

In diesem Kapitel wurde der Weg zur digitalen Gestaltung des Assistenzsystems skizziert. Hierbei ist festzuhalten, dass für den Betrieb das Digitale Assistenzsystem in den Entwicklungsstufen 1-3 von Beginn an verfügbar ist. Grundlage ist ein vollständiges

digitales Modell, dessen Plandaten vom Planer angelegt werden. Die Entwicklungsstufe 4 hingegen wird erst freigeschaltet, nachdem das Assistenzsystem Daten aufgenommen hat und die Parameter Übergangswahrscheinlichkeiten sowie Beobachtungswahrscheinlichkeiten angelernt hat. Daher stellt das Assistenzsystem ein „Lernendes System“ dar.

Abbildung 59 zeigt die Vorgehensweise zur Einrichtung eines assistierenden Nacharbeitsplatzes. Dabei wird das Grundschema Produkt-Prozess-Ressourcen berücksichtigt. Es handelt sich hier um die Vorgehensweise bei der Planung von Montagesystemen. Dahinter steht die Idee, dass das Produkt keine funktionalen Kompromisse eingehen muss, sondern der Prozess und die dazugehörigen Betriebsmittel an das Produkt anzupassen sind. Beginnend bei der Produktanalyse werden Verbaureihenfolgen bestimmt, woraus sich der Produktzustandsgraph ableiten lässt. Dieser beherbergt ebenfalls die Fehlercodes. Parallel dazu müssen die wesentlichen Produktmerkmale und -funktionen analysiert werden, die später für die qualitative Bewertung des Prozesses sowie zur Produktzustandsschätzung genutzt werden.

Der Produktzustandsgraph beschreibt rein die physische Zusammensetzung des Produktes. Daher werden im nächsten Schritt erforderliche Nacharbeitsprozesse definiert. Dabei müssen auch die Merkmale, die für die Prüfung und Schätzung des Produktzustands herangezogen werden, ausgewählt werden.

Im letzten Schritt erfolgt die Auswahl geeigneter Ressourcen. Dies umfasst sowohl die Betriebsmittel als auch eine Mitarbeiterdatenbank. Die Betriebsmittel werden den definierten Prozessen zugeordnet, wobei für die Betriebsmittel in den Kategorien Prozessinformation, -ausführung, -prüfung und -dokumentation entsprechende Parameter definiert werden. Zudem werden Fehlerbehebungsprozesse beschrieben. Zuletzt erfolgt die Gewichtung der Kanten im Produktzustandsgraph.

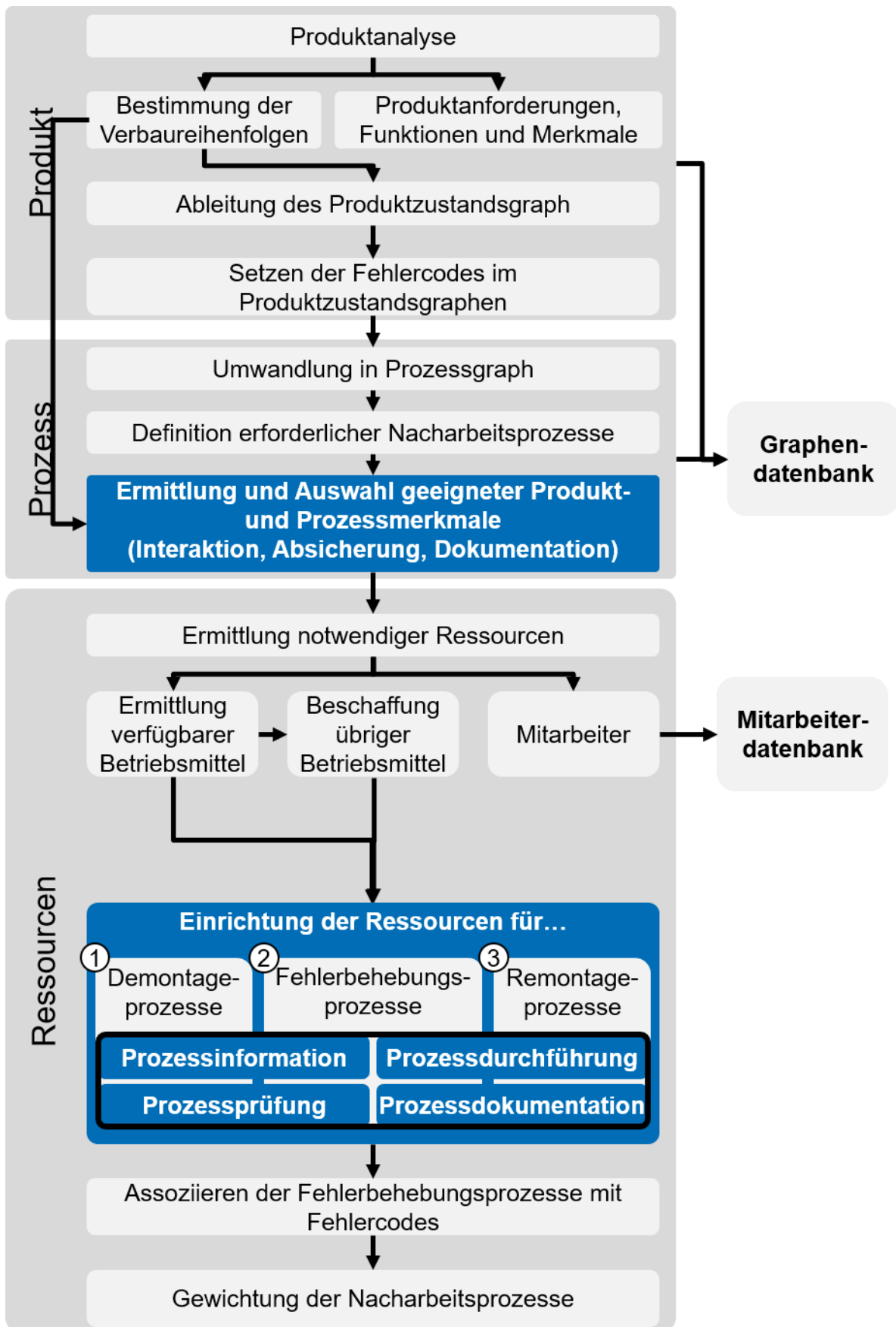


Abbildung 59: Vorgehensweise zur Befähigung eines Kognitiven Assistenzsystems für die Nacharbeit. [Eigene Darstellung]

7 Aufbau der Steuerung des Assistenzsystems

Das Kognitive Assistenzsystem besteht aus mehreren Teilsystemen, die in den vorangegangenen Kapiteln beschrieben wurden. In diesem Kapitel wird die Kommunikation dieser Systeme erläutert. Die Steuerung ist der zentrale Bestandteil des Digitalen Kognitiven Assistenzsystems. Sie reagiert auf die Eingabe des Mitarbeiters, sorgt für die prozessspezifische Informationsbereitstellung, reagiert auf Abweichungen vom Montageprozess und steuert kontextspezifisch weitere Betriebsmittel an. Zudem bewertet sie die Ausgaben der Betriebsmittel. Die Steuerung des Assistenzsystems ist an eine Graphendatenbank angeschlossen.

Abbildung 60 zeigt das Sequenzdiagramm der Standardkommunikation in der zweiten Entwicklungsstufe. Zu Beginn gibt ein Mitarbeiter die Initialdaten (Fehlercode, MitarbeiterID) in das Mitarbeiterführungssystem (MFS) ein. Diese Informationen werden an die zentrale Steuerung des Assistenzsystems weitergeleitet. Auf Basis dieser Daten fragt die Steuerung aus der Datenbank den Produktzustandsgraphen entsprechend des eingegebenen Fehlercodes ab. Aus diesem Graph wird der Pfad ermittelt, der dann der Steuerung als digitaler Nacharbeitsplan dient. Der Nacharbeitsplan wird in Auszügen dem Mitarbeiterführungssystem übermittelt, sodass Prozesse textuell und bildlich dargestellt werden können. Basierend auf dem Nacharbeitsplan werden vom Assistenzsystem Schritt für Schritt die entsprechenden Module parametrieren. Dies erfolgt über das MQTT-Protokoll. Nach erledigter Anweisung erfolgt die Rückmeldung der Prüfbetriebsmittel, sodass die Steuerung die Ergebnisse auswerten und dokumentieren kann. Zeitgleich wird dem Mitarbeiterführungssystem ein Befehl zum Setzen eines grünen Hakens übermittelt, damit der Nutzer über den erfolgreichen Abschluss des Prozesses in Kenntnis gesetzt wird. Die Prozesse werden in einer Schleife abgearbeitet, bis der Auftrag abgeschlossen ist.

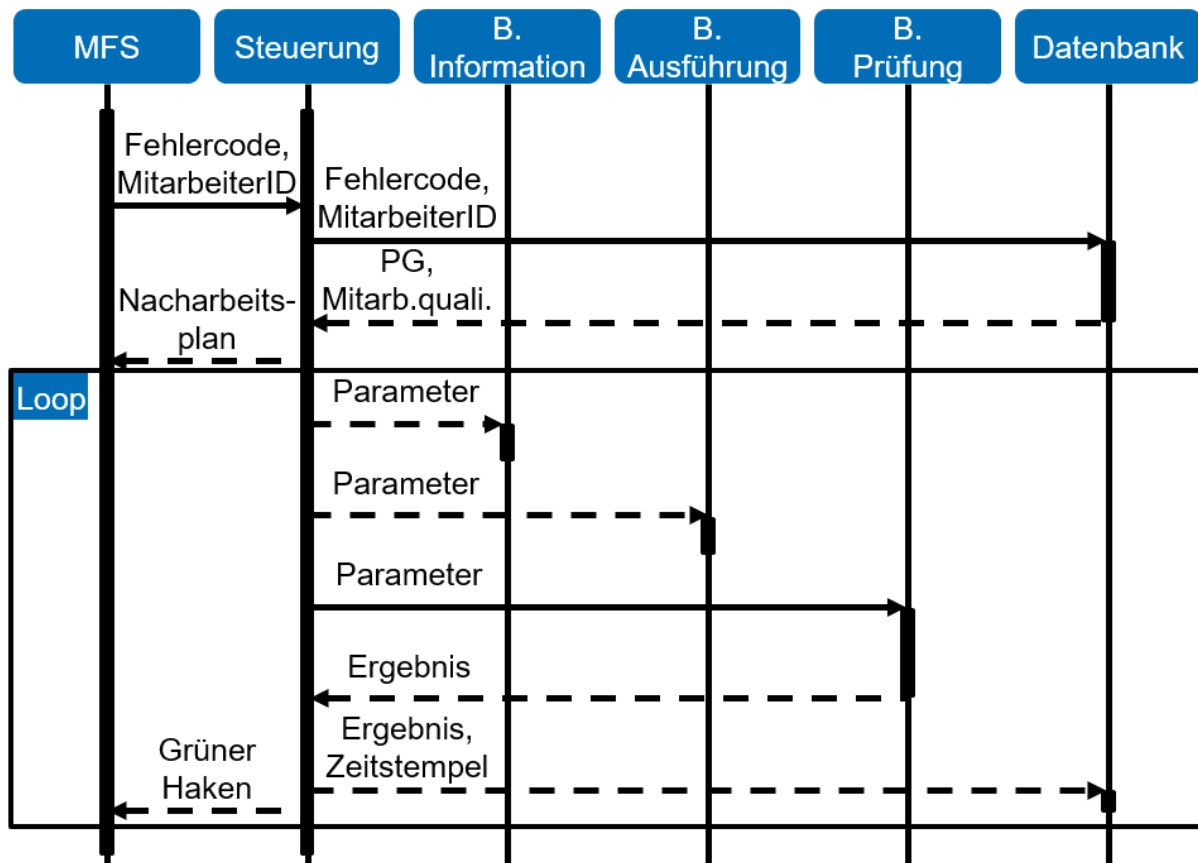


Abbildung 60: Sequenzdiagramm der Standardkommunikation des Assistenzsystems in der zweiten Entwicklungsstufe. [Eigene Darstellung]

Der Mitarbeiter hat jederzeit die Möglichkeit, einen anderen Fehlercode anzugeben, wobei in dem Fall geprüft wird, ob die Eingabe erfolgen kann. Der nachträgliche Ablauf ist jedoch nahezu gleich zu dem bereits beschriebenen Ablauf. Der einzige Unterschied liegt in der Generierung des Nacharbeitsplans. Abbildung 61 zeigt das Sequenzdiagramm für das Assistenzsystem in der vierten Entwicklungsstufe. Die Initialisierungsphase ist gleich der vorangegangenen beschriebenen Initialisierung. Allerdings wird diesmal auch der Belief State nachverfolgt, indem eine weitere Instanz erstellt wird, die ebenfalls die Rückmeldung der Prüfung auswertet. Der Belief State samt dem Aktionsvorschlag wird der Steuerung übermittelt. Der Aktionsvorschlag ist entweder die Fortsetzung der Nacharbeit [N], da der Belief State ausreichend sicher ist, oder die Abfrage im Mitarbeiterführungssystem [A], da im Belief State starke Unsicherheiten über den aktuellen Zustand bestehen. Basierend auf dem Belief State sowie der vorgeschlagenen Aktion passt die Steuerung den nächsten Prozess an. Diese Routine wird so lange durchgeführt, bis der Auftrag abgeschlossen ist.

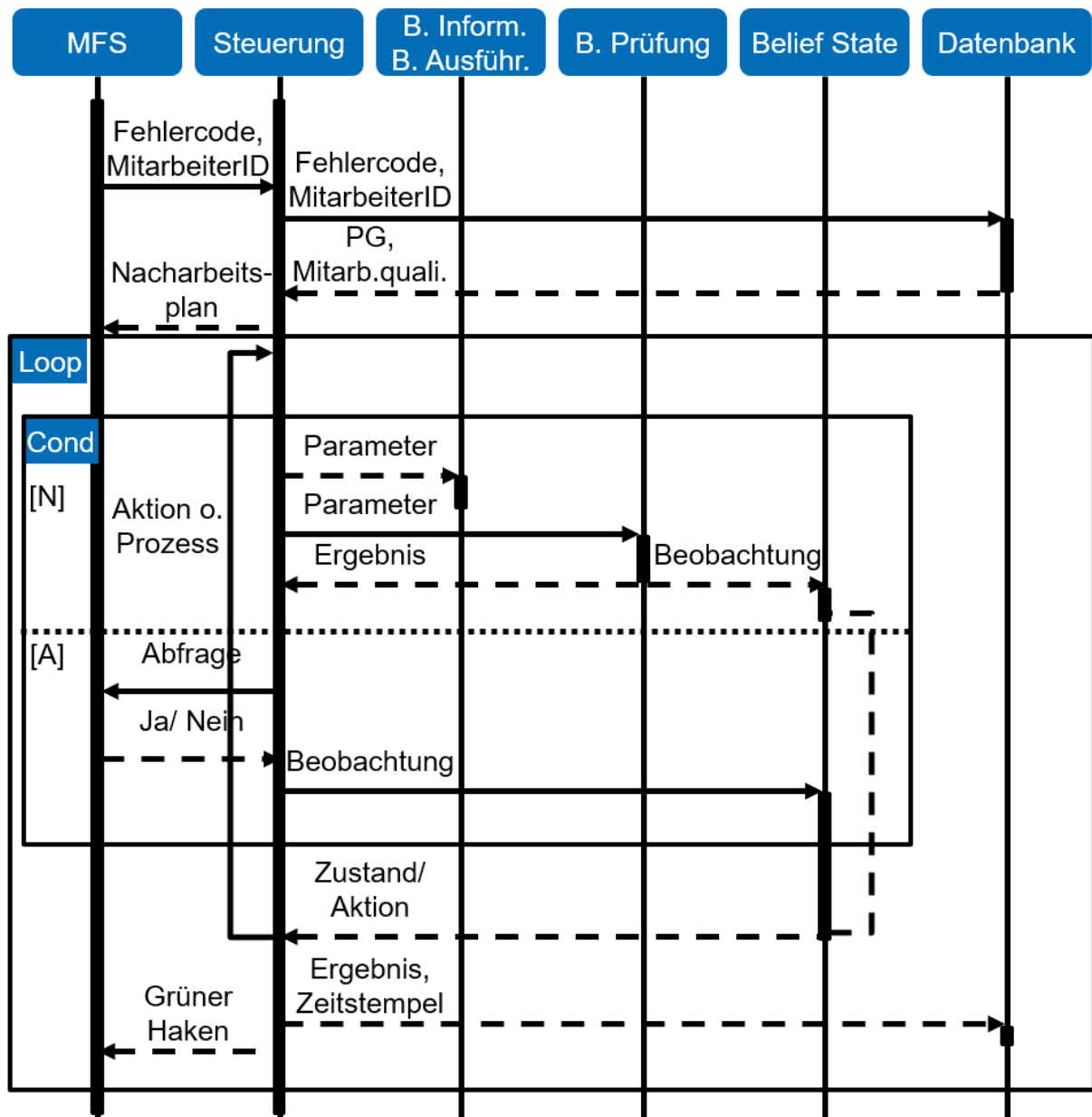


Abbildung 61: Sequenzdiagramm des Standardablaufs der Entwicklungsstufe vierter Stufe. ¶
[Eigene Darstellung]

Das Assistenzsystem greift auf unterschiedliche Protokolle und Systeme zurück. Wesentliche Elemente sind in der Programmiersprache *Python* umgesetzt. Da Betriebsmittel zum Teil anbieterabhängige Schnittstellen besitzen, wurden in *Python* ebenfalls Übersetzer geschrieben, die Anweisungen der Steuerung des Assistenzsystems über MQTT in die jeweilige Systemsprache übersetzen.

8 Optimierte Auftragszuweisung in der Nacharbeit

In den vorangegangenen Kapiteln wurden Entwicklungen mit Hinblick auf das Digitale Kognitive Assistenzsystem getätigt. Wie in Kapitel 2.3 beschrieben, werden am Assistenzsystem Daten und Informationen generiert, die einer erweiterten Nutzung zugutekommen. In dieser Arbeit besteht die erweiterte Nutzung des Assistenzsystems aus der leistungsgerechten Zuweisung der Nacharbeitsaufträge (Produkte) zur Realisierung folgender Ziele:

1. Optimierte Auftragszuweisung zu Schichtbeginn
2. Auftragszuweisung während der Schicht zur Bestandsreduzierung
3. Durchschleusung bestimmter Varianten innerhalb einer definierten Zeit

Die Auftragszuweisung ist nicht ohne Weiteres durch die Auswertung der vorliegenden Informationen umsetzbar. Neben den Mitarbeiter- sowie den Prozessdaten ist die Analyse des Lagers notwendig. Das Lagersystem bzw. das Bereitstellungssystem spielt dabei eine entscheidende Rolle, denn dieses hat maßgeblichen Einfluss auf die Wahlfreiheit eines Auftragsystems bei der Zuweisung der Aufträge an die Mitarbeiter.

8.1 Betrachtung und Auswahl relevanter Lagersysteme

Die Auswahl des Bereitstellungssystems begrenzt die Zugänglichkeit zu den darin enthaltenen Aufträgen. Hierbei muss bedacht werden, dass diese Lagersysteme verschiedene Vor- und Nachteile mit sich bringen. Zuerst wird betrachtet, welche Lagersysteme vor dem Hintergrund der Nacharbeit überhaupt in Betracht gezogen werden sollten. Lagersysteme können in unterschiedlicher Art und Weise gegliedert werden. In dieser Arbeit kann die Gliederung im Hinblick auf Lagergut (Stückgut, Schüttgut, Gas und Flüssigkeitenlager) [156 S.74] und Lagermittel bzw. Lagerbauarten (Bodenlagerung, Statische Lagerung, Dynamische Regallagerung, Lagerung auf Fördermitteln) [156 S.74] eingegrenzt werden. Im vorliegenden Fall wird ausschließlich das Stückgut Lager betrachtet, weil davon ausgegangen wird, dass Nacharbeit lediglich an „diskreten Gütern sogenannten Lagereinheiten (LE)“ [156 S.76] stattfinden kann. Die Lagerbauart ist im einfachsten Fall in Form eines Bodenlagers realisiert, wobei das Bodenlager gestapelt oder ungestapelt sein kann. Im Gegensatz zur Bodenlagerung, vor allem aber der gestapelten Bodenlagerung stehen bei der statischen bzw. dynamischen Lagerung meist Regale zur Verfügung.

In der Nacharbeit wird für den vorliegenden Fall ein Bodenlager verwendet. Ziel ist es ohnehin, den Gesamtbestand effizient zu reduzieren und daher möglichst zeitnah zu bearbeiten. Des Weiteren sind statische, v. a. aber dynamische Lager kostenintensiv und daher von nachrangigerem Interesse. In der Realität werden oftmals Nacharbeitsbereiche mit Investitionen in (möglichst) geringem Umfang bedacht. Denn allein die Herstellung eines mangelhaften Produktes wird als Verschwendung betrachtet. Daher möchte man vorzugsweise Optimierungen an der Fehlerursache vornehmen. Abbil-

Abbildung 62 zeigt das Layout von Bodenlagersystemen, wobei ein Stapelsystem nicht berücksichtigt wurde. Das erste Lagersystem reiht die Lagereinheiten hintereinander und nebeneinander an. Dabei kann der Zugriff lediglich in Pfeilrichtung erfolgen. Das schränkt die Auswahlfreiheit ein. Das Lagersystem aus Abbildung 62b sieht Gänge vor, sodass Lagereinheiten über diese Gänge bevorzugt ausgewählt werden können. Allerdings können Gangflächen nicht als Lagerflächen genutzt werden, wodurch die Kapazität des Lagers sinkt. Eine dritte Alternative wäre das Lagersystem „ohne Ordnung“. Dabei wäre jede Lagereinheit zwar zugänglich, jedoch geht dies zu Lasten der effizienten Flächennutzung bzw. Lagerkapazität. Da das Lagersystem aus Abbildung 62b und c keine Restriktionen bei der Auswahl aus dem Lager darstellt, kann die Entwicklung für das Lagersystem aus Abbildung 62a für die beiden anderen Fälle mit angewendet werden. Zudem ist das Lagersystem aus der ersten Darstellung effizienter und im vorliegenden Fall (exemplarisch) vorzufinden.

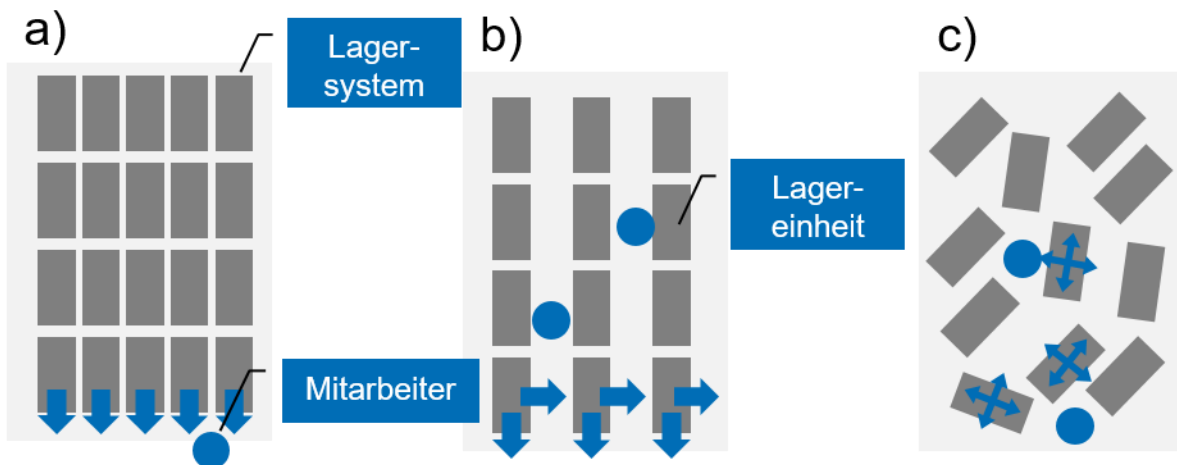


Abbildung 62: Layout von Bodenlagersystemen. [Eigene Darstellung]

8.2 Modellierung der Zugriffsbeschränkung auf den Auftragsbestand

Werden Produkte im Prüfstand der Endmontage als fehlerhaft erkannt, so werden sie in den Nacharbeitsbereich transportiert. Oftmals verweilen dort die Produkte in einem Lager, bis ein Mitarbeiter das Produkt zur Nachbearbeitung entnimmt. Die Entnahmedisziplin ergibt sich aus dem Layout bzw. physischen Aufbau des Lagers. Abbildung 63 zeigt eine schematische Darstellung eines Lagers in vereinfachter Form. Zu beachten ist die Richtung des Materialflusses im Lager sowie die Restriktionen, die es auferlegt. Diese Lagerart erfüllt die Funktion einer Bereitstellungs- bzw. Abstellfläche.

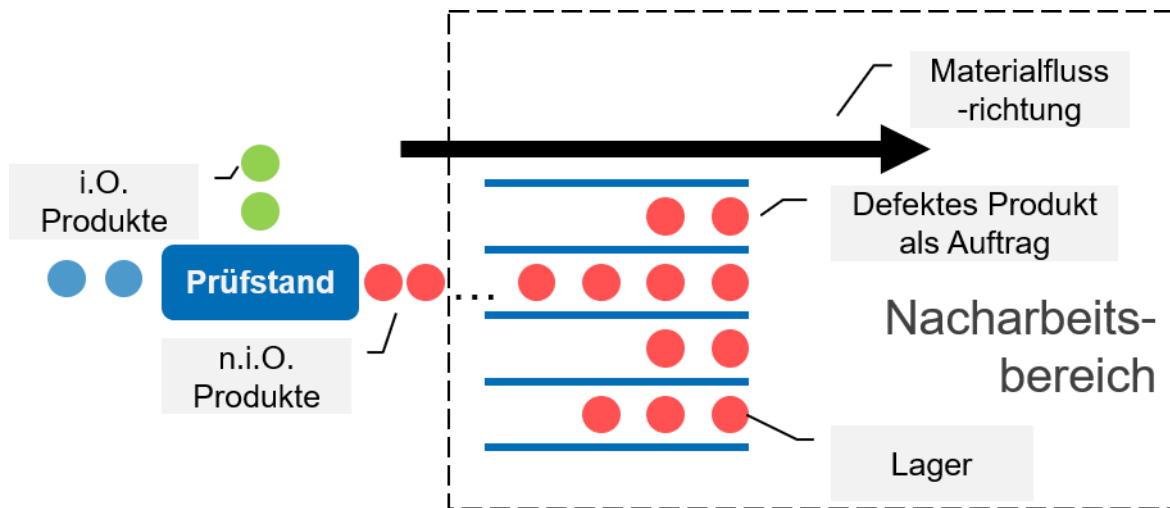


Abbildung 63: Schematische Darstellung des Materialflusses durch einen Auftragspuffer. [Eigene Darstellung]

Die Zugriffsdisziplin ergibt sich aus dem Layout der Bereitstellungsfläche. Im Wesentlichen ist dies der FIFO sowie der LIFO-Ansatz. Da allerdings mehrere Reihen zugänglich sind, ist reihenübergreifend ein zufälliger Zugriff (SIRO: Selection-in-Random-Order) möglich. Im konkreten Anwendungsfall wird die FIFO-Disziplin betrachtet, da diese im Referenzfall vorzufinden ist. Da die Auftragseingänge nicht deterministisch sind, sondern exponentialverteilt und auch die Abarbeitung der Aufträge nicht als deterministisch angesehen werden kann, ist das System in der Kendall-Notation wie folgt zu beschreiben: $M/M/n/\infty/\infty/FIFO$.

Die Anzahl verfügbarer Nacharbeitsplätze ist mit n zu beziffern. Die Kapazität des Lagers wird als unendlich groß angenommen, da kein Alarmsystem bei der Überfüllung des Lagers berücksichtigt wird. Auch die eintreffenden Nachaufträge können beliebig viele sein. Die Entnahmedisziplin ist FIFO.

8.2.1 Modellierung der Zugriffskombinationen

Ein Mitarbeiter kann lediglich einen Auftrag aus dem Lager entnehmen. Unter der Annahme, dass alle Reihen mindestens ein n.i.O.-Produkt enthalten, stehen ihm mehrere Möglichkeiten zur Auswahl (SIRO) zur Verfügung. Die Auftragszuweisung berücksichtigt jedoch nicht allein einen Zugriff für einen Mitarbeiter, sondern mehrere Zugriffe für entsprechend viele Mitarbeiter gleichzeitig. Je mehr Mitarbeiter zur Verfügung stehen, desto mehr Zugriffskombinationen sind möglich. Abbildung 64 zeigt eine Auswahl an Zugriffsmöglichkeiten am Beispiel vier zu entnehmender Nacharbeitsaufträge. Um die Anzahl der Zugriffskombinationen zu bestimmen, wird im Folgenden auf Verfahren aus der Kombinatorik zurückgegriffen.

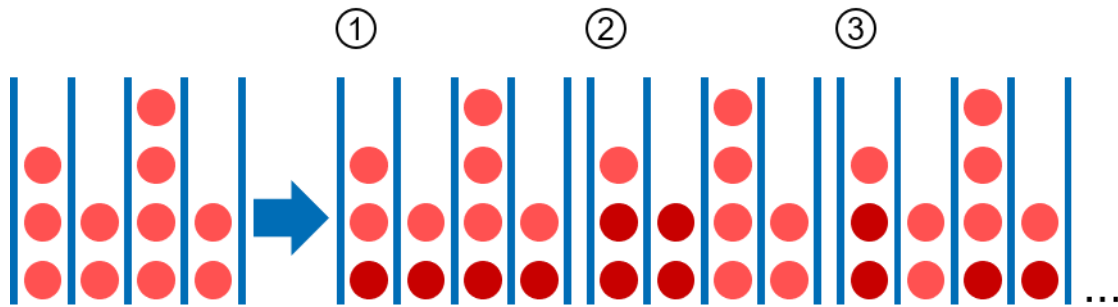


Abbildung 64: Entnahmemöglichkeiten aus einer Bereitstellungsfläche für vier zu entnehmender Produkte. [Eigene Darstellung]

Zur Ermittlung der Anzahl von Entnahmemöglichkeiten aus einem FIFO-Lager kann der Binomialkoeffizient verwendet werden. Mithilfe des Binomialkoeffizienten können die Möglichkeiten aus einer Menge mit n unterscheidbaren Elementen k auszuwählen ermittelt werden. Die Anzahl der Möglichkeiten wird wie folgt berechnet:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (24)$$

Zur Bestimmung der Entnahmemöglichkeiten aus dem Lager wird eine Analogie zum Stapeln von Kisten genutzt. Abbildung 65a zeigt eine Bereitstellungsfläche mit vier in Teilen befüllte Reihen. Betrachtet man Abbildung 65b ist die Analogie zu einem Stapel erkennbar. Ein Stapel wird von der Basis nach oben (gegen die Schwerkraft) aufgebaut. Ähnlich werden Aufträge von der Bereitstellungsfläche aufgrund der Zugänglichkeit entnommen. Abbildung 65c zeigt eine Interpretation eines Stapelexperimentes in Form von Stars and Bars (dt.: Sterne und Balken), die von William Feller vorgestellt wurde [157 S.38]. Dieser Vergleich zeigt die Möglichkeiten, wie Lager mit ununterscheidbaren Elementen befüllt werden können. Hierbei stellen die Stars die Elemente des Lagers und die Bars die Abgrenzung von einer Reihe zur nächsten Reihe dar. Das Stapeln ist nicht notwendig. Es interessiert lediglich die Anzahl der Elemente pro Reihe. In dieser Kombination sind zwei Bars nebeneinander vorstellbar, die dann als eine Reihe ohne enthaltene Elemente interpretiert werden kann. Im Stars-and-Bars-Modell gibt es genauso viele Plätze, wie es Elemente des Lagers sowie Reihen reduziert um eins gibt. Die Frage lautet: Wie viele Kombinationsmöglichkeiten gibt es, um mit den Bars die Plätze zu belegen, wobei die Plätze zwischen den Bars mit Stars aufgefüllt werden?

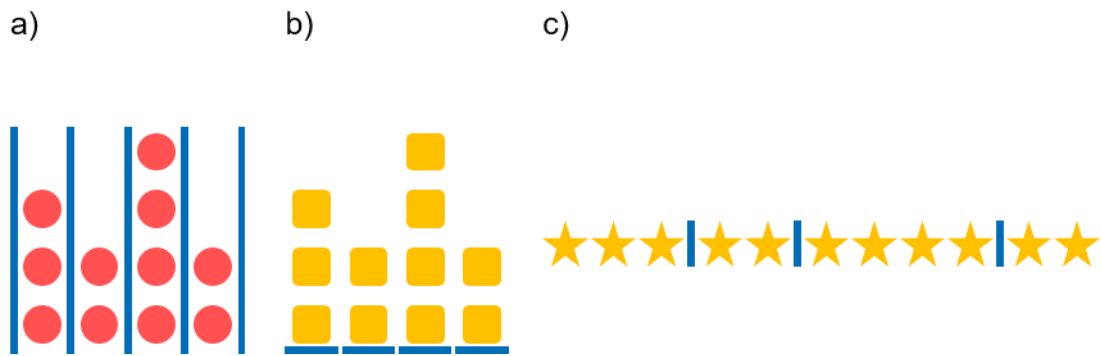


Abbildung 65: a) Bereitstellungsfläche b) Stapel c) Stars and Bars. ¶[Eigene Darstellung]

Die Kombinationsmöglichkeiten des Stars-and-Bars-Problems können mit Hilfe des Binomialkoeffizienten aus Gleichung (24) bestimmt werden, wie Gleichung (25) zeigt [157 S.38]:

$$\binom{s+b-1}{b-1} = \frac{(s+b-1)!}{(b-1)!(s+b-1-b+1)!} = \frac{(s+b-1)!}{(b-1)!s!} \quad (25)$$

Da der Binomialkoeffizient eine gewisse Symmetrie in der Form $\binom{n}{k} = \binom{n}{n-k}$ aufweist, ist es ohne Bedeutung, ob die Kombinationsmöglichkeiten der Bars oder die Möglichkeiten der Stars berechnet werden, da der Rest immer mit dem jeweils anderen Element aufgefüllt wird. Für den vorliegenden Fall werden die Stars (s) als Zugriffe auf einen Auftrag umgedeutet, wobei die Bars (b) die Lagerreihen voneinander trennen. Die Zugriffe bzw. Zuweisungen hängen von der Anzahl zur Verfügung stehender Mitarbeiter ab.

Abbildung 66 zeigt eine Auswahl von Zugriffsmöglichkeiten bei vier freien Mitarbeitern bzw. vier Mitarbeitern, die während der Auftragszuordnung berücksichtigt werden.

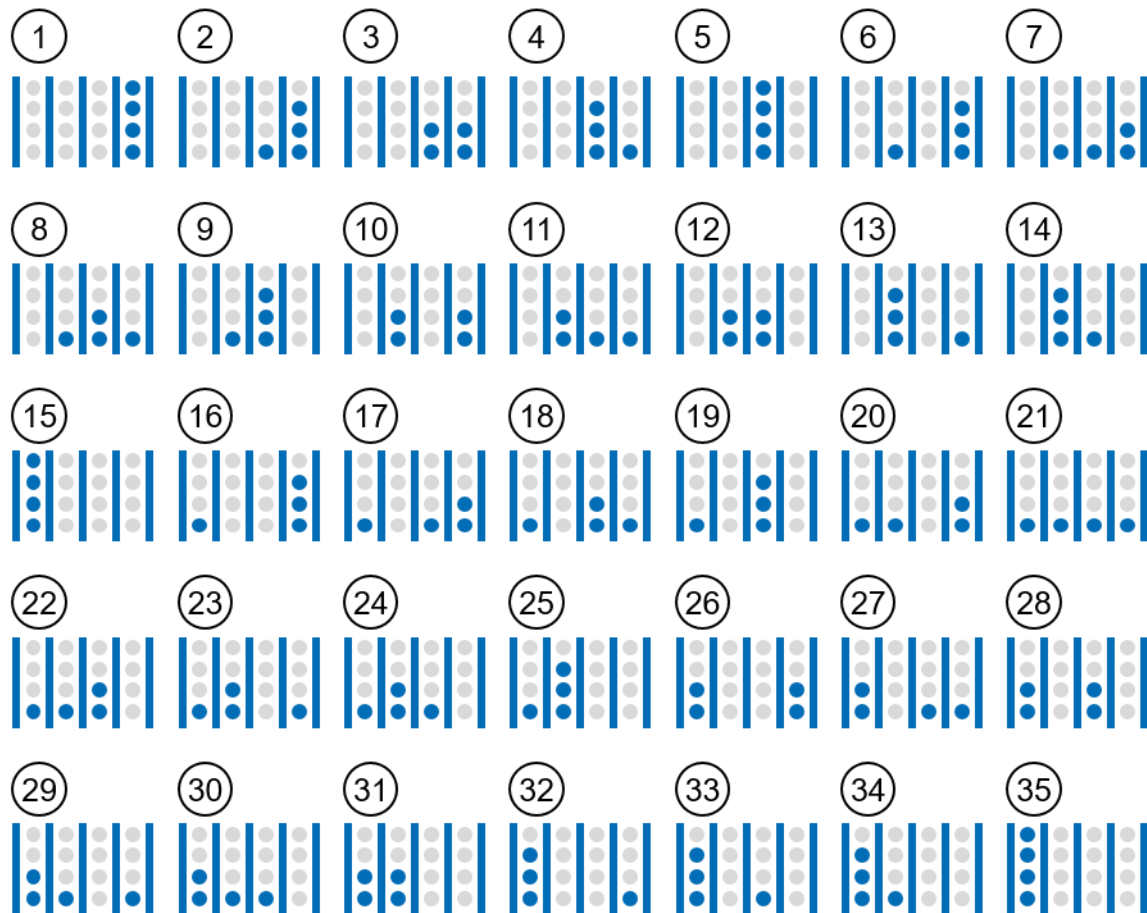


Abbildung 66: Zugriffsmöglichkeiten bei der Vergabe von vier Aufträgen. ¶[Eigene Darstellung]

Bisher wurde lediglich die Anzahl der Kombinationen ermittelt. Im nächsten Schritt ist es erforderlich, die eigentlichen Kombinationen zu bestimmen. Anhand der eigentlichen Kombinationen kann festgestellt werden, welcher Auftrag bearbeitet wird. Hierzu wird das Stars-and-Bars-Beispiel genauer erläutert. In der Regel ermittelt der Binomialkoeffizient die Anzahl der Möglichkeiten, k unterscheidbare Elemente aus einer Menge n unterscheidbarer Elemente auszuwählen, ohne dass Elemente in der Auswahl wiederholt auftauchen oder dass die Reihenfolge von Belang ist. Da weder Stars noch Bars voneinander zu unterscheiden sind, werden alle Elemente auf feste Stellen positioniert, wie in Abbildung 67 zu sehen ist. Gemäß dieser Abbildung wäre die Kombinationsausgabe [D, G; L], da hier die Bars platziert sind.



Abbildung 67: Stars and Bars auf fixen Stellen positioniert. ¶[Eigene Darstellung]

Die Kombinationen können mittels Algorithmus nach Knuth generiert werden [158 S.4]. Der Algorithmus stellt sicher, dass die Kombinationen in lexikalisch geordneter Form generiert werden, wobei die Reihenfolge der Elemente innerhalb der Kombination nicht berücksichtigt wird. Somit gelten Mengen mit gleichen Elementen in unterschiedlicher

Reihenfolge (z. B. [A,B,C], [C,B,A]) nicht als weitere Kombinationsmöglichkeit, sondern als gleichbedeutend.

Aus dem Ergebnis des lexikalischen Kombinationsalgorithmus nach Knuth ist jedoch nicht unmittelbar die Anzahl entnehmbarer Aufträge pro Reihe ersichtlich. Der Algorithmus gibt lediglich an, an welcher Position sich die Trennlinien (Bars) befinden. Übertragen auf die Aufträge bedeutet dies, dass anhand der Bars abgezählt werden kann, wie viele Aufträge zwischen den Bars liegen. Hierbei gilt es jedoch zu beachten, dass die berücksichtigten Bars nicht exakt der Anzahl an Reihen entsprechen, sondern eins kleiner als die tatsächliche Anzahl der Reihen in der Bereitstellungsfläche sind. $b - 1$ Trennlinien/ Bars genügen, um die Elemente in b Reihen einzuteilen, wie in Abbildung 65c erkennbar ist.

Aus einer generierten Kombination kann die Anzahl zu entnehmender Aufträge pro Reihe wie folgt abgeleitet werden. Zu Beginn muss die Anzahl der Stellen bekannt sein. Diese ergeben sich aus der Summe aus der Anzahl zu vergebender Aufträge (Stars) und der Anzahl der Reihen abzüglich 1 (Bars). Die Stellen werden aufsteigend nummeriert. In Abbildung 68 wird eine alphabetische Nummerierung vorgenommen, die aber problemlos in eine bezifferte Nummerierung umgewandelt werden kann. Zunächst wird die Anzahl freier Positionen zwischen dem ersten Bar der Kombination und der ersten Position (in der Regel A bzw. 1) berechnet. Danach werden aus zwei Bars die Anzahl freier Positionen zwischen den beiden Bars berechnet. Für das letzte Element in der Kombination gilt die Anzahl freier Positionen zwischen dem letzten existierenden Platz und dem letzten Platz in der Kombinationsmenge. In Abbildung 68a–e ist zu sehen, dass die Bars jeweils auf unterschiedlichen Plätzen positioniert werden. Es werden lediglich die Bars betrachtet. Aus den drei Positionen der Bars lässt sich die Anzahl der Stars pro Gruppe ermitteln.






	Stars und Bars auf verschiedenen Positionen	Kombinationen verschiedener Positionen der Bars	Zugehörige Anzahl zu entnehmender Aufträge pro Slot
a)	 $\frac{\text{A}}{\text{A}} \quad \frac{\text{B}}{\text{B}} \quad \frac{\text{C}}{\text{C}} \quad \frac{\text{D}}{\text{D}} \quad \frac{\text{E}}{\text{E}} \quad \frac{\text{F}}{\text{F}} \quad \frac{\text{G}}{\text{G}}$	[A,B,C]	[0,0,0,4]
b)	 $\frac{\text{A}}{\text{A}} \quad \frac{\text{B}}{\text{B}} \quad \frac{\text{C}}{\text{C}} \quad \frac{\text{D}}{\text{D}} \quad \frac{\text{E}}{\text{E}} \quad \frac{\text{F}}{\text{F}} \quad \frac{\text{G}}{\text{G}}$	[A,B,D]	[0,0,1,3]
c)	 $\frac{\text{A}}{\text{A}} \quad \frac{\text{B}}{\text{B}} \quad \frac{\text{C}}{\text{C}} \quad \frac{\text{D}}{\text{D}} \quad \frac{\text{E}}{\text{E}} \quad \frac{\text{F}}{\text{F}} \quad \frac{\text{G}}{\text{G}}$	[A,D,G]	[0,2,2,0]
d)	 $\frac{\text{A}}{\text{A}} \quad \frac{\text{B}}{\text{B}} \quad \frac{\text{C}}{\text{C}} \quad \frac{\text{D}}{\text{D}} \quad \frac{\text{E}}{\text{E}} \quad \frac{\text{F}}{\text{F}} \quad \frac{\text{G}}{\text{G}}$	[B,D,G]	[1,1,2,0]
e)	 $\frac{\text{A}}{\text{A}} \quad \frac{\text{B}}{\text{B}} \quad \frac{\text{C}}{\text{C}} \quad \frac{\text{D}}{\text{D}} \quad \frac{\text{E}}{\text{E}} \quad \frac{\text{F}}{\text{F}} \quad \frac{\text{G}}{\text{G}}$	[B,D,F]	[1,1,1,1]

Abbildung 68: Ableitung der Anzahl zu entnehmender Aufträge pro Reihe in der Bereitstellungsfläche von der Kombination von an fixen Positionen platzierten Bars. [Eigene Darstellung]

Die Elemente der Kombinationsmöglichkeiten, bspw. [0, 0, 0, 4] aus Abbildung 68 zeigen lediglich an, aus welcher Reihe wie viele Aufträge zu entnehmen sind. Allerdings gibt die Kombination keine Auskunft darüber, welche Aufträge explizit bearbeitet werden. Dennoch können die Aufträge über ihre Position im Lager ermittelt werden. Zur Ermittlung des exakten Auftrags wird eine Auftragsklasse generiert. Diese wird in Kapitel 10.1.1 vorgestellt. Zudem gilt es zu beachten, dass die Anzahl der Möglichkeiten die maximale Anzahl der Zugriffe pro Auftragsvergabe widerspiegelt. In der Realität kann die Kombinationsmöglichkeit geringer ausfallen, da bestimmte Lagerreihen auch bspw. leer sein können oder weniger Produkte enthalten können, als die Kombination zulässt. Daher ist vor der Auftragsvergabe die Existenz des Auftrags zu prüfen.

8.2.2 Modellierung der Zuweisungskombination

Bisher wurden lediglich die Entnahmemöglichkeiten der Aufträge aus der Bereitstellungsfläche betrachtet. Allerdings müssen diese Aufträge auch den einzelnen Mitarbeitern zugewiesen werden. Die Entnahmekombination berücksichtigt zwar die Zahl der verfügbaren Mitarbeiter, jedoch betrachtet sie nicht die Möglichkeit, die Aufträge an unterschiedliche Mitarbeiter zu verteilen. Die Zuweisungskombination beachtet die Möglichkeiten, die Aufträge an die verfügbaren Mitarbeiter zuzuweisen. Abbildung 69

zeigt anhand von fünf Aufträgen (nummerierte Kreisflächen), wie diese in verschiedenen Kombinationen Mitarbeitern zugewiesen werden können. Da genauso viele Aufträge aus der Bereitstellungsfläche entnommen werden, wie Mitarbeiter zur Verfügung stehen, bleibt bei der Zuweisung kein ausgewählter Auftrag übrig. Stellt man sich vor, dass die verfügbaren Mitarbeiter nebeneinander gereiht und für einen bestimmten Platz stehen, so können die Aufträge auf verschiedenen Plätzen positioniert werden. Somit entspricht die Anzahl der Zuweisungsmöglichkeiten der Permutation zuzuweisender Aufträge.

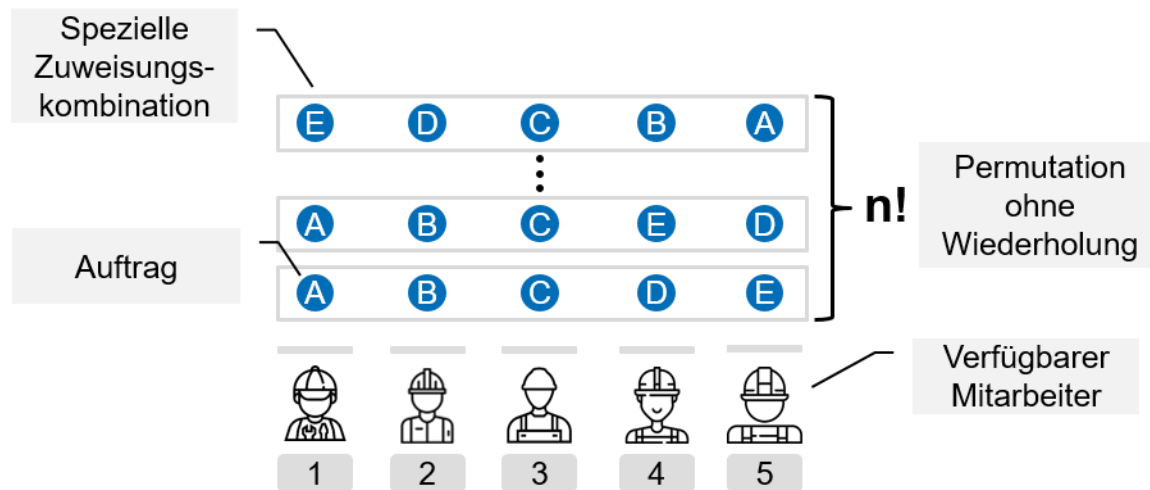


Abbildung 69: (Auftrags-) Zuweisungskombination an verfügbare Mitarbeiter [Eigene Darstellung]

Wird die Annahme zugrunde gelegt, dass kein Auftrag gleich ist oder mehrfach vergeben werden soll, eignet sich die Permutation ohne Wiederholung zur Bestimmung der Zuweisungskombinationen. Diese lässt sich als Fakultät permutierender Elemente berechnen.

$$n! = n * (n - 1) * (n - 2) \dots * 1 \quad (26)$$

Dabei entspricht die Variable n der Anzahl zuzuweisender Aufträge. Sind zwei Aufträge gleich, was vorkommen kann, wenn in der Serienmontage durch einen unerfahrenen Mitarbeiter Fehler wiederholt vorkommen, kann die Anzahl der Zuweisungskombinationen durch die Permutation mit Wiederholung berechnet werden. Dabei bezeichnet k die Anzahl ununterscheidbarer Objekte. Vor dem Hintergrund der Auftragszuweisung dient dies dazu, Aufträge hinsichtlich des Bearbeitungsablaufes bzw. -aufwandes als gleichwertig zu betrachten.

$$\frac{n!}{k!} \quad (27)$$

Formel (27) kann weiter verallgemeinert werden, wenn mehrere (s) Objektgruppen/Auftragsgruppen erstellt werden können, deren zugehörige Objekte untereinander ununterscheidbar sein sollen. Zur Berechnung der Kombinationsmöglichkeiten unter den angegebenen Bedingungen eignet sich der Multinomialkoeffizient:

$$\frac{n!}{k_1! * k_2! * \dots * k_s!} \quad (28)$$

Dabei entspricht s der Anzahl der Gruppen, wobei k die Anzahl der Objekte je Gruppe wiedergibt. Daher gilt auch:

$$n = \sum_{i=1}^s k_i \quad (29)$$

Damit der Auftrag tatsächlich zugewiesen werden kann, ist die Kenntnis der tatsächlichen Kombination notwendig. Hierzu kann ebenfalls der Algorithmus nach Knuth verwendet werden.

Die gesamte Kombinationsanzahl C setzt sich aus dem Produkt aus Entnahmemöglichkeiten (aus der Bereitstellungsfläche) und der Anzahl der Auftragszuweisungsmöglichkeiten (an verfügbare Mitarbeiter) zusammen. Lässt man die Anzahl gleicher Aufträge außer Acht ergibt sich als maximale Gesamtkombination:

Gesamtkombination = Zugriffskombination \times Auftragszuweisungskombination

$$C = \binom{n+k-1}{n} n! \quad (30)$$

$$C = \frac{(n+k-1)!}{n!(n+k-1-n)!} \times n! \quad (31)$$

$$C = \frac{(n+k-1)n!}{n!(n+k-1-n)!}$$

$$C = \frac{(n+k-1)!}{(k-1)!} \quad (32)$$

Vorausgesetzt, dass genügend Aufträge verfügbar sind, kann bereits bei kleinen Lagergrößen und einstelligen Auftragszahlen die Anzahl der Gesamtkombinationen in den siebenstelligen Bereich reichen. Dies wirft die Frage auf, ob die optimale Lösung oder eine annehmbare Lösung generiert werden soll. Zwar sind Abläufe gerade in der manuellen Nacharbeit im Vergleich zu hochautomatisierten Produktionsbereichen nicht sehr zeitkritisch, aber die Bewertung von Millionen Kombinationsmöglichkeiten ist dennoch sehr zeitaufwendig. Daher wird in den nächsten Kapiteln die Herangehensweise mittels Metaheuristiken beschrieben. Diese können zwar keine optimale Lösung garantieren, aber eine anbieten, die den vorgegebenen Kriterien entspricht. Dies setzt voraus, dass eine entsprechende Lösung existiert.

8.3 Zuweisung der Reparaturaufträge an verfügbare Mitarbeiter

Zur Ermittlung geeigneter Zuweisungen wird im Folgenden auf Metaheuristiken zurückgegriffen, wie sie in Kapitel 4.2 bereits beschrieben wurden. Sowohl die Herangehensweise auf Basis des populationsbasierten Genetischen Algorithmus als auch der Simulated Annealing werden in den folgenden Kapiteln erläutert.

8.3.1 Ermittlung von Näherungslösungen mit Hilfe des Genetischen Algorithmus

Die prinzipielle Funktionsweise des Genetischen Algorithmus wurde in Kapitel 4.2 erläutert. Daher wird hier lediglich punktuell auf einige Parameter und Einflussgrößen des Algorithmus eingegangen. Abbildung 70 zeigt in fünf Schritten den Ablauf des Genetischen Algorithmus.

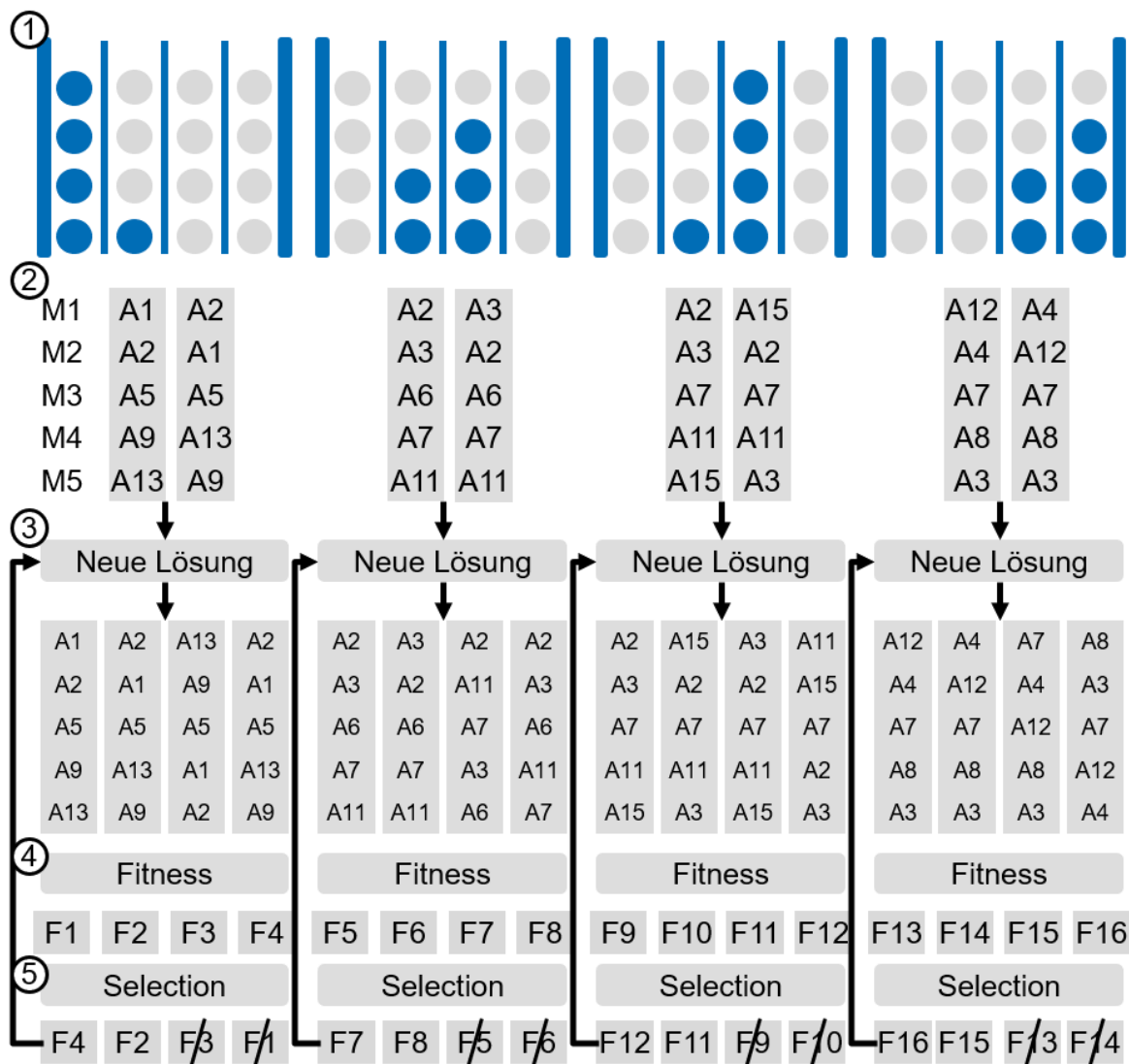


Abbildung 70: Graphenbasierte Darstellung der Bereitstellung samt initialer Freiheitsgrade [Eigene Darstellung]

Im ersten Schritt, in der Initialisierungsphase, werden Lösungskandidaten zufällig generiert. (1) Beginnend bei der Auswahl von Aufträgen aus dem Lager werden (2) Lösungsvektoren erstellt, die die ausgewählten Aufträge (bspw. A1, A2, A5, A9, A13) den verfügbaren Mitarbeitern zuweisen. Jede Stelle des Vektors ist mit einem verfügbaren Mitarbeiter assoziiert.

Basierend auf den nunmehr der Elterngeneration zugerechneten Vektoren werden im zweiten und dritten Schritt unter Anwendung von Rekombinations- und Mutationsalgorithmen neue Lösungen erstellt (3). Dies geschieht durch Rekombination der selektierten Lösungen. Im vorliegenden Fall muss bedacht werden, dass eine beliebige Rekombination zweier Lösungsvektoren zu invaliden Lösungen führen kann. Abbildung 71 verdeutlicht die Auswirkungen unbedachter Rekombinationen. Es wird deutlich, dass die neue Lösung keine valide Entnahmekombination darstellt. Daher muss bei der Rekombination die Machbarkeit berücksichtigt werden.

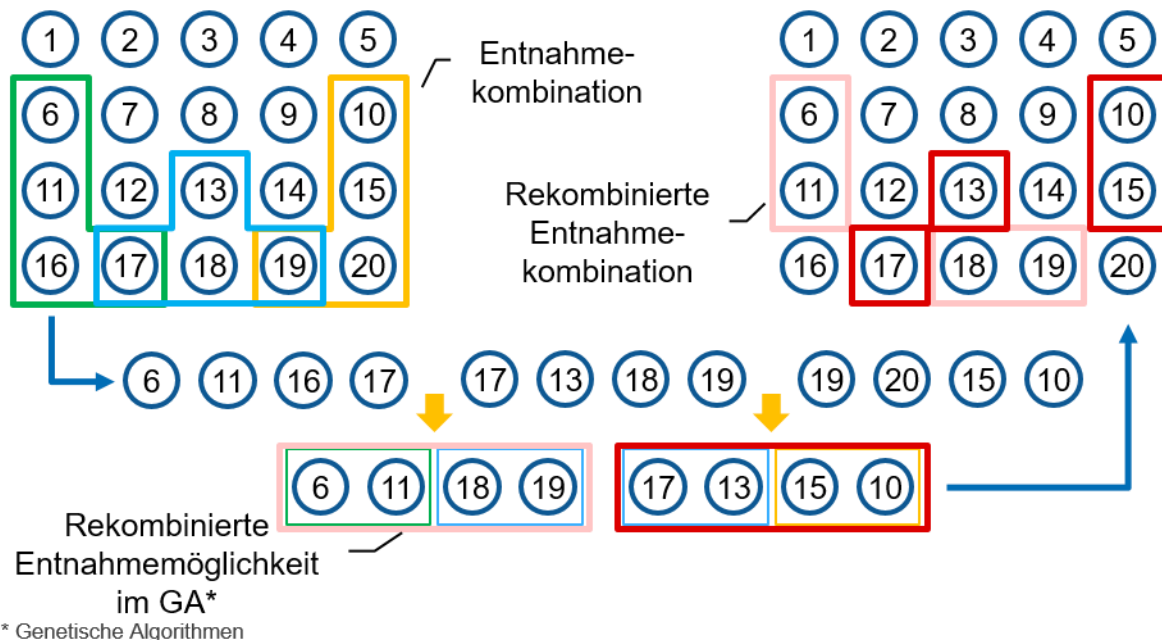


Abbildung 71: Die Entnahmekombination wird im Genetischen Algorithmus im Rekombinations-schritt zu einer unerlaubten Kombination, deren Entnahme nicht möglich ist. [Eigene Darstellung]

Die Generierung von validen Lösungsvektoren kann sichergestellt werden, indem die Rekombination lediglich innerhalb einer Gruppe von Vektoren ermöglicht wird, die alle derselben Zugriffskombination zugehörig sind. Abbildung 70 zeigt in (2) und in (3) Lösungsvektoren, die alle derselben Zugriffskombination (1) zugerechnet werden können. Damit aber auch neuere Zugriffskombinationen Berücksichtigung finden, werden diese in größeren Abständen aktualisiert. Abbildung 72 zeigt schematisch die Suche nach optimierten Lösungen unter Berücksichtigung neuer Zugriffskombinationen. Wenig aussichtsreiche Zugriffe werden von der weiteren Betrachtung ausgeschlossen, wohingegen aussichtsreiche Zugriffe weiterverfolgt werden und Nachbarn dieser Auswahl hinzukommen. Ein Nachbar im Sinne des Lösungsraums ist eine Zugriffskombination, die lediglich in wenigen Elementen von der Bezugskombination abweicht.

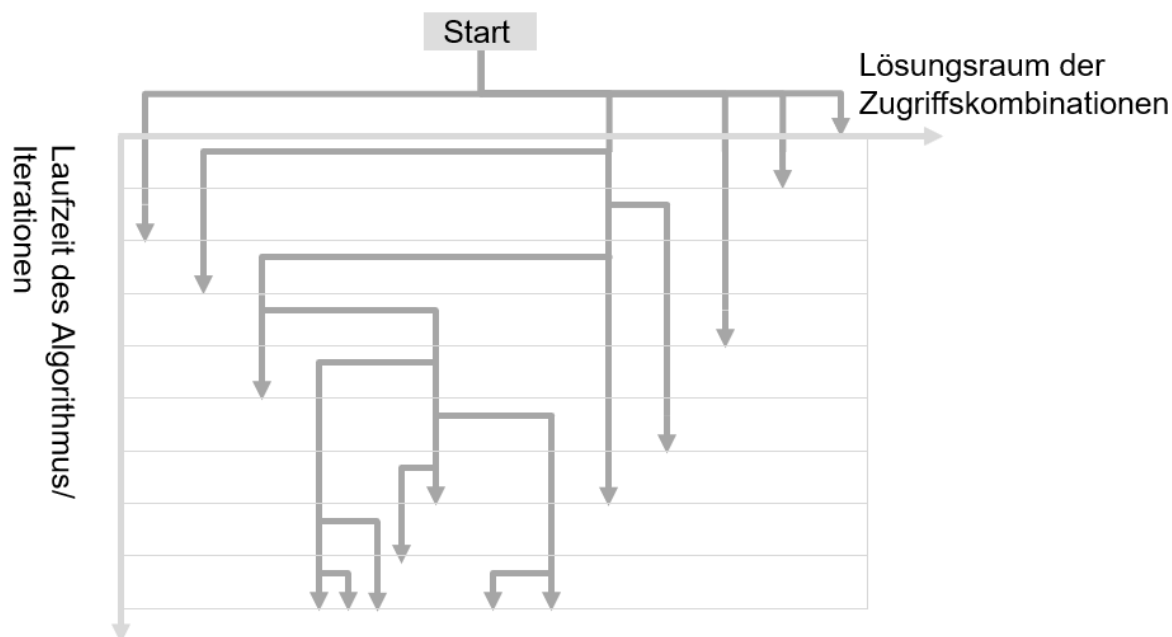


Abbildung 72: Schematisches Beispiel zur Suche im Kombinationsraum der Lagerzugriffe [Eigene Darstellung]

Mit bzw. innerhalb einer Zugriffskombination erfolgt die Auftragszuweisung an Mitarbeiter. Hierbei ist zu beachten, dass eine unachtsam gewählte Rekombination zu doppelter Auftragsvergabe führen kann, was einer Auftragsdoppelvergabe gleichkommt. Daher wird die Rekombinationsmethode „ordered crossover“ angewandt. Diese Methode stellt sicher, dass bei der Rekombination kein Element (Gen) in der Lösung (Chromosom) doppelt vorkommt. Auch bei der Mutation muss bedacht werden, dass durch die Mutation kein Gen doppelt erscheinen darf. Dies wird durch die Swap-Mutation-Methode realisiert. Bei dieser Methode werden zwei Gene innerhalb des Chromosoms vertauscht, sodass weder ein Gen fehlt noch ein Gen doppelt vorkommt.

Für die neuen Lösungen werden im vierten Schritt die Fitnesswerte ermittelt (4). Die Fitnessfunktion wird wie folgt definiert:

$$K(M_v, O) = \frac{1}{|M_v|} \sum_{m_i \in M_v} \sum_{p_j \in o_i} p_{m_i j} \quad (33)$$

M_v bezeichnet die Liste der verfügbaren Mitarbeiter. O ist die Liste der zugeteilten Nacharbeitsaufträge. Der Auftrag auf Platz i aus O wird dem Mitarbeiter m_i auf Platz i zugewiesen. o_i ist ein Auftrag aus O . o_i ist der zu m_i korrespondierende Auftrag. p_j bezeichnet einen Prozess aus dem Nacharbeitsauftrag, wobei j der Laufindex über alle Nacharbeitsprozesse des Auftrags ist. $p_{m_i j}$ ist die Prozessvertrautheit des Mitarbeiters m_i mit dem Prozess j . Die Prozessvertrautheit wird in Kapitel 8.4 beschrieben.

Im fünften Schritt erfolgt die rangbasierte Selektion der Lösungsvektoren.

Im sechsten Schritt wird entschieden, ob ein Abbruchkriterium zutrifft und im Falle positiver Rückmeldung wird die bisher beste Lösung ausgegeben. In der Regel sind die folgenden Kriterien entscheidend für einen Abbruch:

1. Die Fitness einer Lösung genügt den Anforderungen, sodass ein Lösungskandidat als Näherungslösung angenommen wird.
2. Ein zeitlicher Grenzwert bspw. durch Hochzählen wurde erreicht, ohne dass eine angemessene Lösung erzielt wurde. In dem Fall wird die bis dahin beste Lösung ausgegeben.
3. Die Fitnesswerte stagnieren über mehrere Iterationen hinweg und es ist nicht davon auszugehen, dass weitere Iterationen signifikant bessere Lösung generieren werden.

Trifft kein Abbruchkriterium zu, wird der Algorithmus ab Schritt zwei wiederholt.

8.3.2 Ermittlung von Näherungslösungen mit Hilfe des Simulated Annealing

Simulated Annealing ist mit dem Bergsteigeralgorithmus vergleichbar [90 S.81] und wurde in Kapitel 4.2 beschrieben. Aufgrund der stochastischen Komponente kann der Algorithmus aus dem Initialzustand lokalen Optima entkommen.

Im ersten Schritt wird eine initiale Lösung generiert. Hierzu wird eine Lösung betreffend der Entnahme aus der Bereitstellungsfläche und im Anschluss eine Zuweisungskombination ausgewählt. Abbildung 73 zeigt ein Beispiel für die Zugriffs- bzw. Zuweisungskombination. Die aus dem Lager entnommenen Produkte (1-5) werden als Aufträge (A1-A5) den Mitarbeitern zugewiesen.

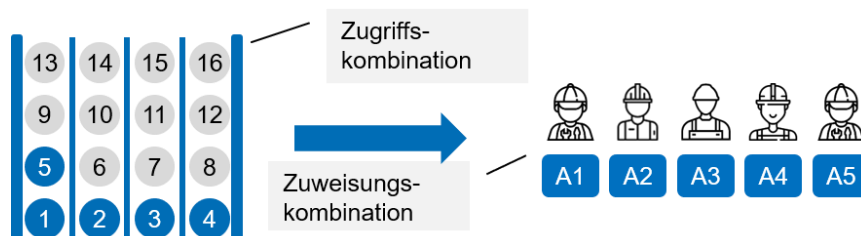


Abbildung 73: Beispiel für die Zugriffskombination und die Zuweisungskombination im Simulated Annealing [Eigene Darstellung]

Im zweiten Schritt wird eine Nachbarlösung generiert. Abbildung 74 zeigt die Kombinationsmöglichkeiten unter Berücksichtigung der Restriktionen eines FIFO-Lagers. Als Nachbarlösungen müssen im Simulated Annealing Lösungen gelten, die hinsichtlich der Referenzlösung eine geringe Abweichung aufweisen. Für die Erstellung der Nachbarlösungen werden im aktuellen Fall zwei Möglichkeiten in Betracht gezogen:

1. Generierung der Nachbarlösung durch Ersetzen
2. Generierung der Nachbarlösung durch Vertauschen

Nachbarlösungen können durch das Ersetzen eines Auftrags durch einen anderen nicht bereits zugewiesenen Auftrag generiert werden. Diese Lösung unterscheidet sich lediglich in einer Zuweisung von der Referenzlösung. Eine alternative Möglichkeit besteht aus dem Vertauschen zweier bereits zugewiesener Aufträge. Diese Lösung unterscheidet sich von der Referenzlösung um zwei Neuzuweisungen.

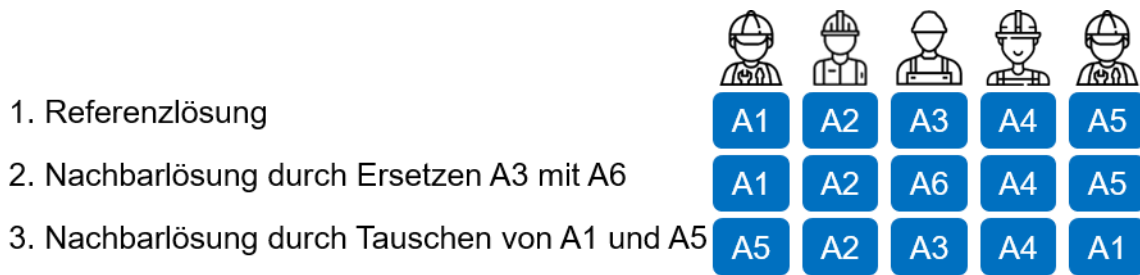


Abbildung 74: Möglichkeiten zur Generierung von Nachbarlösungen [Eigene Darstellung]

Im dritten Schritt wird die neu generierte Lösung bewertet. Das Ergebnis der Bewertung wird als Energie der Lösung bezeichnet. Dabei ist die Energiefunktion mit der Fitnessfunktion aus dem Genetischen Algorithmus gleichzusetzen. Bessere Lösungskandidaten werden immer als vorläufige Lösung angenommen. Werden in weiteren Iterationen schlechtere Lösungskandidaten generiert, werden diese nur mit einer bestimmten Wahrscheinlichkeit als vorläufige Lösung akzeptiert. Daher ist es wichtig, Lösungskandidaten aus der näheren Umgebung der vorläufigen Lösung zu generieren.

Trifft kein Abbruchkriterium zu, wird der Algorithmus ab Schritt zwei wiederholt.

8.4 Prozessvertrautheit der Mitarbeiter

Die Prozessvertrautheit dient als Schätzwert für die zu erwartende Prozessdauer, die von Mitarbeiter zu Mitarbeiter unterschiedlich ausfällt. Als Parameter der Kostenfunktion wird diese zur besseren Vergleichbarkeit mit der Abfertigungsrate in einer Zeiteinheit ausgedrückt. Daher wird die durchschnittliche Prozessausführungszeit des Mitarbeiters pro Prozess herangezogen. Diese ist in der Mitarbeiterdatenbank in ArangoDB erfasst und wird bei jeder Ausführung des Prozesses aktualisiert. Hierzu nimmt das Kognitive Assistenzsystem nach erfolgter Ausführung des Prozesses einen Datenbankeintrag vor. Da während der Durchführung des Nacharbeitsauftrags der Belief State ständig aktualisiert wird, wird die Dauer pro Prozessschritt nicht zeitgleich zur Ausführung der einzelnen Prozesse notiert, sondern nach Abschluss des Nacharbeitsauftrags. Nach Abschluss des Nacharbeitsauftrags wird die Beobachtungssequenz genutzt, um mittels Viterbialgorithmus die Zustandssequenz zu ermitteln. Die Dauer zwischen den Beobachtungen kann unmittelbar an den Prozess zwischen zwei Zuständen zugeordnet werden.

Für eine gute Zuweisung muss die Vertrautheit des Mitarbeiters mit dem Produkt sowie dem Fehlerfall bekannt sein. Zur Ermittlung der Vertrautheit wird untersucht, wie ein Mitarbeiter in der Vergangenheit mit dem Auftrag umgegangen ist. Die durchschnittliche Ausführungsdauer eines Prozesses wird als Maß für die Vertrautheit genutzt.

Die Vertrautheit V eines Mitarbeiters bezüglich eines Prozesses wird im folgenden Fall durch die durchschnittliche Prozessausführungsdauer D und die Berechtigungen B bestimmt.

$$V = p_{m,j} = D * B \quad (34)$$

Zur Ermittlung der Vertrautheit müssen der Mitarbeiter sowie die Prozesse des Nacharbeitsauftrags bekannt sein. Die Prozesse des Nacharbeitsauftrags ergeben sich durch den bekannten Fehlercode sowie der Vorabermittlung des Nacharbeitsplans bei gegebenem Fehlercode.

8.5 Fallspezifische Anwendung der Metaheuristiken

Die in den vorherigen Kapiteln beschriebene Optimierungsmethode kann für die zuvor benannten Fälle: Initiale Auftragszuweisung, Kontinuierliche Auftragszuweisung und Optimierte Auftragszuweisung zur Variantendurchschleusung genutzt werden. Allerdings sind für alle drei Fälle geringfügige Anpassungen notwendig. Die initiale Auftragszuweisung findet lediglich einmal zu Schichtbeginn statt. Aufgrund der großen Lösungsräume kann sie von der entwickelten Methode profitieren, bedarf jedoch keiner separaten Betrachtung. Die kontinuierliche Auftragszuweisung hingegen muss punktuell angepasst werden.

8.5.1 Kontinuierliche Auftragszuweisung

Zugewiesene Nacharbeitsaufträge werden unterschiedlich schnell abgearbeitet und Neuaufträge treffen kontinuierlich ein. Daher muss die Zuweisung wiederholt erfolgen. Ziel der optimierten kontinuierlichen Auftragszuweisung ist die Reduzierung der Bestände im Lager. Hierzu sind Auftragseingänge (Ankunftsrate der Aufträge λ) sowie Auftragsausgänge (Abfertigungsrate der Aufträge μ) zu berücksichtigen. Der Bestand wird reduziert, wenn gilt: $\mu > \lambda$. Aufgrund der hohen Anzahl manueller Umfänge in der Nacharbeit hängt die Abfertigungsrate von der Leistung der Mitarbeiter ab. Um die Bestandsreduzierung zu erzielen, müssen daher die Aufträge den Mitarbeitern so zugewiesen werden, dass diese optimal abgearbeitet werden. Hierfür müssen sowohl die Anforderungen des Nacharbeitsauftrags als auch die Fähigkeiten der Mitarbeiter bekannt sein. Ein Abgleich zwischen beiden Merkmalen dient der Bestimmung des Nacharbeitsaufwands. Die Nacharbeitsanforderungen des (defekten) Produktes können aus dem Nacharbeitsplan abgeleitet werden. Die Fähigkeiten der Mitarbeiter ergeben sich aus den vom Assistenzsystem festgehaltenen Zeiten pro Prozess. Die in Kapitel 8.3.1 beschriebene Kostenfunktion dient als Schätzmaß für die durchschnittlich benötigte Zeit pro Auftrag. Somit gilt für die Abfertigungsrate $\mu \approx \frac{1}{K(M_v, O)}$. Daher wird bei der Bestandsreduzierung zusätzlich zu der bereits vorliegenden Definition der Kostenfunktion zusätzlich das Kriterium $\mu > \lambda$ überprüft.

8.5.2 Optimierte Auftragszuweisungen zur Variantendurchschleusung

Ein weiteres Ziel besteht in der bevorzugten Bearbeitung bestimmter Varianten in der Nacharbeit. Zwar ist bei einem eingehenden Kundenauftrag die Serienmontage für die

Einhaltung der Stückzahlen verantwortlich, dennoch kann bei kritischer Terminierung der Nacharbeitsbereich bei der Erreichung der Stückzahlen unterstützend tätig werden.

Zur priorisierten Variantendurchschleusung wird der Lösungsraum eingeschränkt. Abbildung 75 zeigt, dass der Lösungsraum so ausgewählt wird, dass die hintersten Produkte immer die bevorzugten Varianten sind. Reihen, die keine priorisierte Varianten enthalten, werden somit ausgeblendet.

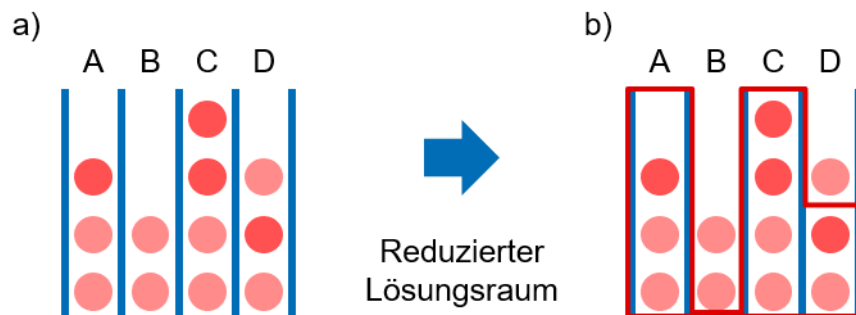


Abbildung 75: Möglichkeiten der Auftragsentnahme für Variantendurchschleusung [Eigene Darstellung]

Mit dieser Reduzierung des Lösungsraums wird verhindert, dass Reihen oder Produktvarianten nachgearbeitet werden, die keinerlei Beitrag zur Durchschleusung der Varianten leisten.

8.6 Zusammenfassung der Methode zur optimierten Auftragszuweisung

In diesem Kapitel wurden zwei unterschiedliche Methoden zur Zuweisung von Nacharbeitsaufträgen an verfügbare Mitarbeiter vorgestellt. Für den Genetischen Algorithmus wurden Restriktionen in Zusammenhang mit dem Lager und ihren Auswirkungen auf die Rekombinationsphase betrachtet. Für Simulated Annealing war diese Beobachtung nicht notwendig. Zudem wurden diese allgemeinen Methoden hinsichtlich der Anwendung bei der initialen Auftragszuweisung, kontinuierlichen Auftragszuweisung und bevorzugten Variantendurchschleusung beschrieben. Zuletzt wurde der Zusammenhang zwischen dem Kognitiven Assistenzsystem sowie den abgeleiteten Mitarbeiterfähigkeiten erläutert, die für die optimierte Auftragszuweisung erforderlich sind. Die beschriebenen Methoden werden in einer Simulationsumgebung getestet, die in Kapitel 10 dargestellt wird.

9 Erprobung und Validierung der Methode in einer Pilot- und Simulationsumgebung

Dieses Kapitel beschreibt die Ergebnisse der Validierung in einem Piloten des kognitiven Assistenzsystems. Das zur Validierung herangezogene Produkt ist ein Automobilmotorge triebe, das in Abbildung 76 dargestellt ist. Es wird hochkant gezeigt, da dies ebenfalls die Ausgangsposition für die im Validierungsszenario stattfindende Nacharbeit ist. Die Komponenten des Assistenzsystems bestehen aus unterschiedlichen Stahl- und Aluminiumlegierungen sowie aus Kunststoff. Das Getriebe lässt unterschiedliche Verbaureihenfolgen von Beginn an zu. Dennoch sind größere Baugruppen aus Montagesicht lediglich aufeinandergestapelt. Aus Gründen der Darstellbarkeit und der Praktikabilität wird das entwickelte Assistenzsystem sowie die dahinterliegende Methode und Datenstruktur an einem ausgewählten Bereich des Getriebes dargestellt.



Abbildung 76: Das betrachtete Produkt ist ein 8 Gang Automatikgetriebe. Die Vorzugslage im Validierungsszenario ist die Aufrichtung. [Eigene Darstellung]

In diesem Abschnitt werden die vier Entwicklungsstufen des Kognitiven Assistenzsystems betrachtet. Diese Fälle sollen repräsentativ einen Nacharbeitsablauf sowie die unterschiedlichen Erfordernisse und Anforderungen während der Nacharbeit darstellen. Die erste Entwicklungsstufe wird hierbei nicht gesondert betrachtet, da diese gemeinsam mit der zweiten Entwicklungsstufe erprobt werden kann, daher bleiben lediglich drei Fälle zur Erprobung:

1. Betrachtung des Standardfalls der Reparatur
2. Betrachtung bei abweichender Vermutung der Fehlerursache
3. Betrachtung der Synchronisation des realen und digitalen Produktzustands

9.1 Betrachtung des Standardfalls der Nacharbeit

Als Standardfall der Assistenz in der Nacharbeit wird der Fall bezeichnet, der im Wesentlichen den Ablauf der Nacharbeit wie ein konventionelles Kognitives Montageassistenzsystem bewerkstelligt. Im Gegensatz zur herkömmlichen Montageassistenz ist im Nacharbeitsfall eine fallabhängige Generierung der Nacharbeitspläne auf Grundlage eingegebener Fehlercodes erforderlich. Nach der Planungs- und Einrichtungsphase erfolgt die Betriebsphase des Assistenzsystems. Dazu scannt ein Mitarbeiter den Fehlercode mit einem Handscanner im Mitarbeiterführungssystem ab. Falls dieser Code menschenlesbar ist, besteht alternativ die Möglichkeit zur händischen Eingabe. Abbildung 77 zeigt die Eingabemaske des Mitarbeiterführungssystems sowie das Fenster zur Eingabe des Fehlercodes.

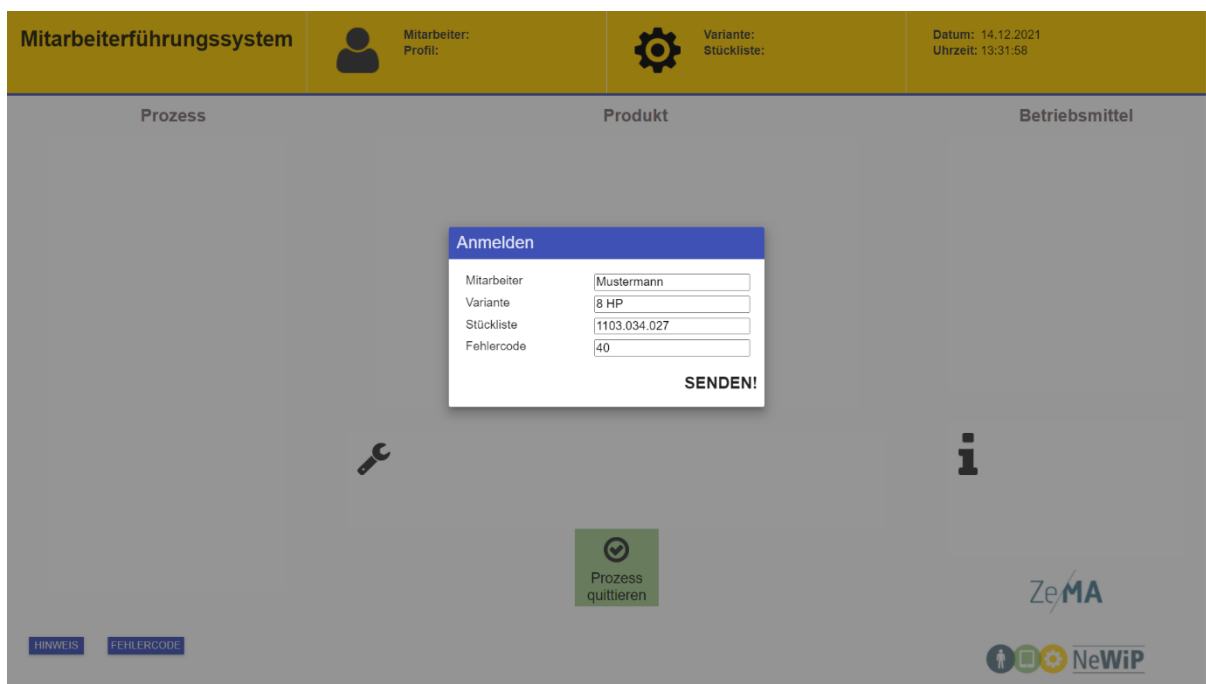


Abbildung 77: Eingabemaske vor Beginn der Auftragsbearbeitung [Eigene Darstellung]

Über eine TCP-Schnittstelle wird der Code an die Steuerung des Assistenzsystems übermittelt. Die Steuerung ermittelt über den Datenbankzugriff auf die ArangoDB den günstigsten Pfad zum fehlerbehafteten Bauteil/ Baugruppe. Dem Mitarbeiter erscheint im Mitarbeiterführungssystem eine aktualisierte Liste der Nacharbeitsprozesse, an der er sich orientieren kann. Zeitgleich werden aus der JSON-Datei pro Prozessschritt die Betriebsmitteldaten ausgelesen und die Betriebsmittel entsprechend parametrisiert und angesteuert. Im Folgenden wird der Ablauf der Reparatur am Mittendifferenzial dargestellt. Diese Baugruppe wird in Abbildung 78 abgebildet.



Abbildung 78: Gegenstand der Betrachtung: Baugruppe Zwischengehäuse-Verteilergewächse ohne die Schrauben zur Befestigung an das restliche Getriebe. [Eigene Darstellung]

Als erstes wird mit einer 3D-Kamera die Verfügbarkeit des nachzuarbeitenden Produktes überprüft. Erst dann wird dem Nacharbeitsplan folgend der Schrauber freigegeben und die Schrauben mit einem Laserprojektor angestrahlt. Das Lösen der Schrauben in der Demontage ist für gewöhnlich nicht an Restriktionen wie der Einhaltung der Schraubreihenfolge bzw. Lösereihenfolge gebunden. Zu Demonstrationszwecken werden bestimmte Funktionalitäten des Schraubers bereits in der Demontagephase aufgerufen. Der Schrauber wird lediglich freigegeben, wenn die vorgesehenen Schrauben anvisiert werden. Dies wird über die Positionsermittlung des Schraubers ermöglicht. Sobald der Schrauber eine nicht vorgesehene Schraube anvisiert, wird der Schrauber umgehend geblockt. Er wird erst dann freigegeben, wenn die vorhergesehene Schraube anvisiert wird und stoppt ggf., wenn die Schraubparameter erzielt wurden (Remontage). Die gelösten Schrauben werden dann entsorgt. Die Entsorgung muss sichergestellt werden, damit die Schrauben nicht wiederholt verbaut werden können. Dazu wird ein Behälter verwendet, der entweder auf einer Waage abliegt oder eine Waage integriert hat. Der Prozess wird erst dann als „erfolgreich durchgeführt“ bewertet, wenn alle Schrauben von der Waage erfasst wurden. Diese können nacheinander oder getrennt in den Behälter abgelegt werden. In weiteren Schritten erfolgt die Entnahme von Bauteilen aus den Produkten. Die Entnahme selbst wird durch einen Bildabgleich überprüft. Im folgenden Schritt muss das demontierte Bauteil gezielt abgelegt werden. Die gezielte Ablage des Bauteils kann in der Demontage notwendig sein, da bei der Remontage die Verschleppung von Schmutz zu vermeiden ist. Bei freier Ablagemöglichkeit wäre dies nicht sichergestellt. Zur Überprüfung ist ein Positionstracker am Arbeitshandschuh angebracht. Legt der Mitarbeiter das Bauteil auf die

vorhergesehene Fläche ab, muss er zwangsweise mit der Hand den vordefinierten virtuellen Arbeitsraum durchdringen. Wird der Positionstracker am Handschuh durch die Kamera erkannt, wird die ordnungsgemäße Positionierung des Bauteils angenommen und der Prozess als planmäßig durchgeführt markiert. Ebenso werden die restlichen Prozesse bzw. Produktzustände nach Vorgabe durchgeführt und abgeprüft, bis der Produktzustand, an dem der Fehler behoben werden soll, erreicht ist.

Im nächsten Schritt findet die eigentliche Fehlerbehebung statt. Hierbei wird im Rahmen des Beispielfalles ein bisher verbautes Abtriebsrad durch ein neues Abtriebsrad ersetzt. Das ausgebaute Abtriebsrad wird in einer Waagschale abgestellt und somit entsorgt. Das neue, zu verbauende Abtriebsrad wird aus der Materialbereitstellung entnommen und in das Getriebe eingesetzt. Die Überprüfung der einzelnen Prozesse findet mit den bereits vorgestellten Betriebsmitteln statt.

Auf die Fehlerbehebung folgt die Remontage. Hierbei werden die in der Demontagephase ausgebauten Bauteile oder Baugruppen in umgekehrter Reihenfolge wieder montiert. Die entsorgten Bauteile werden durch neue Bauteile, bspw. neue Schrauben ersetzt. Auch hierbei erfolgt eine Entnahmekontrolle durch die Positionskontrolle. Der Vorgang ist erst abgeschlossen, wenn das Getriebe komplettiert wurde.

9.2 Berücksichtigung der Erfahrungsgrade des Mitarbeiters

Der Erfahrungsgrad des Mitarbeiters ist ein Maß für die Vertrautheit des Mitarbeiters mit einem Nacharbeitsprozess. Zur adäquaten Unterstützung des Mitarbeiters müssen Informationen dosiert eingeblendet werden können. Zwar wird in dieser Arbeit nicht die psychologisch geeignete Darstellungsform der Informationen betrachtet, dennoch stellt das Assistenzsystem Funktionen bereit, um Mitarbeitern zugeschnittene Informationen zur Verfügung zu stellen. Dazu bietet das Assistenzsystem zwei Möglichkeiten. Bei der ersten Möglichkeit werden Mitarbeiter einer Erfahrungsgruppe zugewiesen. Diese Methode ist in der Umsetzung schneller und praktikabler. Dabei werden Mitarbeiter in die Erfahrungsgruppen Anfänger oder Experte eingeteilt. Auch Zwischenstufen sind möglich. Bei der Anmeldung am Arbeitsplatz ruft das Assistenzsystem die dem Mitarbeiter zugewiesene Erfahrungsgruppe ab. Einem Anfänger werden alle zum Prozess vorhandenen Informationen eingeblendet. Hierbei kann es zu Redundanzen kommen. Bei einer Verschraubung wird bspw. im Mitarbeiterführungssystem in textueller, aber auch in bildlicher Form die Informationen zum Anziehen der 12 Schrauben am Zwischengehäuse des Getriebes angezeigt. Zusätzlich werden dem Mitarbeiter die Verschraubpunkte mittels Laserprojektor angestrahlt. Einem als „Experte“ eingestuften Mitarbeiter werden Informationen in reduziertem Umfang ohne Redundanz eingeblendet. Im vorliegenden Fall werden dem Mitarbeiter im Verschraubungsprozess keine Projektionen dargestellt. Ihm liegen die Informationen in textueller Form im Mitarbeiterführungssystem vor. Jedoch kann der Experte darüber explizit Zusatzinformationen anfordern.

Die zweite Möglichkeit teilt Mitarbeiter nicht in feste Erfahrungsgruppen ein, vielmehr wird die Vertrautheit einzelner Nacharbeitsprozesse pro Mitarbeiter bewertet. Informationen werden prozessspezifisch eingeblendet. Die Vertrautheit mit einem Prozess ist über die Fehlerrate, Zeitdauer im Vergleich zu anderen Mitarbeitern sowie der letzten Ausführung zu bestimmen. Die zweite Möglichkeit ist aufgrund der erhöhten Datenhandhabung aufwendiger in der Umsetzung. Statt einer einfachen Zuweisung der Mitarbeiter in Erfahrungsgruppen werden für jeden Mitarbeiter einzelne Prozesse bewertet.

9.3 Betrachtung Neuermittlung des Nacharbeitsplans

Der nächste Fall betrachtet die Abweichung vom ursprünglichen Nacharbeitsplan. Hierbei wird dem Mitarbeiter die Möglichkeit eingeräumt, einer anderen Fehlerursachenvermutung nachzugehen.

Dazu muss dieser den entsprechenden Fehlerursachencode eingeben. Hierzu kann der Mitarbeiter im Mitarbeiterassistenzsystem einen Button betätigen, woraufhin ein Eingabefeld zur Eingabe des neuen Fehlercodes erscheint. Nach der Eingabe wird der neue Nacharbeitsplan ausgehend vom aktuellen Produktzustand ermittelt. In Abbildung 79 ist zu sehen, dass der Mitarbeiter bereits die Schritte Verschraubung VG lösen, Schrauben entsorgen, VG entnehmen, VG ablegen und TD entnehmen ausgeführt hat. Zu diesem Zeitpunkt gibt er einen neuen Fehlercode ein. Der Fehlercode kann als bekannt angenommen werden, da dem Mitarbeiter Tabellen zur Verfügung stehen, aus denen Bauteil und Fehlerursache sowie zugehöriger Fehlercode zur Verfügung stehen.

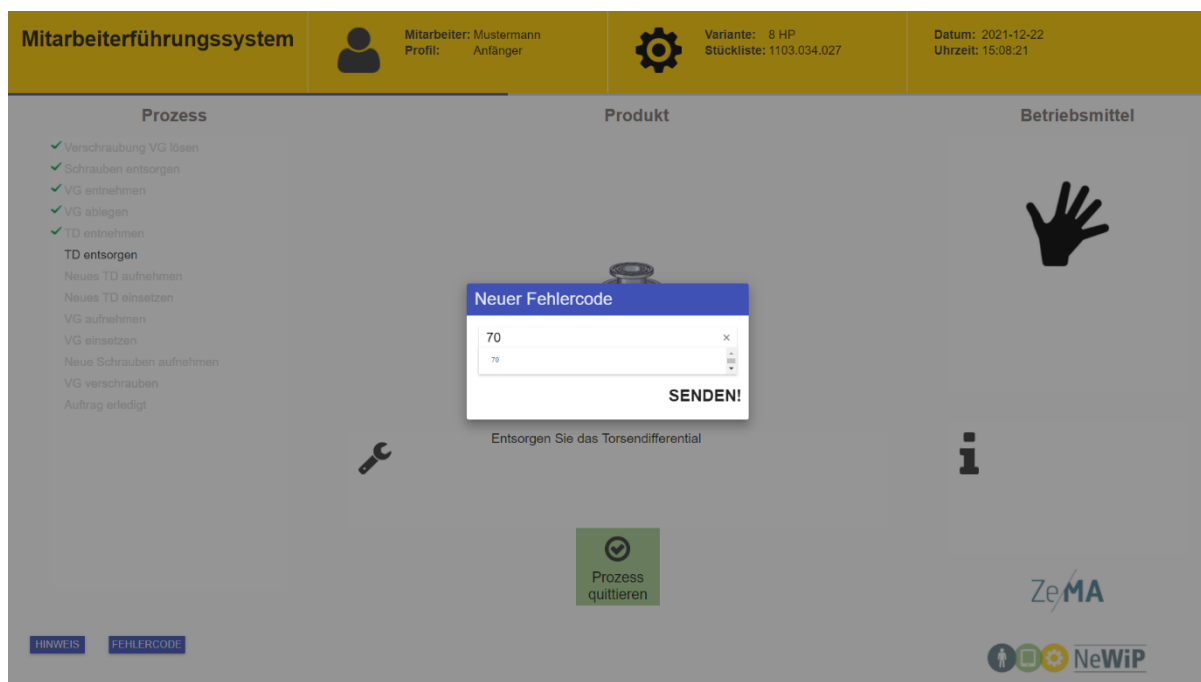


Abbildung 79: Eingabe eines neuen Fehlercodes [Eigene Darstellung]

Bedeutsam für die Validierung am Demonstrator ist die Eingabe des Fehlercodes in verschiedenen Abschnitten des Nacharbeitsplans. Diese sind:

1. Die Demontagephase
2. Die Fehlerbehebungsphase
3. Die Remontagephase

Im ersten Validierungsbeispiel wird nach Eingabe des Fehlercodes ein Nacharbeitsplan ausgegeben und das Assistenzsystem initialisiert. Nachdem einige Demontageprozesse unternommen wurden, gibt der Mitarbeiter in einem beliebigen Produktzustand den Fehlercode gemäß seiner (abweichenden) Vermutung ein. Daraufhin wird beginnend am aktuellen Produktzustand der neue Nacharbeitsplan ausgegeben. Der neue Nacharbeitsplan kann unter Umständen Remontageprozesse vor der Fehlerbehebungsphase beinhalten, wenn zur Fehlerbehebung eine geringere Ausbaustufe genügt. Liegt der bisher beschrittene Pfad (bisher getätigten Prozesse) auf dem neuen Pfad, wird die Remontage erst nach der Fehlerbehebung notwendig. Abbildung 80 zeigt den Start des neuen Nacharbeitsplans ab dem Prozess, ab dem das Torsendifferential wieder eingesetzt werden soll. In den folgenden Schritten werden Prozesse wieder rückgängig gemacht. In diesem Beispiel ist das komplette Produkt zu remontieren. Danach wird die gesamte Baugruppe mit dem Lösen der 12 Schrauben demontiert. Es folgen weitere notwendige Prozesse sowie die Fehlerbehebung. Zuletzt wird die Remontage durchgeführt.

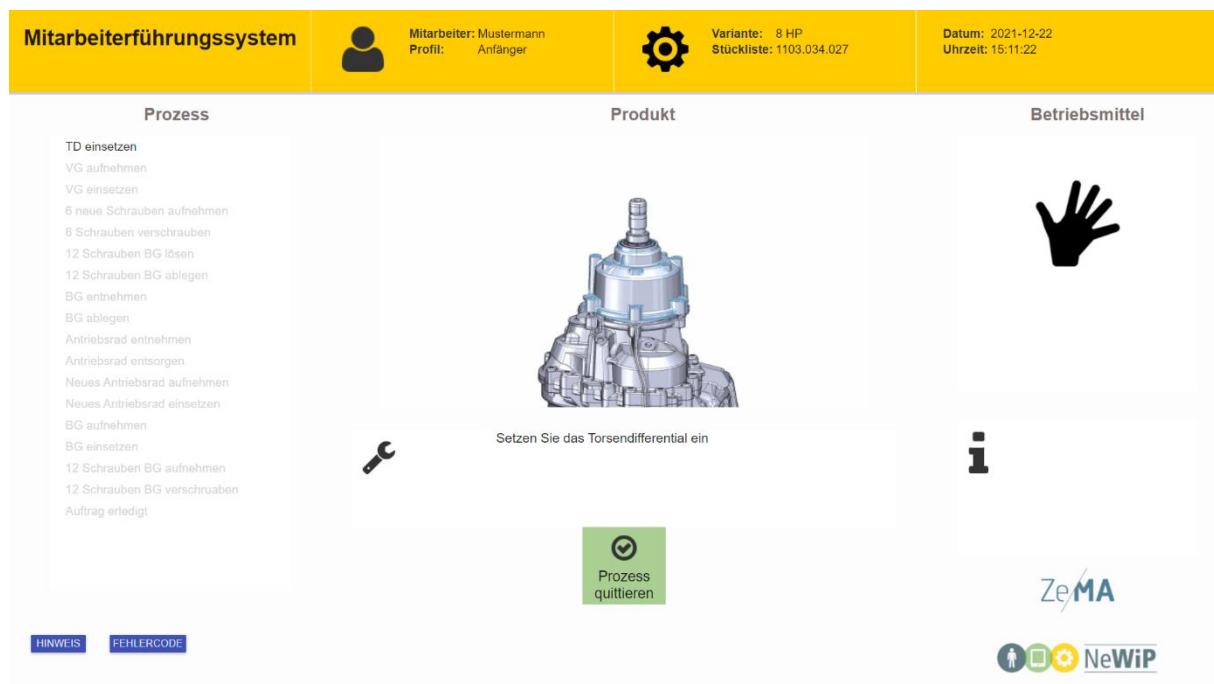


Abbildung 80: Ausgabe des neuen Arbeitsplans ~~ab~~ dem Zustand der Fehlercodeeingabe [Eigene Darstellung]

Im zweiten Fall, der Fehlerursachenvermutung in der Fehlerbehebungsphase, wird erneut ein Fehlerursachencode eingegeben. Dadurch, dass Fehlerbehebungsprozesse

nicht Teil des Produktzustandsgraphen sind, sondern bei der Ermittlung des Nacharbeitsplans in den Scheitelpunkt des Plans eingegeben werden, ist eine direkte Neuermittlung des Pfades nicht möglich. Daher muss der Fehlerbehebungsprozess erst abgeschlossen werden, bis der Neuvermutung nachgegangen werden kann.

Die Eingabe des Fehlercodes in der Remontagephase wiederum erlaubt ähnlich wie in der Demontagephase die Ermittlung basierend auf den Produktzustandsgraphen.

9.4 Betrachtung der Abweichung vom vorgesehenen Nacharbeitsplan

Bisher wurde ein Assistenzsystem vorgestellt, das lediglich vorgegebene Arbeitsschritte gemäß einem digital erstellten Arbeitsplan erwartet. Jedoch kann der Mitarbeiter ohne jegliche Ankündigung (Eingabe eines neuen Fehlercodes) vom vorgesehenen Arbeitsplan abweichen. Aufgabe des Assistenzsystems ist damit ebenfalls, die Synchronisation des digitalen Zustands mit dem realen Zustand sicherzustellen. Die Verwendung der Wahrscheinlichkeitsmethoden mittels Markov-Ketten wird in zwei Etappen genutzt. Zum einen wird lediglich der Belief State nachverfolgt. Zum anderen wird aufbauend auf den Belief State eine Handlungsempfehlung ausgegeben.

9.4.1 Trainingsphase der vierten Entwicklungsstufe in einer Simulationsumgebung

Zur Ermittlung der HMM-Parameter λ werden Beobachtungen aus der Nacharbeit genutzt. Dabei ist die Initialverteilung π so definiert, dass der komplettierte Produktzustand immer der Startzustand ist. Für die weiteren Wahrscheinlichkeiten werden Trainingsdaten aus der Nacharbeit generiert. Dazu wurde im Rahmen dieser Arbeit eine Simulationsumgebung genutzt. Die Simulationsumgebung bildet mögliche Beobachtungssequenzen ab. Da die Wahrscheinlichkeiten pro Fehlercode erstellt werden, werden zur Erstellung möglicher Beobachtungssequenzen alle möglichen fehlerrelevanten Nacharbeitspfade herangezogen. Diese werden dann in einen Graphen zusammengefasst. Für die Übergangswahrscheinlichkeit $T(s'|s)$ kann initial die Annahme getroffen werden, dass der korrekte Pfad mit einer höheren Wahrscheinlichkeit eingeschlagen wird. Zur Bestimmung der initialen Wahrscheinlichkeitsverteilung über alle möglichen Produktzustände werden die Zustände in einen bevorzugten Zustand s'_{pref} und zu vermeidende Zustände (restliche Produktzustände) s'_{avoid} eingeteilt.

$T(s'_{pref}|s)$ bezeichnet die Übergangswahrscheinlichkeit zum bevorzugten nächsten Produktzustand, der auf dem ermittelten Pfad liegt. Für die Berechnung werden zunächst die Wahrscheinlichkeiten über alle nächstmöglichen Produktzustände gleichverteilt. Anschließend werden pro zu vermeidendem Nachfolgezustand der zur Gesamtzahl möglicher Nachfolgezustände quadrierte Bruchteil hinzuaddiert, wie Gleichung (35) zeigt.

$$T(s'_{pref}|s) = \frac{1}{|s'|} + \frac{|s''_{avoided}|}{|s'|^2} \quad (35)$$

Für die Übergangswahrscheinlichkeit der zu vermeidenden Produktzustände wird ebenfalls zunächst die anteilige Wahrscheinlichkeitsverteilung genommen, allerdings werden für alle s'_{avoid} die Teiler aus der quadrierten Anzahl aller Nachfolgezustände abgeleitet und der entsprechende Anteil subtrahiert, wie Gleichung (36) zeigt.

$$T(s'_{avoid}|s) = \frac{1}{|s'|} - \frac{1}{|s'|^2} \quad (36)$$

Die Berechnungen der initialen Übergangswahrscheinlichkeiten betrachten weder Vorlieben des Mitarbeiters noch sonstige allgemeine Fehlertrends. Allerdings ist es anhand der Formel möglich, eine Wahrscheinlichkeitsverteilung zu generieren, die über alle möglichen Produktzustände die Summe 1 liefert, wie aus Gründen der Vollständigkeit Gleichungen (37)-(42) zeigen.

$$T_{pref} + T_{avoid} * |s'_{avoid}| = 1 \quad (37)$$

$$\frac{1}{|s'|} + \frac{|s''_{avoided}|}{|s'|^2} + \left(\frac{1}{|s'|} - \frac{1}{|s'|^2} \right) * |s'_{avoid}| = 1 \quad (38)$$

$$\frac{1}{|s'|} + \frac{|s''_{avoided}|}{|s'|^2} + \frac{|s'_{avoid}|}{|s'|} - \frac{|s'_{avoid}|}{|s'|^2} = 1 \quad (39)$$

$$\frac{1 + |s'_{avoid}|}{|s'|} = 1 \quad (40)$$

Da die folgende Beziehung gilt: $|s'| = |s'_{pref}| + |s'_{avoid}|$

und $s'_{pref} = 1$ gilt, da immer nur ein Nachfolgezustand bevorzugt wird, gilt auch:

$$\frac{1 + |s'| - |s'_{pref}|}{|s'|} = \frac{1 + |s'| - 1}{|s'|} = 1 \quad (41)$$

$$\frac{|s'|}{|s'|} = 1 \quad (42)$$

Dabei wird angenommen, dass der kürzeste Pfad die größere Wahrscheinlichkeit hat, beschritten zu werden. Abbildung 81 zeigt die Erstellung von Fehlercodegraphen mittels Zusammenfassung von fehlercodespezifischen Pfaden. Beginnend beim Startpunkt A wird zufällig ein Nachbarknoten ausgewählt, wobei die Auswahlwahrscheinlichkeit nicht gleichverteilt ist. In mehreren Iterationen wird eine gewichtete Durchmischung der Trainingsdaten pro Fehlercode erreicht.

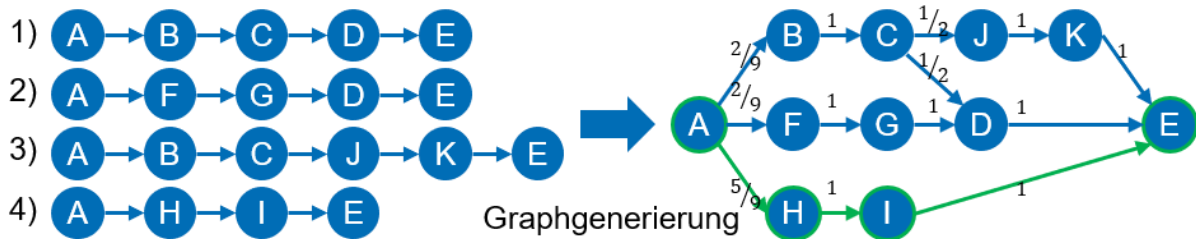


Abbildung 81: Erstellung des Fehlercodegraphen zur Generierung der Trainingsdaten [Eigene Darstellung]

In der Simulationsumgebung werden drei Arten von Beobachtungs- bzw. Transitions-matrizen unterschieden.

1. Ideale Matrizen: Diese repräsentieren die „wahre“, aber in der Regel unbe-kannte Beobachtungs- bzw. Transitionsmatrix und werden nicht trainiert.
2. Initiale Matrizen: Diese werden entsprechend bestimmter Regeln oder auch zu-fällig initiiert und müssen während der Laufzeit des Kognitiven Assistenzsys-tems trainiert werden.
3. Trainierte Matrizen: Diese repräsentieren die Beobachtungs- bzw. Transitions-matrix, nachdem sie durch den Baum-Welch-Algorithmus trainiert wurden. Der Baum-Welch-Algorithmus ist ein Expectation-Maximization-Algorithmus, der sich des Forward-Backward-Algorithmus bedient, wie in Kapitel 6.4.1.1.2 und 6.4.1.1.3 beschreiben wurde. Die trainierten Matrizen werden im Assistenzsys-tem der Entwicklungsstufe 4 genutzt.

Für die Simulationsumgebung werden die Pfade automatisch generiert. Die Überset-zung in die Beobachtungssequenz kann nicht automatisch erfolgen, da bestimmte Be-obachtungen nur bei speziellen Transitionspfaden möglich sind. Diese Regeln lassen sich allerdings nicht von der Beobachtungsmatrix ableiten. Daher werden in der Simu-lationsumgebung die Beobachtungspfade manuell erstellt. Aus diesen ergibt sich dann die ideale Beobachtungsmatrix durch mengenmäßigen Abgleich zwischen Zustand und Beobachtung und anschließender Normierung der Matrix.

Abbildung 82 zeigt die ideale Transitionsmatrix bzw. die ideale Beobachtungsmatrix für einen Nacharbeitsplan, der zum Produktzustand 9 reicht. Nach dem Training sollten die trainierten Matrizen lediglich geringfügige Abweichungen zu den entsprechenden idealen Matrizen aufweisen.

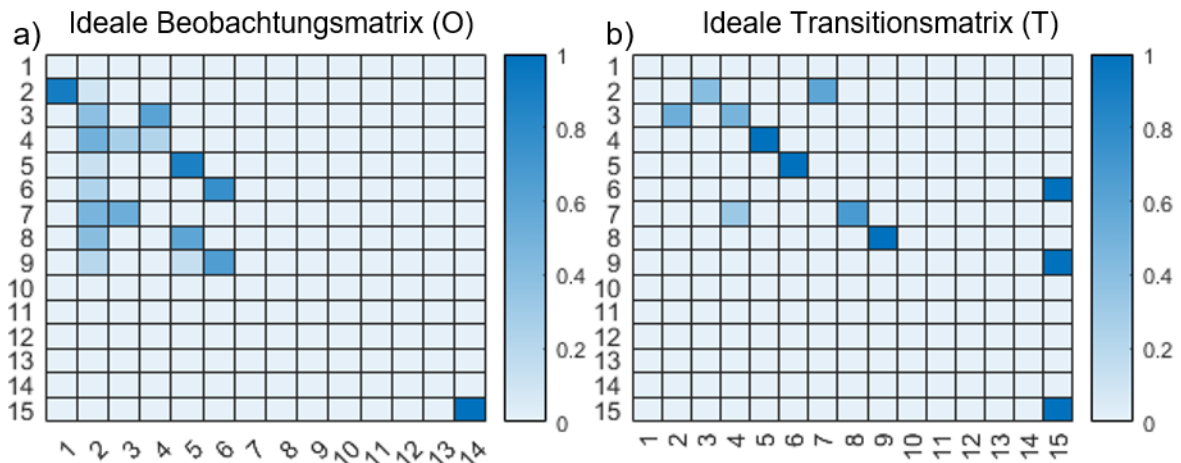


Abbildung 82: a) Ideale Beobachtungsmatrix b) Ideale Transitionsmatrix [Eigene Darstellung]

Abbildung 83 zeigt die Abweichung zwischen den initialen Matrizen und den idealen Matrizen sowie den trainierten Matrizen und den idealen Matrizen. Im Diagramm stellt die Intensität der Blautönung die Absolute Abweichung dar. Es ist erkennbar, dass nach dem Training bis auf wenige Felder die trainierten Matrizen nur geringfügige Abweichungen zu den idealen Matrizen aufweisen. Die intensiven Farben insbesondere in der trainierten Transitionsmatrix sind auf die Irrelevanz des theoretischen Pfadabschnitts zurückzuführen. Da der dunkelgefärbte Bereich der Matrix die Zustände wiedergibt, die nicht in den Trainingspfaden erhalten waren, wurden diese Bereiche auch nicht verbessert.

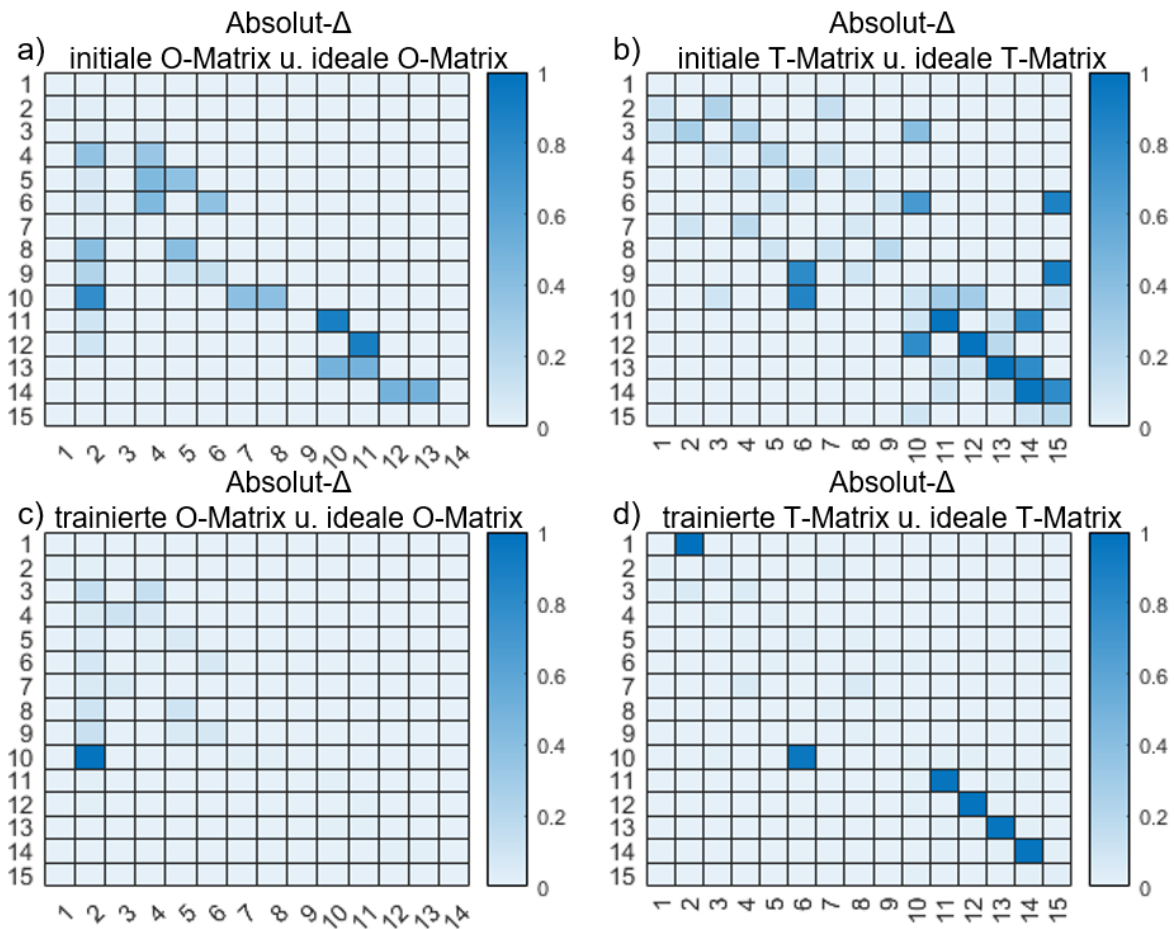


Abbildung 83: Abweichung der trainierten Matrizen und der idealen Matrizen [Eigene Darstellung]

Abbildung 84 zeigt die Beobachtungs- und Transitionsmatrizen vor und nach dem Training. Die Beobachtungsmatrizen führen in den Spalten die Beobachtungen auf und in den Zeilen die Produktzustände. Die Transitionsmatrizen führen in den Zeilen den Ausgangszustand und in den Spalten den Folgezustand auf. Insbesondere an Abbildung 84d lässt sich exemplarisch zeigen, wie nach dem Training bestimmte Nachfolgezustände als wahrscheinlicher angesehen werden. Diese Matrizen gelten pro Fehlercode. Da sich die Pfade in der Nacharbeit aber überschneiden, können sich auch Trainingsmatrizen überlagern, sodass nicht für jeden Nachbearbeitungspfad die Transitionsmatrix von Beginn an trainiert werden muss. Auch anhand der Beobachtungsmatrix kann exemplarisch das Erscheinen bestimmter Beobachtungen hervorgehoben werden. Abbildung 84c ist im Vergleich zu Abbildung 84a weniger diffus. Trotz der eindeutigen Ergebnisse nach dem Training gilt, dass die Matrizen, sofern eine reale Möglichkeit besteht andere Beobachtungen zu machen oder Nachfolgezustände zu haben, keine Nulleinträge an den entsprechenden Stellen beinhalten dürfen, da ansonsten der Algorithmus Fehler auswirft, weil bspw. ganze Kanten gelöscht werden. Daher werden Nulleinträge, sofern diese in der Initialmatrix nicht vorhanden waren, durch sehr kleine Werte ersetzt. Abbildung 84e und Abbildung 84f zeigt die Änderungen, die sich nach dem Training zwischen initialen und trainierten Matrizen ergeben.

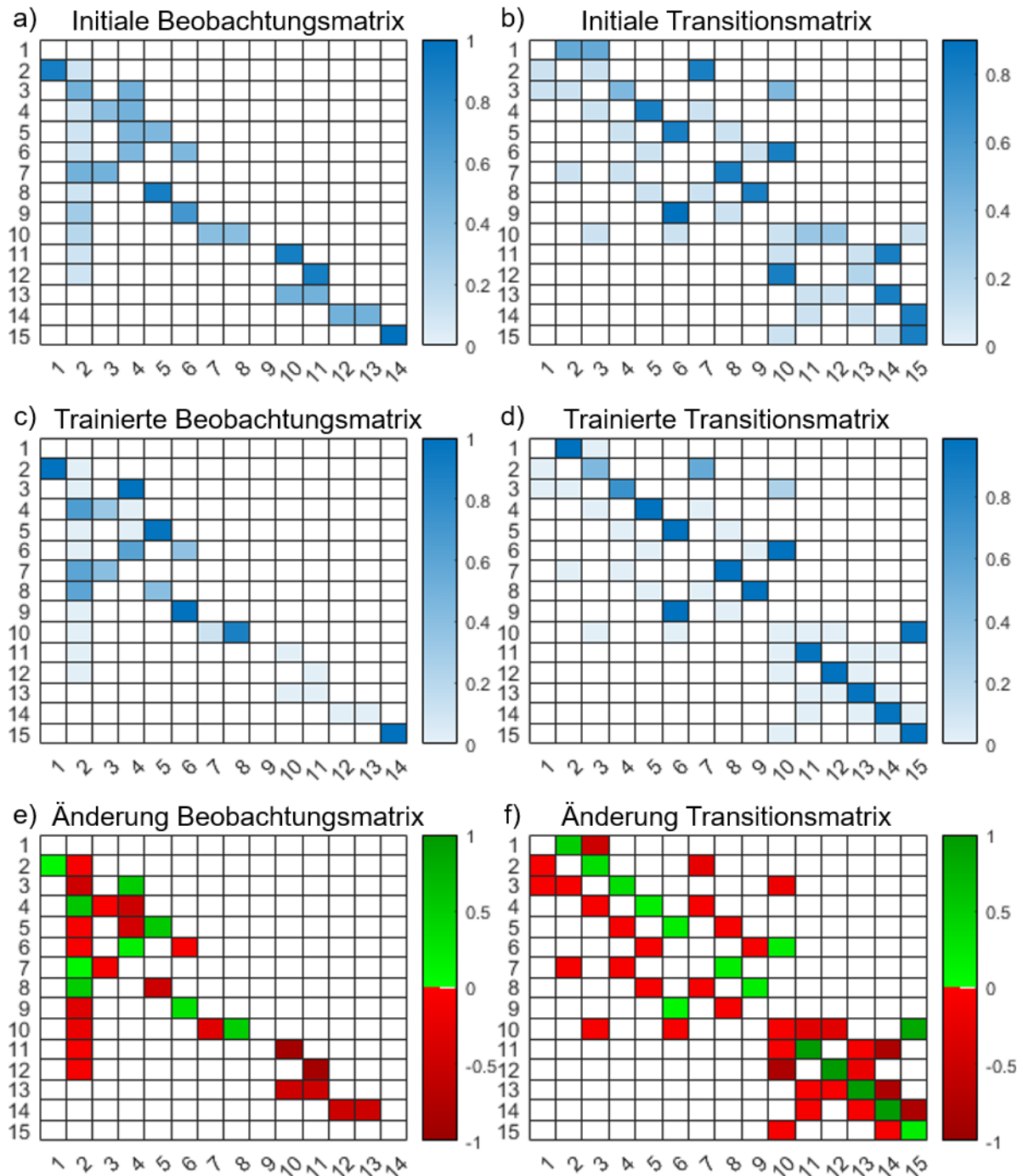


Abbildung 84: Beobachtungs- und Transitionswahrscheinlichkeiten vor und nach dem Training [Eigene Darstellung]

9.4.2 Betriebsphase der vierten Entwicklungsstufe

Die Nachverfolgung des Belief States wird am Beispiel der Demontage am Mittendifferential demonstriert. Zur übersichtlicheren Darstellung werden die Nacharbeitsprozesse auf 13 Zustände beschränkt. Damit die Produktzustände in der Beschreibung

nachvollziehbar bleiben, werden in der folgenden Beschreibung Bauteile und Baugruppe statt Binärvektoren verwendet. Tabelle 9 listet die genutzten Abkürzungen für die Bauteile oder Baugruppen.

Tabelle 9: Genutzte Abkürzungen zur Zustandsbeschreibung des Produktes

G	Getriebe
S6	6 Schrauben
S12	12 Schrauben
VG	Verteilergehäuse
TD	Torsendifferential
AN	Antriebsrad
AB	Abtriebsrad
Plattform	Getriebezustand ohne G, VG, TD, 12S, und 6S

In Tabelle 10 werden die in diesem Beispiel berücksichtigten Produktzustände aufgelistet und den Buchstaben aus Abbildung 85 zugeordnet. Bspw. bedeutet der Zustand zu G_S12: Komplettes Getriebe **ohne** 12 Schrauben. Zustand G_S12_S6 bedeutet: Komplettes Getriebe ohne 12 Schrauben und ohne 6 Schrauben. Aus der Zustandsbezeichnung wird die Demontagesequenz nicht ersichtlich.

Tabelle 10: Zustände aus Abbildung 85

1	Start	9	G_S6_VG_TD
2	G	10	Plattform
3	G_S12	11	Plattform_AN
4	G_S12_S6	12	Plattform_AB
5	G_S12_S6_VG	13	Plattform_AN_AB
6	G_S12_S6_VG_TD	14	Leer
7	G_S6	15	Auftrag abgeschlossen
8	G_S6_VG	---	

Abbildung 85 zeigt einen beispielhaften Verlauf des Belief States, während der Mitarbeiter durch den Prozess schreitet. In der linken Spalte ist der Verlauf im untrainierten Fall zu sehen. Die rechte Spalte zeigt den Verlauf des Belief States nach dem Training an. In der ersten Zeit ist die Verteilung des Belief States eindeutig, wie in Abbildung 85a-d zu sehen ist. Dies lässt sich durch produktspezifische Restriktionen erklären.

Hierbei erfolgt, nachdem der Initialzustand erkannt wurde, der per Definition beim zusammengebauten Produkt beginnt (Abbildung 85a, b), die Demontage von 12 Schrauben mit einem Elektronenschrauber. Danach wird ein Irrtum im Nacharbeitsprozess erkannt und der Mitarbeiter schlägt einen anderen Weg ein. Er demontiert in diesem Fall die 6 Schrauben des Verteilergehäuses, wobei sich ab diesem Zeitpunkt der Belief State breiter auffächert, wie in Abbildung 85e und f zu sehen ist. Dabei weist der Belief State aufgrund der initialen Gewichtung der Transitions- und Beobachtungsmatrizen eine Tendenz zu Zustand 7 auf. Da die Sicherheit in diesem Fall ausreichend hoch ist, wird von der Steuerung des Assistenzsystems keine Anfrage zur Klarstellung des Belief State angesteuert. Im trainierten Fall ist der Zustand weniger deutlich. Daher wird von der Steuerung des Assistenzsystems eine Abfrage im Mitarbeiterassistenzsystem durchgeführt. Darauf basierend wird der Belief State, wie in Abbildung 85h dargestellt, korrigiert und weist nun eine höhere Sicherheit auf in Zustand 3 befindlich zu sein. Die Unsicherheit in Abbildung 85f könnte als Fehlleistung des Algorithmus interpretiert werden, allerdings ist die Transitionswahrscheinlichkeit in diesem Fall ebenfalls nicht eindeutig, wie Abbildung 82a zeigt. Daher ist die Abfrage an dieser Stelle erforderlich und in Übereinstimmung mit den idealen Matrizen. Da die Abfrage allerdings im untrainierten Fall nicht erfolgt ist, wird von einem falschen Produktzustand (7) ausgegangen. Basierend auf dem falsch angenommenen Zustand ist auch der Folgezustand mit einer großen Unsicherheit behaftet, was aber in diesem Fall mittels einer Mitarbeiterabfrage korrigiert werden kann.

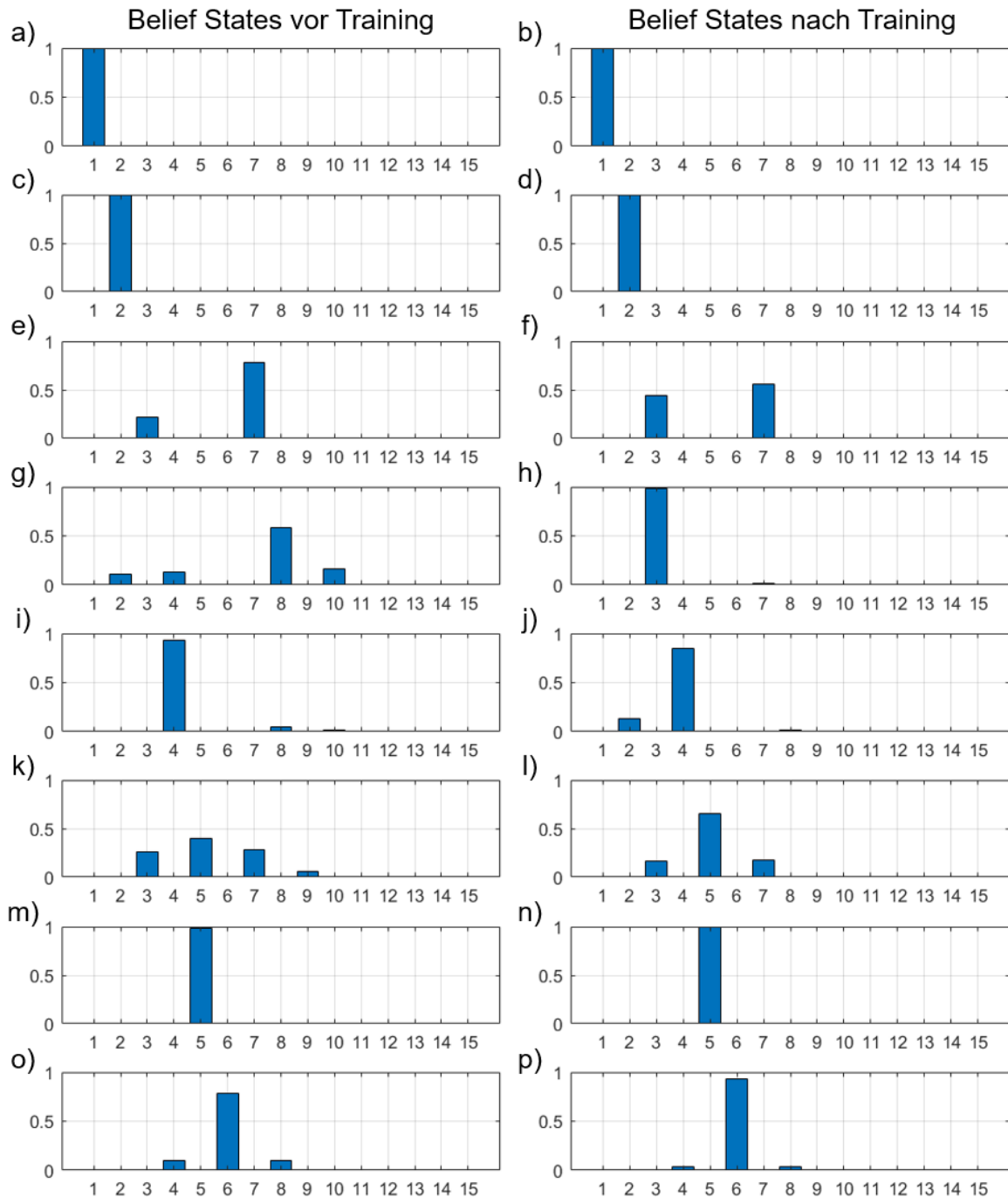


Abbildung 85: Verteilung des Belief State während der Durchführung der Nacharbeitsprozesse
[Eigene Darstellung]

Basierend auf den Belief States werden Handlungsalternativen vorgeschlagen. Dabei wird eine Risikoabwägung durchgeführt. Der Belief State wird als Gewichtung zur Feststellung der Risiken aus diesem Zustand heraus genutzt. Das Assistenzsystem wägt zwei Aktionen ab und spricht sich für eine Handlungsempfehlung aus. Die erste Option ist die Fortsetzung der Nacharbeitsprozesse. Die zweite Option besteht aus der Bestätigung des aktuellen Produktzustands in Form einer Nutzeranfrage. Zur Ermittlung der besten Handlungsanweisung wird die Standardabweichung für diskrete Wahrscheinlichkeitsverteilungen genutzt. Zur Ermittlung der Nützlichkeit einer Handlung

wird die Wahrscheinlichkeitsverteilung über den Belief State herangezogen. Je breiter der Belief State gestreut ist, desto schlechter auch der Reward. Grund hierfür ist, dass bei breiter Streuung große Unsicherheit über den realen Zustand vorliegt. In diesem Fall wird eine Nutzeranfrage eingeleitet, um den Belief State einzugrenzen. Dies geschieht über die Ermittlung der Standardabweichung.

Da aber die Sicherheit im Belief State zu gering ausfällt, schlägt der Agent der Steuerung vor, eine Abfrage im Mitarbeiterführungssystem durchzuführen. Hierbei wird explizit nach einem Zustand gefragt. Nach erfolgter Rückmeldung aktualisiert der Agent den Belief State. Dies resultiert in einer wesentlich größeren Tendenz Richtung Produktzustand 8, wie in Abbildung 85d zu sehen ist.

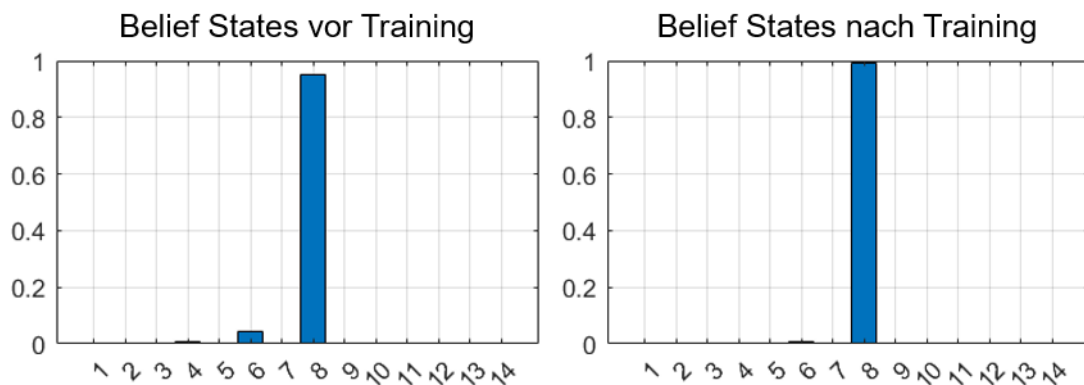


Abbildung 86: Die Verteilung des Belief State nach Mitarbeiter-Rückmeldung [Eigene Darstellung]

Die Übergangs- und Beobachtungswahrscheinlichkeiten wurden bisher als gegeben angenommen. Allerdings müssen diese während der Nacharbeit in den vorhergehenden Entwicklungsstufen ermittelt werden, wie in Kapitel 9.4.1 beschrieben wird.

10 Simulation und Validierung der Auftragszuweisung

Die Resultate der Methode zur optimierten Auftragszuweisung werden in einer Simulationsumgebung validiert. Diese wird in der Programmiersprache *Python* umgesetzt. Anhand des Bestandsverlaufs der Bereitstellungsfläche wird ein Vergleich zwischen zufälliger und durch die Metaheuristik ermittelter Auftragszuweisung gezogen.

10.1 Aufbau der Simulationsumgebung

Die Simulationsumgebung ist in der Programmiersprache *Python3* umgesetzt. Die Simulation läuft in folgenden Schritten ab:

1. Definition der Eingabeparameter
 - a. Mitarbeiterzahl samt Erfahrungsgrad und Berechtigungen
 - b. Schichtdauer
 - c. Ereignismodellierung
2. Ermittlung der besten Zuweisungskombination
3. Aktualisierung der Bereitstellungsfläche

Diese Schritte werden in einer Schleife durchgeführt, bis ein Abbruchkriterium für die Simulation erreicht wurde. Dieses Abbruchkriterium ist nicht mit dem Abbruchkriterium der Metaheuristik zu verwechseln. Im Gegensatz zur Metaheuristik ist ein Abbruchkriterium der Simulation, dass die Arbeitsschicht vorbei ist und die Simulation eine Arbeitsschicht durchgespielt hat. Vor dem Start der Simulationsumgebung werden die Parameter „verfügbare Mitarbeiter“ sowie die „verfügbare Arbeitszeit“ definiert.

Die Ereignismodellierung betrifft die Ankunftsrate der Nacharbeitsprodukte in das Lager. Eine stochastische Bestimmung genügt, da hier die Annahme zugrunde liegt, dass zwar jeder Eingang einzeln betrachtet variieren kann, eine größere Zahl von Eingängen aber einer modellierbaren Regelmäßigkeit folgt. Die Ankunftsgröße kann auf mehrere Art und Weise angegeben werden. Zum einen wird die Zahl der Ereignisse pro Zeitintervall angegeben. Diese Betrachtungsweise kann durch die Poissonverteilung modelliert werden, siehe Abbildung 87a. Gemäß Formel (4) muss hierfür lediglich die mittlere Anzahl der Ereignisse bekannt sein. Oftmals können diese Zahlen allein aus der Erfahrung ermittelt werden. Zum anderen kann die Zeit zwischen zwei Ereignissen angegeben werden. Hierfür eignet sich die Erlangverteilung, wie sie für die Modellierung von Bedien- und Wartezeiten üblich ist [Vgl. 159 S.148]. Abbildung 87b stellt die Eingänge in das Lager erlang-verteilt dar. Gemäß Formel (5) ist der Formparameter k vorgegeben. Die Zeit zwischen den Ereignissen kann variieren. Die Dichtefunktion der Erlangverteilung wird vor dem Hintergrund der Simulationsumgebung als die Wahrscheinlichkeitsverteilung über die Zeitabstände für die Anzahl bestimmter Ereignisse interpretiert. Dabei steht $x \in X$ für den zeitlichen Abstand zwischen den Ereignissen. Beispielsweise gilt $k = 2$ für das Abwarten genau zweier Ereignisse. λ ist der Kehrwert der Ereignisrate.

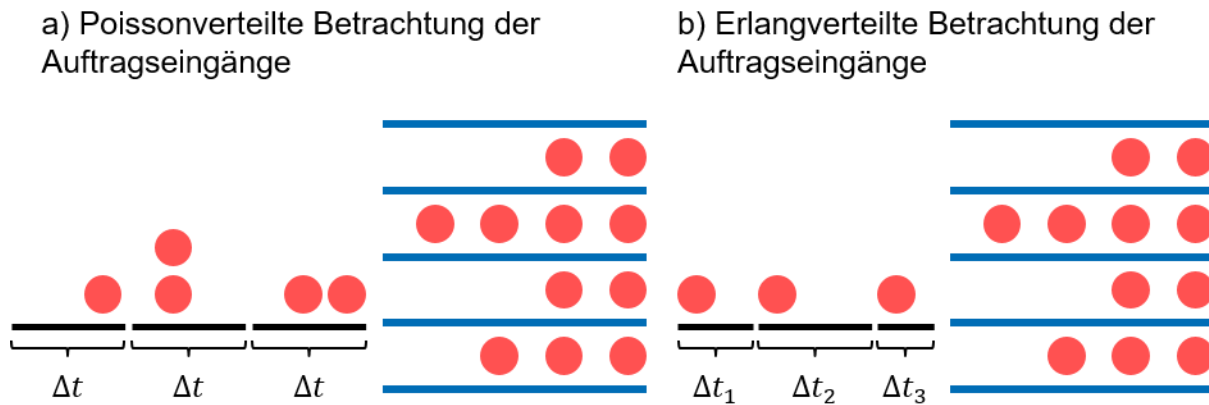


Abbildung 87: Stochastische Betrachtungsweisen zur Modellierung der Auftragseingänge a) Modellierung Poissonverteilt b) Modellierung Erlangverteilt. [Eigene Darstellung]

In der Simulationsumgebung kann die poissonverteilte Modellierung der Eingänge sinnvoll sein, wenn die Simulation einen Basistakt hat. Der Parameter λ entspricht dem Erwartungswert, der aus der Beobachtung abgeleitet werden kann. Im Falle der Nacharbeit betrifft dies die durchschnittliche Produktfehlerrate. Der Parameter k bezeichnet die Zufallsvariable. In der Simulationsumgebung ist dies die Wahrscheinlichkeit, dass genau k Produkte im Prüfstand als fehlerhaft erkannt werden und somit in das Wartesystem (Nacharbeitsbereich mit Bereitstellungsfläche) eingehen. In der Simulationsumgebung lässt sich mit der Poissonverteilung für einen bestimmten konstanten Zeitraum die Anzahl der Auftragseingänge in das Lager stochastisch generieren. Somit wächst zunächst der Bestand, wobei zur gleichen Zeit auch die Auftragsabgänge betrachtet werden müssen. Die Abgänge werden durch die Kostenfunktion der Metaheuristik bestimmt und stellen deren Lösung dar.

Nach der Auswahl des Ereignismodells wird das Lager instanziiert. Hierzu muss die Kapazität des Lagers definiert werden. Zudem erlaubt das Simulationsprogramm die Definition eines initialen Bestands. Die Aufträge im Lager haben sowohl einen Fehlercode, eine Variante als auch eine Position.

Im ersten Validierungsszenario werden 40 Mitarbeiter in Betracht gezogen, denen zum gleichen Zeitpunkt bspw. zu Schichtbeginn je ein Auftrag zugewiesen werden soll. Die Schichtdauer wird mit 8 h angegeben, jedoch in Sekunden umgerechnet. Die Vertrautheit mit einem Prozess wird in der Mitarbeiterdatenbank abgespeichert und ist pro Mitarbeiter und pro Prozess definiert.

Für die Ereignisrate wird ein Produktionsoutput von 10.000 Getrieben pro Tag mit einer Fehlerrate von 2 % angenommen. Damit ergibt sich unter Berücksichtigung des 2 Schichtbetriebs à 8 Stunden eine mittlere Fehlerrate von:

$$\lambda = \frac{10.000 * 0,02}{2 * 8} \approx 12,5/h$$

Das Simulationsprogramm beginnt mit dem Auslesen der Startwerte bspw. den verfügbaren Mitarbeitern sowie auch der Kenntnis über die Mitarbeiter. Das Simulationsprogramm errechnet zudem die verbleibende Zeit bis zum Abschluss des Auftrags pro

Mitarbeiter. Dies geschieht zum einen über eine lineare Schätzung. Diese geht von einer gleichbleibend zeitlichen Verteilung über alle Prozesse aus. Hierzu wird die Durchschnittszeit T_{mean} über die bisherigen auftragsbezogenen Prozesse ermittelt. Die Durchschnittszeit wird mit der Anzahl verbleibender Prozesse multipliziert, siehe Abbildung 79a. Diese Methode ist zwar leicht in der Umsetzung, allerdings ist ihre Aussagekraft umso höher, je mehr Prozesse bereits durchgeführt wurden. Es ist leicht nachvollziehbar, dass bei einem abgeschlossenen Prozess mit dieser Methode keine genaue Aussage über die Ausführungsdauer der nächsten Prozesse zu treffen ist, wenn gilt: $|P_{remain}| \gg |P_{Done}|$. P_{remain} bezeichnet hierbei die Menge abzuschließender Prozesse und $|P_{done}|$ die Anzahl abgeschlossener Prozesse. T_{mean} bezeichnet die (bisher) durchschnittliche Zeit für den Auftrag. Zur Schätzung der verbleibenden Zeit T_{est} wird die durchschnittliche Zeit T_{mean} mit der Anzahl verbleibender Prozesse multipliziert. Prinzipiell gilt, je weiter der Auftragsabschluss in der Zukunft liegt, desto ungenauer wird die Schätzung.

$$T_{mean} = \frac{T_{real}}{|P_{done}|} \quad (30)$$

$$T_{est} = |P_{remain}| * T_{mean} \quad (31)$$

Zum anderen kann anhand von historischen Werten die restliche Zeit T_{est} bis zum Auftragsabschluss geschätzt werden. Dabei wird zu jedem Prozess die durchschnittliche Zeit, die dieser Prozess benötigt $T_{p_{mean}}$ aufaddiert. Hierzu müssen die Zeiten jedoch immer erfasst werden, sodass eine Datengrundlage verfügbar ist, anhand derer die Zeiten ermittelt werden können. Abbildung 79 illustriert beide Herangehensweisen.

$$T_{est} = \sum_{p \in P_{remain}} T_{p_{mean}} \quad (32)$$

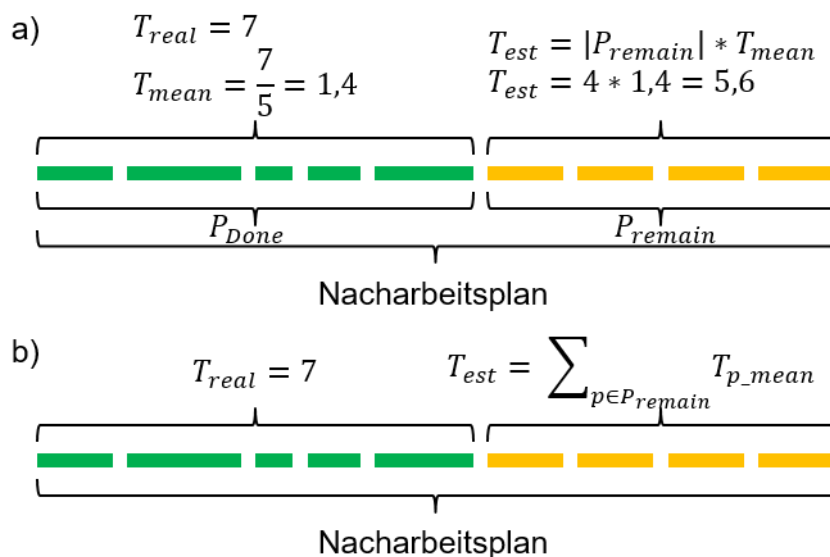


Abbildung 79: Schätzung der Restzeit bis zum Abschluss des Reparaturauftrags (a) Lineare Schätzung b) Schätzung basierend auf historischen Daten. [Eigene Darstellung]

Zur Ermittlung der Auftragszuweisung werden lediglich Mitarbeiter berücksichtigt, die eine gewisse Fortschrittsschwelle im Nacharbeitsauftrag überschritten haben. Für die Berechnung der besten Zuweisung werden auch diejenigen Mitarbeiter in Betracht gezogen, die schätzungsweise in nächster Zeit den Auftrag abschließen werden. Dabei werden den freien Mitarbeitern diejenigen Aufträge zugewiesen, die nach der Berechnung am ehesten den Prioritätsauftrag zugänglich machen. Der Kombinationsraum umfasst dabei lediglich die Lagerreihen, die auch tatsächlich die Prioritätsaufträge enthalten. Die restlichen Lagerreihen können außer Acht gelassen werden. Der komplette Raum muss allerdings ständig beobachtet werden, da neue Reparaturaufträge auf den Bereitstellungsflächen abgestellt werden.

Beträgt die verbleibende Bearbeitungszeit mindestens eines Mitarbeiters 0, so werden unter Anwendung der Metaheuristik sowie allen anderen als verfügbar betrachteten Mitarbeitern neue Aufträge zugewiesen. Ob ein Mitarbeiter bei der Auftragsvergabe berücksichtigt wird oder nicht, hängt davon ab, ob dieser einen Schwellenwert bereits unterschritten hat. Damit wird die vorzeitliche Betrachtung von bald abzuschließenden Mitarbeitern abgebildet. Schließlich kann das Warten auf einen Mitarbeiter lohenswert sein, wenn dieser erfahren ist und einen Auftrag entgegennehmen kann, der höhere Expertise erfordert. Bei der Berücksichtigung dieser Mitarbeiter muss die verbleibende Zeit bis zum Abschluss ihrer bisherigen Aufträge mitberücksichtigt werden. Werden verfügbaren Mitarbeitern Nacharbeitsaufträge zugewiesen, wird das Lagermodell aktualisiert. Zudem sind diese Mitarbeiter für die Dauer der geschätzten Bearbeitungszeit bis zum Unterschreiten eines Schwellwertes nicht verfügbar.

10.1.1 Modellierung der Auftragsklasse

Die Zugriffskombination gibt lediglich Auskunft über die Anzahl der zu entnehmenden Aufträge pro Reihe. Ist die Entnahmedisziplin bekannt, bspw. FIFO, können dadurch die Positionen der Aufträge ermittelt werden. Basierend darauf lässt sich der betroffene Auftrag ermitteln. Damit der Auftrag einem Mitarbeiter zugewiesen und von diesem bearbeitet werden kann, muss der Auftrag bestimmte Informationen tragen. Es ist bspw. ratsam, die Position des Auftrags zu kennen. Dies erlaubt es nachzuvollziehen, ob bei gegebener Zugriffskombination der Auftrag erfasst wurde. Im Folgenden wird der Nacharbeitsauftrag als Klasse beschrieben, wie in Abbildung 88 zu sehen ist. Diese Beschreibung erlaubt die Abbildung des Nacharbeitsauftrags in einer digitalen Umgebung, was später relevant sein wird.

Auftrag	
-	fehlercode : uint
-	produktID : uint
-	seriennummer : uint
-	position : uint[2]
<hr/>	
+	setPosition() : uint[2]
+	getPosition() : uint[2]

Abbildung 88: Aufbau der Klasse des Nacharbeitsauftrags

Die Auftragsklasse enthält neben dem Fehlercode (*fehlercode*), die Attribute *produktID* sowie *seriennummer*, die Auskunft über das Produkt geben. Ein weiteres Attribut ist *position*, die die Auftragssposition im Lager wiedergibt. Zudem enthält die Klasse zwei Methoden: *setPosition()* und *getPosition()*, die entweder die Position setzen oder die Position zurückgeben. Diese Werte bestehen aus einem Zahlenpaar, das Reihenummer und Zeilennummer wiedergibt. Das Setzen der Position wird notwendig, wenn ein Auftrag zuvor entnommen wurde und der betrachtete Auftrag nach dem FIFO-Prinzip weiter nach vorne wandert. Durch das Erfragen der Position des Auftrags auf der Bereitstellungsfläche wird sichergestellt, dass auch die Aufträge als Kandidaten betrachtet werden, die im Kombinationsraum liegen.

Eine weitere Klasse ist die Bereitstellungsfläche, deren Diagramm in Abbildung 89 zu sehen ist. Ihre Instanz beschreibt das Layout des Lagers samt Aufnahmekapazität der Reihe *kapazitaetproReihe* sowie die Zugriffsdisziplin Disziplin (FIFO, LIFO etc.).

Bereitstellungsflaeche	
-	zeilen : uint
-	reihen : uint
-	kapazitaetproReihe : uint
-	Disziplin : string

Abbildung 89: Aufbau der Klasse der Bereitstellungsfläche

Des Weiteren muss die Position eines jeden Auftrags auf der Bereitstellungsfläche bekannt sein. Dadurch sind bei der Entnahme von Aufträgen die Auftragseigenschaften bekannt. Bei der Entnahme aus der Bereitstellungsfläche nach dem FIFO-Prinzip rücken Aufträge Richtung Ausgang nach. Das Auftragsentnahmesystem der Bereitstellungsfläche wird analog zu einem System beschrieben, in dem Kugeln in einem Rohr nach unten fallen, wobei ein Rohr einzelne Reihen und die Kugeln die Aufträge darstellen. Werden Reihen mathematisch als Vektoren dargestellt, können Shiftmatrizen verwendet werden, um deren Verschiebung darzustellen. Die Verschiebung wird durch Matrizenmultiplikation erreicht. Da im Falle einer Reihe seitwärtige Verschiebungen nicht betrachtet werden, wird lediglich die Shiftmatrix zur Verschiebung nach unten betrachtet.

Die Shiftmatrix ist eine Matrix mit den Einträgen 1 oder 0. Eine Multiplikation mit der Shiftmatrix verursacht je nach Reihenfolge eine Verschiebung der Ursprungsmatrix „um eine Position entlang der Hauptdiagonalen“ [160]. Gleichung (43) zeigt den Aufbau einer exemplarischen Shiftmatrix.

$$Z = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (43)$$

Daher muss der (Reihen-)Vektor in der Form $\mathbb{R}^{n \times 1}$ vorliegen. Wird nun ein Auftrag aus einer Reihe entnommen, rücken die Einträge oberhalb des untersten Eintrags um eins nach unten. Zu beachten ist, dass die Shiftmatrix von links multipliziert wird, um eine Verschiebung der Matrizeneinträge nach unten zu erzielen. Das folgende Beispiel stellt eine Shiftoperation dar.

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (44)$$

Soll der Vektor um u Zeilen nach unten verschoben werden, so gilt:

$$Z^u \times V = V'$$

Wobei V den Reihenvektor darstellt und V' den neuen Reihenvektor, nach dem die Verschiebung stattgefunden hat. Da die Vektoren um unterschiedliche Beträge verschoben werden können, gelten dementsprechend verschiedene Shiftmatrizen für unterschiedliche Vektoren. Somit können unterschiedliche Reihenvektoren durch die Matrizenmultiplikation mit abgebildet werden. Ein Eintrag in V' wird nach Gleichung (45) berechnet:

$$v'_{ikl} = \sum_{j=1}^m Z_{ijl} * v_{jkl} \quad (45)$$

Diese Gleichung wird mit Hilfe von Abbildung 90 illustriert. Dabei stellt S die Shiftmatrix dar, wobei die Dimension l abhängig von der Anzahl von Reihenvektoren ist. Abhängig von der Verschiebung kann das Matrizenprodukt der Shiftmatrix mit sich selbst berechnet werden. Hierbei hängt u von l ab. Die Shiftmatrix wird nicht mit der Adjazenzmatrix multipliziert.

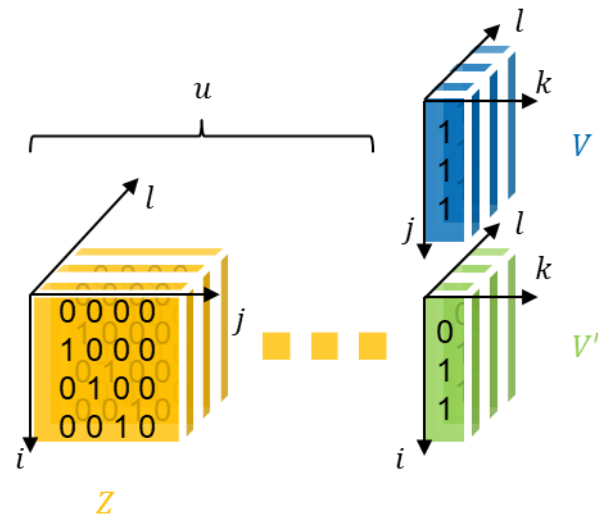


Abbildung 90: Matrizenmultiplikation zur Berechnung der statischer Reihenvektoren [Eigene Darstellung]

10.2 Ergebnisse mit Genetischem Algorithmus

Eine Methode zur Ermittlung optimierter Zuweisung ist die Verwendung des Genetischen Algorithmus. Der populationsbasierte Ansatz erlaubt im Gegensatz zum Simulated Annealing die zeitgleiche Betrachtung mehrerer Lösungen.

Abbildung 91 zeigt den Verlauf der Fitnesswerte im Genetischen Algorithmus sowie die der zufälligen Auftragszuweisung. Der Genetische Algorithmus liefert schneller bessere Werte, da er gezielter nach den Lösungen sucht.

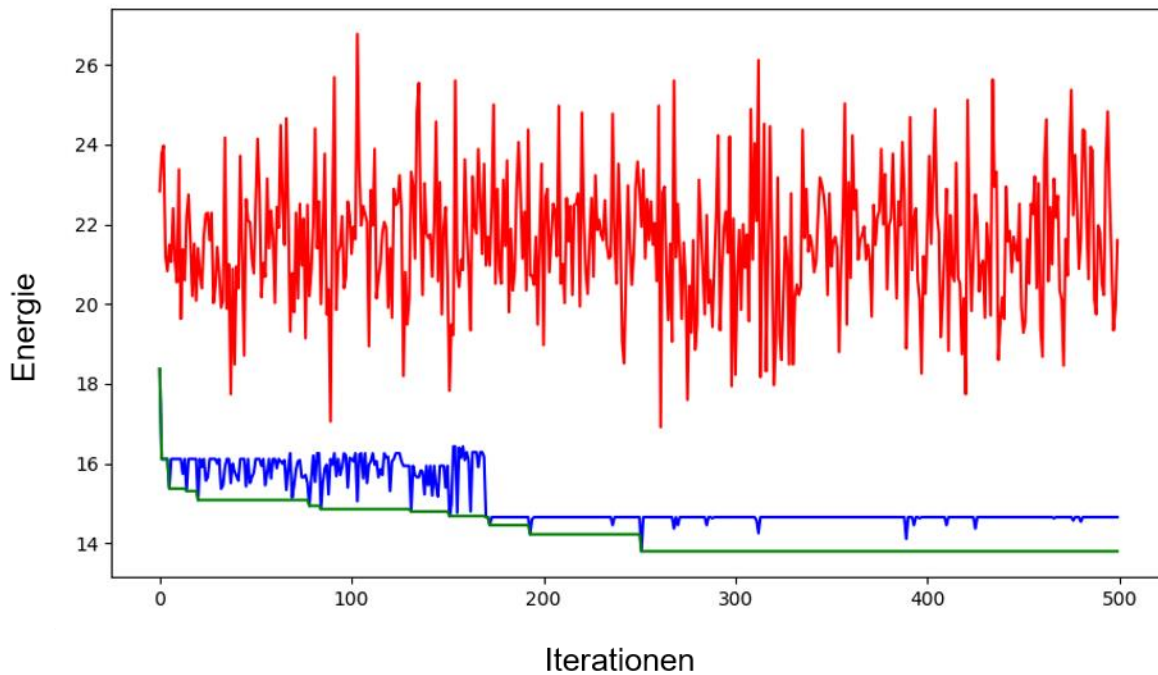


Abbildung 91: Vergleich zwischen Fitnessverläufen bei zufälliger Zuweisung (rot) und Zuweisung mittels Genetischem Algorithmus (blau). Die jeweils besten Werte sind mit dem grünen Verlauf abgebildet. [Eigene Darstellung]

10.3 Ergebnisse mit Simulated Annealing

Die Ergebnisse mit Simulated Annealing sind in Abbildung 92 abgebildet. Hierbei wird nur eine Zuweisung berücksichtigt und nicht eine Reihe von Zuweisungen über die gesamte Schicht. Der rot eingefärbte Verlauf beschreibt die Energie einer zufälligen Zuweisung. Eine zufällige Auftragszuweisung ist mit der willkürlichen Auftragsentnahme aus einer Bereitstellungsfläche vergleichbar. Der grün eingefärbte Verlauf stellt die Energie der Lösungen mittels Simulated Annealing dar. Es ist erkennbar, dass mittels Simulated Annealing gezielter bessere Lösungen zu finden sind. Die blaue Einfärbung stellt die jeweils von Simulated Annealing als bisher beste Lösung ermittelte Lösung dar.

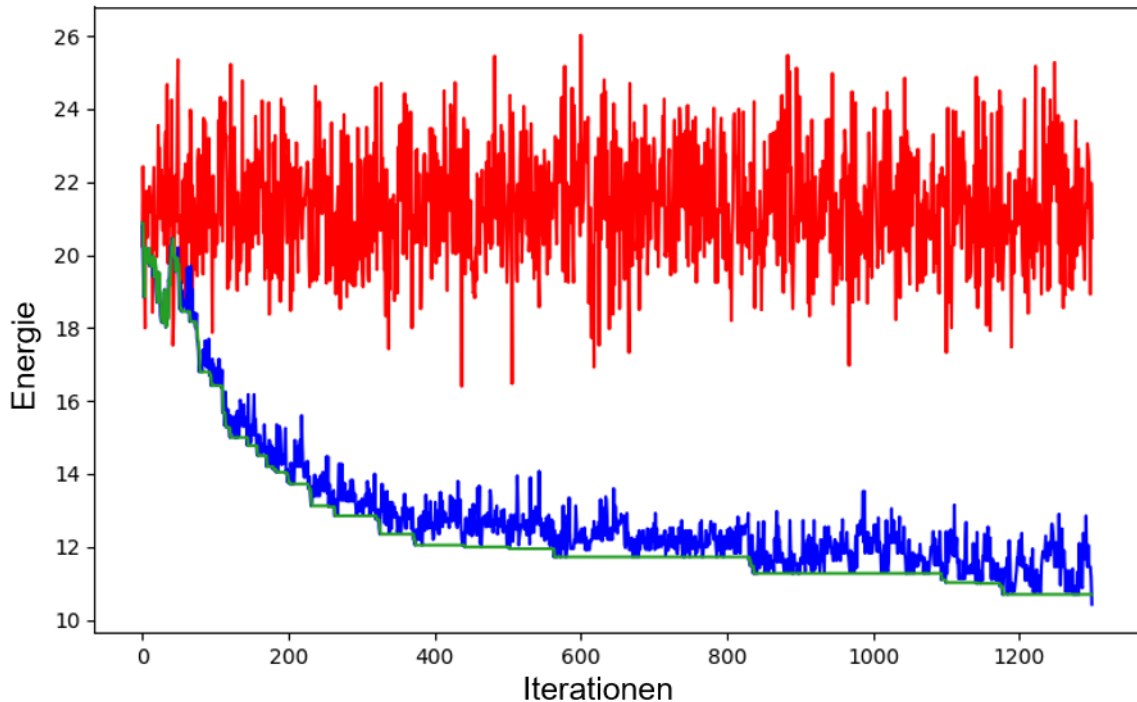


Abbildung 92: Lösung auf Basis von Simulated Annealing im Vergleich zur zufälligen Suche: Rot: Energie der Zufälligen Zuweisung, Blau: Energie der besten Lösung mittels Simulated Annealing, Grün: Energie der Nachbarlösungen mittels Simulated Annealing. [Eigene Darstellung]

Abbildung 93 zeigt den Verlauf des Lagerbestands über den gesamten Simulationszeitraum. Die Sprünge lassen sich durch die stündliche Ankunft neuer Aufträge erklären. Das Sinken der Bestände wird allerdings im simulativen Minutentakt berücksichtigt. Die rote Kurve zeigt den Lagerbestand bei zufälliger Auftragszuweisung. Dabei steigt der Lagerbestand tendenziell an. Die blaue Kurve zeigt den Bestandsverlauf bei Zuteilung der Aufträge nach Simulated Annealing. Diese Kurve hat tendenziell einen stetigen bzw. fallenden Verlauf. Es ist aber zu berücksichtigen, dass die Kurve nicht zwangsläufig fallen muss. Ob der Bestand abgebaut wird, ist zudem von der Vertrautheit der Mitarbeiter mit den Prozessen bzw. der Größe der Belegschaft abhängig. Der Vergleich zeigt aber, dass mit Simulated Annealing bessere Ergebnisse hinsichtlich des Bestandsabbaus ausgegeben werden. Hintergrund ist die gezielte Suche besserer Lösungen durch Simulated Annealing und ein Summeneffekt bei durchlaufender Simulation.

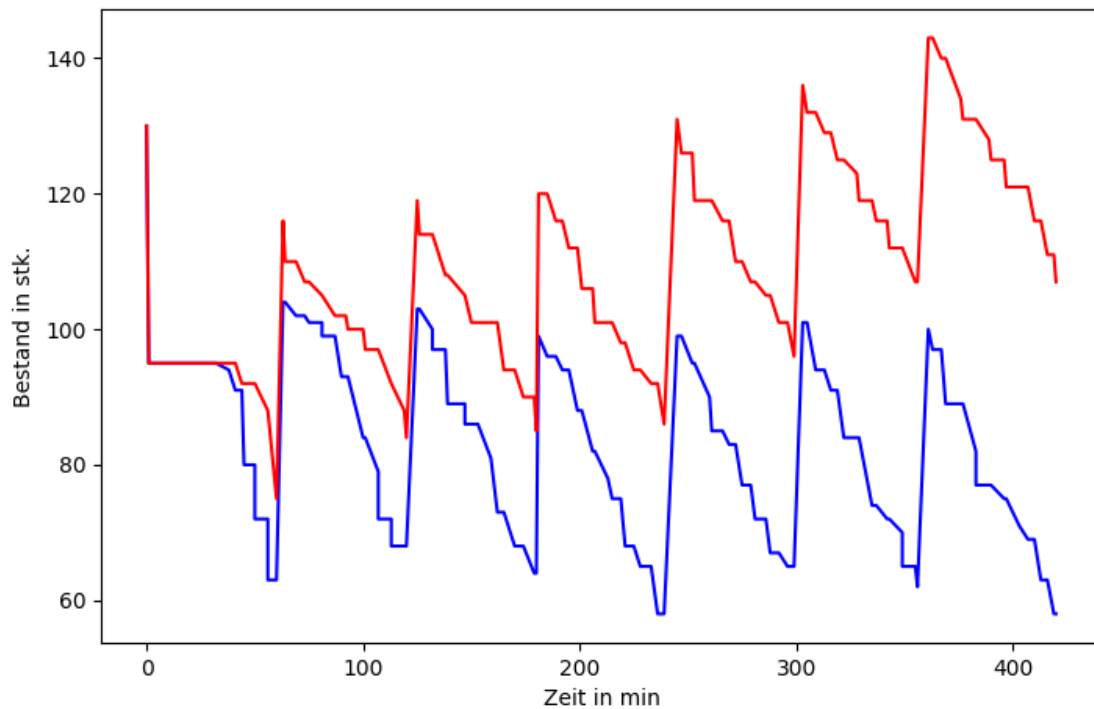


Abbildung 93: Vergleich der Lagerbestände bei zufälliger Zuweisung und Zuweisung basierend auf Simulated Annealing. Rot: Bestandsverlauf bei zufälliger Zuweisung; Blau: Bestandsverlauf bei Zuweisung mit Simulated Annealing. [Eigene Darstellung]

Die Simulationen wurden mehrfach durchgeführt und zeigen immer bessere Ergebnisse als die zufällige Zuweisung der Aufträge.

Ein zweiter Fall der Betrachtung ist die bevorzugte Durchschleusung bestimmter Varianten. Hierzu wird vorausgesetzt, dass die Belegung des Lagers samt Platzierung der Varianten bekannt ist. Dies ist z. B. möglich, indem zum Zeitpunkt der Einlagerung die Produktvariante bspw. durch einen Scandvorgang angegeben wird. Zudem ist die Nachverfolgung des Lagerbestandes samt Platzierung im Lager erforderlich. Die Steuerung ermittelt anhand der Anzahl entnommener Aufträge aus dem Lager sowie der Kenntnis des Platzes die aktualisierte Lagerbelegung. Zudem werden die Aufträge bevorzugt zugewiesen, die am ehesten eine Durchschleusung oder Zugänglichkeit der priorisierten Variante versprechen. Einerseits geschieht dies unter Betrachtung der vorhandenen priorisierten Aufträge pro Lagerreihe, andererseits aber auch durch Betrachtung der vermuteten Bearbeitungszeit vorausgehender fremder Varianten. Abbildung 94 zeigt den Verlauf der Energiefunktion im Falle der bevorzugten Durchschleusung bestimmter Varianten (Priorisierter Fall). Es zeigt sich erneut, dass Simulated Annealing auch in diesem Fall gezielter bessere Lösungen findet.

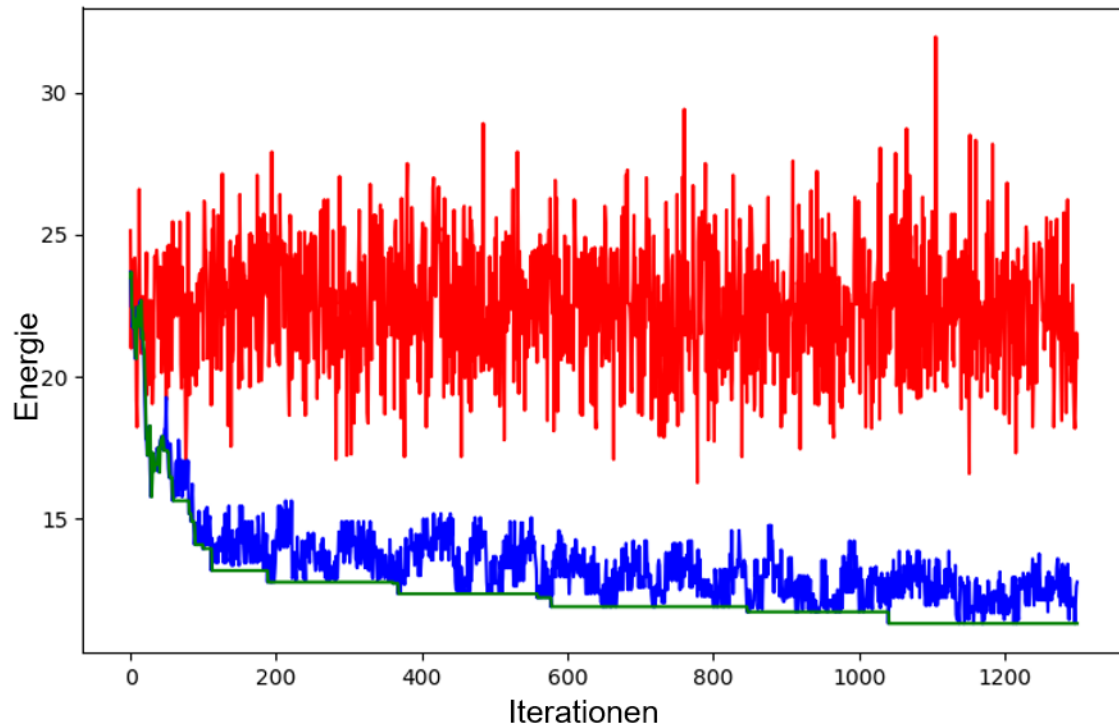


Abbildung 94: Lösung auf Basis von Simulated Annealing bei priorisierter Durchschleusung im Vergleich zur zufälligen Suche: Rot: Energie der zufälligen Zuweisung, Blau: Energie der besten Lösung mittels Simulated Annealing, Grün: Energie der Nachbarlösungen mittels Simulated Annealing. [Eigene Darstellung]

Abbildung 95 zeigt den Verlauf des Lagerbestands bei der bevorzugten Durchschleusung von einer bestimmten Produktvariante. Auch im priorisierten Fall werden gegenüber der zufälligen Zuweisung Bestände reduziert.

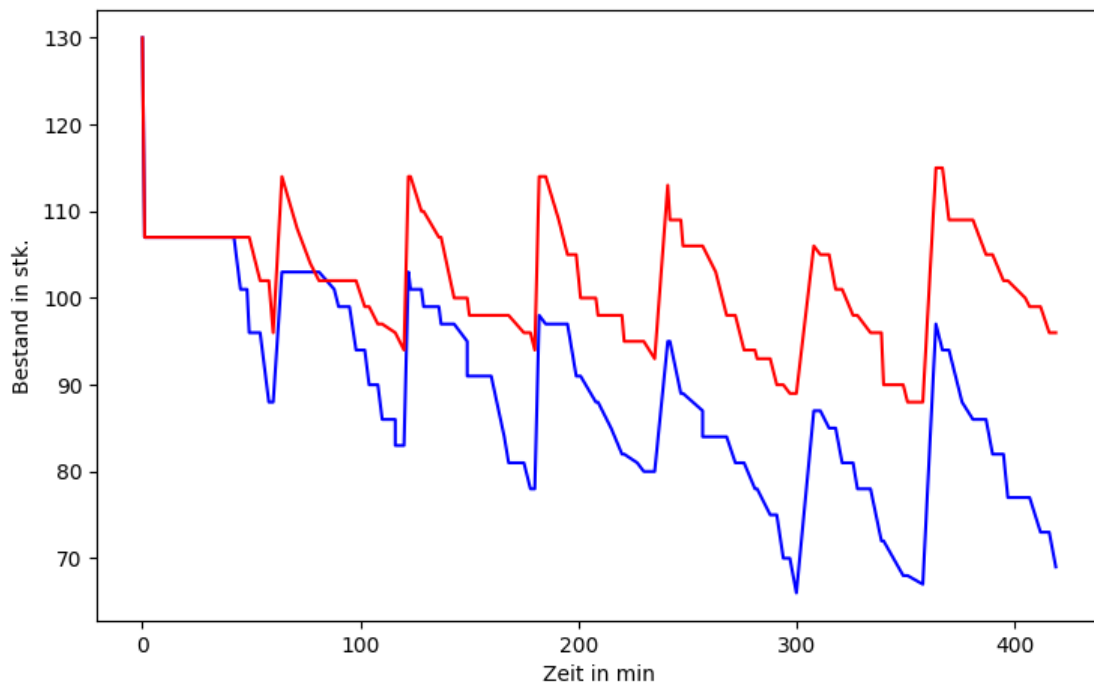


Abbildung 95: Vergleich der Lagerbestände im priorisierten Fall bei zufälliger Zuweisung und Zuweisung basierend auf Simulated Annealing. Rot: Bestandsverlauf bei zufälliger Zuweisung; Blau: Bestandsverlauf bei Zuweisung mit Simulated Annealing. [Eigene Darstellung]

Im Vergleich zur nicht-priorisierten Durchschleusung fällt die Bestandsreduzierung schwächer aus. Dies ist mit der bevorzugten Durchschleusung der Produktvariante zu Lasten der optimalen Mitarbeiterzuweisung zu begründen.

Abbildung 96 zeigt den Bestandsverlauf der priorisierten Produktvariante im Lager. Der rot eingefärbte Verlauf stellt den Bestand bei zufälliger Zuweisung dar. Der blau eingefärbte Verlauf bildet den Bestand bei ermittelter Zuweisung ab. Es ist erkennbar, dass der Bestand an priorisierten Produktvarianten bei der Zuweisung mittels Simulated Annealing im Vergleich zur zufälligen Zuweisung geringer ist.

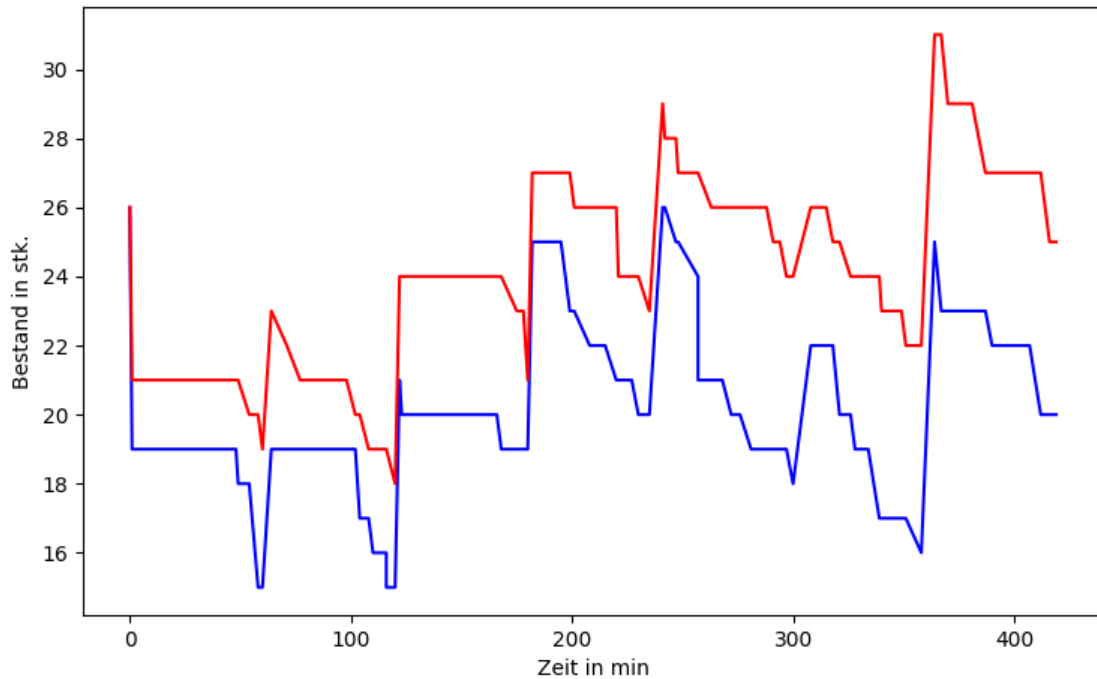


Abbildung 96: Vergleich der Lagerbestände der bevorzugten Produktvariante im priorisierten Fall bei zufälliger Zuweisung (rot) und Zuweisung basierend auf Simulated Annealing (blau). ¶ [Eigene Darstellung]

10.4 Ergebnisse der Variantendurchschleusung

Bei der Durchschleusung der Produktvarianten ist in der Simulationsumgebung die Kenntnis über Variantenposition im Lager erforderlich. Damit wird der Raum für die Auswahl der Aufträge eingegrenzt. Abbildung 97 zeigt das Ergebnis zweier Simulationen, in denen eine spezielle Variante in den Fokus rückt. Es ist ersichtlich, dass die Priorisierungsmethode im Vergleich zur Standardmethode mehr priorisierte Varianten durchschleust. Die Standardfallmethode lässt aber in der Summe mehr Aufträge abarbeiten. Allerdings ist das Ergebnis immer von der Reduzierungsfähigkeit und somit der Lage der Priorisierungsvarianten abhängig.

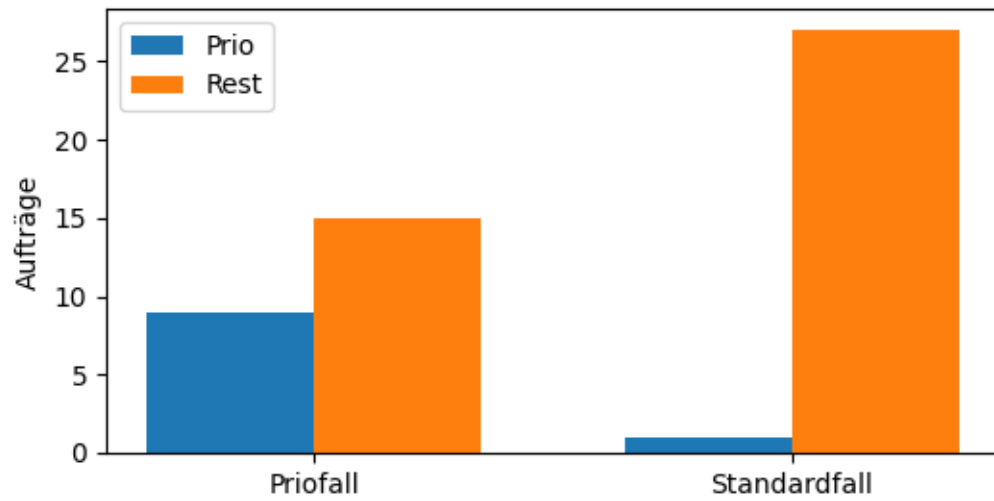


Abbildung 97: Beispielhafter Vergleich zwischen Anzahl der priorisierten Varianten und den restlichen Varianten im Priorisierungsfall und im Standardfall. [Eigene Darstellung]

11 Zusammenfassung und Ausblick

Im letzten Kapitel dieser Arbeit werden die Ergebnisse der Arbeit hinsichtlich der Zielfragen zusammengefasst und diskutiert. Zudem wird ein Ausblick auf aufbauende Arbeiten gegeben.

11.1 Zusammenfassung

In dieser Arbeit wurde eine Blaupause für ein Kognitives Assistenzsystem für den Einsatz in dynamischen Produktionsbereichen entwickelt und umgesetzt. Die dynamische Umgebung wurde anhand des Nacharbeitsbereichs in einem Maschinenbauunternehmen abgebildet. Das Assistenzsystem soll neben der Assistenzfunktion auch für globale Funktionen wie Optimierungen in angrenzenden Bereichen nutzbar sein. Hierzu wird die optimierte Auftragszuweisung in der Nacharbeit als Beispiel genommen. Dabei werden Mitarbeitern Aufträge optimiert zugewiesen, sodass Auftragsbestände schneller reduziert werden können.

Im Folgenden wird die Umsetzung der Ziele dieser Arbeit, die in Kapitel 2.2 formuliert wurden, diskutiert.

Wie kann ein Kognitives Assistenzsystem für den Einsatz in dynamischen Produktionsbereichen befähigt werden?

Im Rahmen dieser Arbeit wurde für das Kognitive Assistenzsystem ein 4-stufiges Entwicklungsmodell gewählt, in dem die nachfolgende Stufe die Eignung des Assistenzsystems für dynamische Produktionsbereiche verbessert bzw. erweitert.

In der ersten Stufe verfügt das Kognitive Assistenzsystem über einen einfachen Digitalen Nacharbeitsplan, der einen mit Sensoren und Aktoren behafteten Arbeitsplatz ansteuert. Der Nacharbeitsplan enthält zu jedem Prozessschritt vier Arten von Daten: Prozessinformation, Prozessausführung, Prozessprüfung und Prozessdokumentation.

In der zweiten Stufe verfügt das Kognitive Assistenzsystem über eine graphenbasierte Informationsstruktur, die die Handhabung der Nacharbeitspläne vereinfacht. Grundlage der Informationsstruktur bildet der Produktzustandsgraph. Dieser Graph wird mit Prozessdaten angereicht. Zudem werden ausgewählte Produktzustände markiert. Diese dienen der Kennzeichnung eines Produktzustands, der mit einem Fehler assoziiert ist, den Fehlercodes.

In der dritten Stufe erlaubt das Kognitive Assistenzsystem dem Mitarbeiter, selbstständig eine neue Fehlerursache einzugeben, sodass aus irgendeinem Zustand heraus ein anderer Produktzustand erreicht werden kann. In diesem Produktzustand kann ebenfalls die Fehlerbehebung stattfinden, wobei erneut die Remontageprozesse angezeigt werden. Der Mitarbeiter kann mithilfe eines Katalogs den zum Produktzustand zugehörigen Fehlercode in Erfahrung bringen.

In der vierten Stufe schätzt das Assistenzsystem den Produktzustand ein und kann auch ohne die Eingabe des Fehlercodes Abweichungen erkennen, den Produktzustand ermitteln und geeignete Maßnahmen durchführen. Damit werden der reale Produktzustand sowie der digitale Produktzustand synchron gehalten. Grundlage für diese Funktionalität bildet ein vorangegangenes Lernverfahren, das über einen bestimmten Zeitraum im Hintergrund läuft und immer wieder interne Vorhersagen trifft, bis eine ausreichende Genauigkeit erzielt wurde.

Bereits mit der fehlerspezifischen Generierung des Nacharbeitsplans auf Basis des angereicherten Graphen war das Kognitive Assistenzsystem für dynamische Produktionsbereiche nutzbar. In den weiteren Stufen wurden die zwei Handlungsfreiheiten des Mitarbeiters berücksichtigt:

1. Freiheit der Fehlernachverfolgung
2. Freiheit der Produkthandhabung

Die erste Freiheit ist gewollt und ermöglicht dem Mitarbeiter, abweichende Fehlerursachen zu verfolgen und zu beheben. Die zweite Freiheit ist ungewollt, aber der Organisation des Nacharbeitsbereichs geschuldet. Der Mitarbeiter hat hierbei grundsätzlich die Möglichkeit, absichtlich oder unabsichtlich das Produkt in einer nicht präferierten Art und Weise zu demontieren. Die Stufen 3 und 4 des Assistenzsystems zielen auf diese Freiheiten ab.

Wie kann das Kognitive Assistenzsystem auf die Bedürfnisse des Mitarbeiters in dynamischen Produktionsbereichen angepasst werden?

Bei der Entwicklung und Gestaltung des Assistenzsystems wurden Mitarbeiterbedürfnisse in Form von vertrautheitsbasierter Dosierung von Informationen berücksichtigt. Dazu wurde die Vertrautheit der Mitarbeiter mit einem Produkt beachtet. Es wurden je nach Prozessvertrautheit ausgewählte Informationen nicht eingeblendet. Ein Beispiel hierfür war die prozess- und mitarbeiterspezifische Abschaltung des Laserprojektors bei der Anzeige der Verschraubereihenfolge. Hierdurch soll die Bevormundung erfahrener Mitarbeiter vermieden werden. Zur Feststellung der Produktvertrautheit wurden zwei Arten der Berücksichtigung vorgestellt:

1. Kategorisierte Produktvertrautheit
2. Individuelle Produktvertrautheit

Im ersten Fall werden Mitarbeiter basierend auf der Einschätzung der Vertrautheit mit dem Produkt von einem Vorarbeiter oder Meister einer Kategorie zugeordnet. Allen Mitarbeitern derselben Kategorie werden die gleichen prozessspezifischen Informationen angezeigt. Im Rahmen dieser Entwicklung gab es die Kategorien: „Anfänger“, „Fortgeschrittener“ und „Experte“.

Im zweiten Fall wurden individuelle Profile angelegt, die sich aus der Beobachtung der Prozesse ergaben. Dabei wurde die Ausführungsdauer, die Berechtigung sowie die letztmalige Prozessausführung berücksichtigt.

Aber nicht allein die Auswahl der anzuzeigenden Informationen, auch die Möglichkeiten der Überstimmung des Assistenzsystems sowie die Nachverfolgung eigener Fehlerursachenvermutung kommen den Bedürfnissen nach möglichst geringer Beschneidung der Handlungsfreiheiten nach.

Wie kann ein Kognitives Assistenzsystem über die herkömmliche Mitarbeiterassistenz hinaus die Produktivität steigern?

Da das Kognitive Assistenzsystem permanent Prozessdaten auswertet, stellt sich die Frage, ob diese auch anderweitig zur Steigerung der Produktivität genutzt werden können. In dieser Arbeit wurde gezeigt, dass neben dem eigentlichen Nacharbeitsauftrag außerdem die Zuweisung der Aufträge auf Basis von Nutzerdaten und Lagermodellen optimiert werden kann. Hierzu wurde ein Lagermodell entwickelt sowie die Zuweisungsstrategien in Form von Metaheuristiken betrachtet. Dabei wurden sowohl der Genetische Algorithmus als auch Simulated Annealing miteinander verglichen. In einer Simulationsumgebung, in der sowohl die Bereitstellungsfläche der funktionsunfähigen Getriebe als auch die Nacharbeitsprozesse simuliert wurden, konnten Optimierungseffekte hinsichtlich geringerer Bearbeitungszeiten und Bestandsreduzierung gezeigt werden.

11.2 Ausblick

Herkömmliche Assistenzsysteme werden weitestgehend im Verhältnis 1 zu 1 verwendet. Ein Assistenzsystem unterstützt genau einen Mitarbeiter. Zudem prüfen Assistenzsysteme häufig nicht eine Vielzahl alternativer Prozesse ab. Daher werden sie vorzugsweise in der Serienmontage eingesetzt. Diese Verwendung des Assistenzsystems schließt viele Branchen und Montageorganisationen aus. In der Gruppenmontage oder Baustellenmontage haben Mitarbeiter sehr viel mehr Möglichkeiten ein Produkt zusammenzusetzen. Daher sind Assistenzsysteme, die mehrere Mitarbeiter gleichzeitig erfassen, besonders geeignet für diese Organisationsformen. Zudem könnten sie sehr kostengünstig sein, da ein 1-zu-1-Invest nicht notwendig wäre.

12 Literaturverzeichnis

- [1] H. Becker, *Phänomen Toyota: Erfolgsfaktor Ethik ; mit 15 Tabellen*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006. [Online]. Available: <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10144293>
- [2] T. Bauernhansl, *Management in der Produktion*, 2020.
- [3] F. Rösler, *Psychophysiologie der Kognition: Eine Einführung in die Kognitive Neurowissenschaft*. Heidelberg: Spektrum Akademischer Verlag, 2011.
- [4] Fraunhofer Austria Research Gesellschaft mit beschränkter Haftung, *Kognitive und digitale Assistenzsysteme*. [Online]. Available: <https://www.fraunhofer.at/de/zusammenarbeit/advanced-industrial-management/kognitive-und-digitale-assistenzsysteme.html>
- [5] Dudenredaktion, *Lernen*. [Online]. Available: <https://www.duden.de/node/151013/revision/609221> (accessed: Jan. 5 2022).
- [6] Oxford Learner's Dictionaries, *learn*. [Online]. Available: <https://www.oxfordlearnersdictionaries.com/definition/english/learn?q=learn> (accessed: Jan. 5 2022).
- [7] J. Winter, *Glossar*. [Online]. Available: <https://www.plattform-lernende-systeme.de/glossar.html>
- [8] Dudenredaktion, *Montage*. [Online]. Available: <https://www.duden.de/node/98870/revision/98906> (accessed: Sep. 13 2021).
- [9] T. Tolio, *Dictionary of Production Engineering IV - Assembly Wörterbuch der Fertigungstechnik IV - Montage Dizionario di Ingegneria della Produzione IV - Assemblaggio*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020.
- [10] B. Lotter and H.-P. Wiendahl, *Montage in der industriellen Produktion: Ein Handbuch für die Praxis*. Berlin, New York: Springer, 2006.
- [11] D. Whitney, *Mechanical assembly and product development*. Oxford: Oxford University Press, 2003.
- [12] J. Bauer, *Produktionslogistik/Produktionssteuerung kompakt*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014.
- [13] E. Westkämper, H.-J. Warnecke, M. Dinkelmann, and H. Haag, *Einführung in die Fertigungstechnik*, 8th ed. Wiesbaden: Vieweg + Teubner, 2010.
- [14] G. Pawellek, *Ganzheitliche Fabrikplanung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.
- [15] T. Bauernhansl, *Fabrikbetriebslehre 1: Management in der Produktion*, 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg; Imprint: Springer Vieweg, 2020.
- [16] U. Bracht, D. Geckler, and S. Wenzel, *Digitale Fabrik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [17] G. Schuh and V. Stich, *Produktionsplanung und -steuerung 1*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [18] D. Habenicht, *Verkettungsarten im Wertstrom schlanker Unternehmen: Analysen, Algorithmen und Auswirkungen auf Leistung, Durchlaufzeiten und Bestände*. Berlin: Springer Vieweg, 2017. [Online]. Available: <http://www.springer.com/>

- [19] M. Bornewasser and S. Hinrichsen, *Informativische Assistenzsysteme in der variantenreichen Montage*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020.
- [20] J. Feldhusen and K.-H. Grote, Eds., *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*, 8th ed. Berlin, Heidelberg: Springer Vieweg, 2013. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-29569-0>
- [21] U. Weidemann, *Montagecluster zur Strukturierung der Fahrzeugendmontage*. Dissertation, 2017.
- [22] M. Wiesbeck, *Struktur zur Repräsentation von Montagesequenzen für die situationsorientierte Werkerführung*. München: Utz, 2014.
- [23] M. Sasiadek, "Planning and analysis of mechanical assembly sequences in design engineering – Part I: The Method," *Teh. vjesn.*, vol. 22, no. 2, pp. 337–342, 2015, doi: 10.17559/TV-20130428110530.
- [24] L. S. Homem de Mello and A. C. Sanderson, "Automatic Generation of Mechanical Assembly Sequences," Fort Belvoir, VA, 1988.
- [25] A. Lambert, "GENERATION OF ASSEMBLY GRAPHS BY SYSTEMATIC ANALYSIS OF ASSEMBLY STRUCTURES," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 85–90, 2002, doi: 10.3182/20020721-6-ES-1901.00017.
- [26] S. Khot, G. Kindler, E. Mossel, and R. O' Donnell, "Optimal Inapproximability Results for MAX - CUT and Other 2 - Variable CSPs?," *SIAM J. Comput.*, vol. 37, no. 1, pp. 319–357, 2007, doi: 10.1137/S0097539705447372.
- [27] M. Kljajin and M. Opalić, "Product disassembly sequences approach in the early stages of product design," *INTERNATIONAL DESIGN CONFERENCE - DESIGN 2004*, 2004.
- [28] L. S. Homem de Mello and S. Lee, *Computer-Aided Mechanical Assembly Planning*. Boston, MA: Springer US, 1991.
- [29] E. V. Konstantinova and V. A. Skorobogatov, "Application of hypergraph theory in chemistry," *Discrete Mathematics*, vol. 235, 1-3, pp. 365–383, 2001, doi: 10.1016/S0012-365X(00)00290-9.
- [30] L. Laperrière and H. A. ElMaraghy, "GAPP: A generative assembly process planner," *Journal of Manufacturing Systems*, vol. 15, no. 4, pp. 282–293, 1996, doi: 10.1016/0278-6125(96)84553-5.
- [31] T. Jeandin and C. Mascle, "A New Model to Select Fasteners in Design for Disassembly," *Procedia CIRP*, vol. 40, pp. 425–430, 2016, doi: 10.1016/j.procir.2016.01.084.
- [32] J. D. Chiodo and W. L. Ijomah, "Use of active disassembly technology to improve remanufacturing productivity: automotive application," *International Journal of Computer Integrated Manufacturing*, vol. 27, no. 4, pp. 361–371, 2014, doi: 10.1080/0951192X.2012.667151.
- [33] J. Chiodo and N. Jones, "Smart materials use in active disassembly," *Assembly Automation*, vol. 32, no. 1, pp. 8–24, 2012, doi: 10.1108/01445151211198683.

- [34] W. Mulder, J. Blok, S. Hoekstra, and F. Kokkeler, *Design for maintenance: Guidelines to enhance maintainability, reliability and supportability of industrial products* / Wienik Mulder, Jeroen Blok, Sipke Hoekstra, Frans Kokkeler. Enschede: University of Twente, 2012.
- [35] World Business Council for Sustainable Development, *Design for disassembly/deconstruction*. [Online]. Available: <https://www.ceguide.org/Strategies-and-examples/Design/Design-for-disassembly-deconstruction>
- [36] A. Diener, *Afterlife: An Essential Guide To Design For Disassembly*. [Online]. Available: <https://www.core77.com/posts/15799/afterlife-an-essential-guide-to-design-for-disassembly-by-alex-diener-15799>
- [37] J. Gottmann, *Produktionscontrolling*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016.
- [38] *Qualitätsmanagementsysteme – Grundlagen und Begriffe (ISO 9000:2015): Deutsche und Englische Fassung EN ISO 9000:2015*, DIN Deutsches Institut für Normung e. V., Berlin.
- [39] M. Kropik, *Produktionsleitsysteme in der Automobilfertigung*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2009.
- [40] S. Tackenberg, S. Duckwitz, C. F. Behlau, and C. M. Schlick, “Instrumentarium zur personenindividuellen Bewertung und Verbesserung der individuellen Arbeitsorganisation,” in *VDI-Buch, Produktivitätsmanagement von Dienstleistungen*, C. M. Schlick, M. Schenk, D. Spath, and W. Ganz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 221–276.
- [41] Centre for Remanufacturing & Reuse (CRR), *What is Remanufacturing?* [Online]. Available: <https://www.remanufacturing.org.uk/what-is-remanufacturing> (accessed: Jul. 15 2020).
- [42] U. Lange, “Kurzanalyse Nr. 18: Ressourceneffizienz durch Remanufacturing - Industrielle Aufarbeitung von Altteilen,” VDI Zentrum Ressourceneffizienz GmbH, Berlin, 2017.
- [43] I. M. Langella, *Planning demand-driven disassembly for remanufacturing*, 1st ed. Wiesbaden: Deutscher Universitäts-Verlag, 2007.
- [44] D. Parker, K. Riley, S. Robinson, H. Symington, *Remanufacturing Market Study: For Horizon 2020, grant agreement No 645984, November 2015*. [Online]. Available: <https://www.remanufacturing.eu/assets/pdfs/remanufacturing-market-study.pdf> (accessed: 2023).
- [45] APRA, *APRA’s Remanufacturing Translation Matrix*. [Online]. Available: http://www.apra-europe.org/dateien/downloads/Reman_Terms_and_Definitions_Translations_Final.pdf
- [46] M. Thierry, M. Salomon, J. van Nunen, and L. van Wassenhove, “Strategic Issues in Product Recovery Management,” *California Management Review*, vol. 37, no. 2, pp. 114–136, 1995, doi: 10.2307/41165792.
- [47] B. M. Beamon and C. Fernandes, “Supply-chain network configuration for product recovery,” *Production Planning & Control*, vol. 15, no. 3, pp. 270–281, 2004, doi: 10.1080/09537280410001697701.

- [48] F. J. García-Rodríguez, C. Castilla-Gutiérrez, and C. Bustos-Flores, "Implementation of reverse logistics as a sustainable tool for raw material purchasing in developing countries: The case of Venezuela," *International Journal of Production Economics*, vol. 141, no. 2, pp. 582–592, 2013, doi: 10.1016/j.ijpe.2012.09.015.
- [49] I. Hartwell and J. Marco, "Management of intellectual property uncertainty in a remanufacturing strategy for automotive energy storage systems," *Jnl Remanufacturing*, vol. 6, no. 1, 2016, doi: 10.1186/s13243-016-0025-z.
- [50] European Commission. Directorate-General for the Environment., *Promoting re-manufacturing, refurbishment, repair, and direct reuse: As a contribution to the G7 Alliance on Resource Efficiency 7-8 February 2017 Brussels, Belgium*. Brussels: Luxembourg : Publications Office, 2017.
- [51] United Nations, UNEP, and Open-ended Working Group of the Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and Their Disposal, Eds., *Revised glossary of terms: Matters related to the work programme of the Open-ended Working Group for 2016-2017: legal, governance and enforcement matters: providing further legal clarity UNEP/CHW/OEWG.10/INF/10*, 2016.
- [52] S. Parsa and M. Saadat, "Intelligent selective disassembly planning based on disassemblability characteristics of product components," *Int J Adv Manuf Technol*, vol. 104, 5-8, pp. 1769–1783, 2019, doi: 10.1007/s00170-019-03857-1.
- [53] R. T. Lund, *Remanufacturing: The experience of the United States and implications for developing countries*, 1st ed. Washington, DC: The World Bank, 1984.
- [54] H. Jodlbauer, *Produktionsoptimierung: Wertschaffende sowie kundenorientierte Planung und Steuerung*, 2nd ed. Wien: Springer, 2008.
- [55] W. Kilger, J. R. Pampel, and K. Vikas, *Flexible Plankostenrechnung und Deckungsbeitragsrechnung*. Wiesbaden: Gabler Verlag, 2012.
- [56] Arkite NV, *Operator Guidance: Guiding operators with projected augmented reality work instructions*. (accessed: 2023).
- [57] Bosch Rexroth AG, *ActiveAssist- im Kern eine Software, im Ergebnis eine All-in-One-Lösung*. [Online]. Available: https://www.boschrexroth.com/ics/cat/content/assets/Online/do/MT_ActiveAssist_Overview__2019-07__20200430_115651.pdf (accessed: 2023).
- [58] ulixes GmbH, *Der Assistent: The worker assistance system for every solution*. [Online]. Available: <https://www.ulixes.com/der-assistent/> (accessed: 2023).
- [59] Optimum datamanagement solutions GmbH, *Der Schlaue Klaus: Werkerassistenzsysteme für die industrielle Fertigung*. [Online]. Available: <https://www.optimum-gmbh.de/produkte/der-schlaue-klaus> (accessed: 2023).
- [60] M. Bovenschulte, *Kognitive Assistenzsysteme: Technikkurzprofil Nr. 38*, 2020.
- [61] R. Weidner, T. Redlich, and J. P. Wulfsberg, *Technische Unterstützungssysteme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
- [62] B. Ludwig, *Planbasierte Mensch-Maschine-Interaktion in multimodalen Assistenzsystemen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

- [63] H. Wandke *, "Assistance in human-machine interaction: a conceptual framework and a proposal for a taxonomy," *Theoretical Issues in Ergonomics Science*, vol. 6, no. 2, pp. 129–155, 2005, doi: 10.1080/1463922042000295669.
- [64] H. W. Knut Polkehn, "Das Web als Assistenzsystem?!", in *Hassenzahl, M. & Peissner, M. (Hrsg.), Tagungsband UP05*, pp. 68–75.
- [65] W. Apt, M. Bovenschulte, K. Priesack, C. Weiß, and E. A. Hartmann, "Einsatz von digitalen Assistenzsystemen im Betrieb," *Forschungsbericht*, no. 502, Januar, 2018.
- [66] W. Apt, M. Schubert and S. Wischmann, *Digitale Assistenzsysteme: Perspektiven und Herausforderungen für den Einsatz in Industrie und Dienstleistungen*. Berlin: VDI/VDE Innovation + Technik GmbH, 2018.
- [67] S. Büttner, O. Sand, and C. Röcker, "Exploring Design Opportunities for Intelligent Worker Assistance: A New Approach Using Projection-Based AR and a Novel Hand-Tracking Algorithm," in *Lecture Notes in Computer Science, Ambient Intelligence*, A. Braun, R. Wichert, and A. Maña, Eds., Cham: Springer International Publishing, 2017, pp. 33–45.
- [68] A. Bannat, "Ein Assistenzsystem zur digitalen Werker-Unterstützung in der industriellen Produktion," München, Technische Universität München, Diss., 2014, Universitätsbibliothek der TU München, München, 2014.
- [69] D. Gorecky, S. F. Worgan, and G. Meixner, "COGNITO," in *ECCE 2011: European Conference on Cognitive Ergonomics 2011 : 24-26 August 2011, Rostock, Germany : The 29th annual conference of the European Association of Cognitive Ergonomics*, Rostock, Germany, 2011?, p. 53.
- [70] E. Lampen, J. Teuber, F. Gaisbauer, T. Bär, T. Pfeiffer, and S. Wachsmuth, "Combining Simulation and Augmented Reality Methods for Enhanced Worker Assistance in Manual Assembly," *Procedia CIRP*, vol. 81, pp. 588–593, 2019, doi: 10.1016/j.procir.2019.03.160.
- [71] L. Zheng, X. Liu, Z. An, S. Li, and R. Zhang, "A smart assistance system for cable assembly by combining wearable augmented reality with portable visual inspection," *Virtual Reality & Intelligent Hardware*, vol. 2, no. 1, pp. 12–27, 2020, doi: 10.1016/j.vrih.2019.12.002.
- [72] M. F. Zäh *et al.*, "Kognitive Assistenzsysteme in der manuellen Montage: Adaptive Montageführung mittels zustandsbasierter, umgebungsabhängiger Anweisungsgenerierung," *wt Werkstattstechnik online*, vol. 97, no. 9, pp. 644–650, 2007. [Online]. Available: <https://mediatum.ub.tum.de/doc/1138574/662911.pdf>
- [73] J. J. G. van Merriënboer and J. Sweller, "Cognitive load theory in health professional education: design principles and strategies," *Medical education*, vol. 44, no. 1, pp. 85–93, 2010, doi: 10.1111/j.1365-2923.2009.03498.x.
- [74] J. Sweller, P. Ayres, and S. Kalyuga, *Cognitive Load Theory*. New York, NY: Springer New York, 2011.
- [75] C. Sennersten and Y. Khaluf, Eds., *COGNITIVE 2016: The Eighth International Conference on Advanced Cognitive Technologies and Applications : March 20-24, 2016, Rome, Italy*. Red Hook, NY: Curran Associates Inc, 2016.

- [76] A. Baddeley, "Working memory and language: an overview," *Journal of Communication Disorders*, vol. 36, no. 3, pp. 189–208, 2003, doi: 10.1016/S0021-9924(03)00019-4.
- [77] A. D. Baddeley, G. J. Hitch, and R. J. Allen, "Working memory and binding in sentence recall," *Journal of Memory and Language*, vol. 61, no. 3, pp. 438–456, 2009, doi: 10.1016/j.jml.2009.05.004.
- [78] C. D. Wickens, "Multiple resources and performance prediction," *Theoretical Issues in Ergonomics Science*, vol. 3, no. 2, pp. 159–177, 2002, doi: 10.1080/14639220210123806.
- [79] J. Ponn and U. Lindemann, *Konzeptentwicklung und Gestaltung technischer Produkte*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [80] R. Diestel, *Graph theory*, 5th ed. Berlin: Springer, 2018.
- [81] M. S. Rahman, *Basic Graph Theory*. Cham: Springer International Publishing, 2017.
- [82] K. Erciyes, *Guide to graph algorithms: Sequential, parallel and distributed*. Cham, Switzerland: Springer, 2018.
- [83] G. Chartrand and S. F. Kapoor, *The Many Facets of Graph Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1969.
- [84] A. Steger, *Diskrete Strukturen: Band 1: Kombinatorik, Graphentheorie, Algebra*, 2nd ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007.
- [85] P. Slepian, *Mathematical Foundations of Network Analysis*. Berlin, Heidelberg: Springer, 1968.
- [86] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning and Data Mining*. Boston, MA: Springer US, 2017.
- [87] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic Algorithms: A Comprehensive Review," in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*: Elsevier, 2018, pp. 185–231.
- [88] B. Chopard and M. Tomassini, *An introduction to metaheuristics for optimization*. Cham, Switzerland: Springer, 2018.
- [89] M. Dorigo, M. Birattari, and T. Stützle, "Metaheuristic," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds., Boston, MA: Springer US, 2017, pp. 817–818.
- [90] Y. Cai, *Instinctive Computing*. London: Springer London, 2016.
- [91] W. Michiels, E. H. L. Aarts, and J. Korst, *Theoretical aspects of local search*. Berlin: Springer, 2007. [Online]. Available: http://deposit.d-nb.de/cgi-bin/dokserv?id=2834321&prov=M&dok_var=1&dok_ext=htm
- [92] S. J. Russell and P. Norvig, *Artificial intelligence: A modern approach*, 3rd ed. Upper Saddle River: Pearson, 2016.
- [93] S. S. Skiena, *The algorithm design manual*. Cham: Springer, 2020.
- [94] U. A.J. and S. P.D., "CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW," *IJSC*, vol. 06, no. 01, pp. 1083–1092, 2015, doi: 10.21917/ijsc.2015.0150.

- [95] P. Larrañaga, C. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators," *Artif Intell Rev*, vol. 13, no. 2, pp. 129–170, 1999, doi: 10.1023/A:1006529012972.
- [96] J. J. Grefenstette, Ed., *Genetic algorithms and their applications: Proceedings of the 2. International Conference on Genetic Algorithms, July 28 - 31, 1987 at the Massachusetts Inst. of Technology, Cambridge, MA*. Hillsdale, NJ: Erlbaum, 1987.
- [97] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
- [98] L. Davis, "Applying Adaptive Algorithms to Epistatic Domains," *International Joint Conference on Artificial Intelligence*, 1985.
- [99] P. LARRANAGA, C. M. H. KUIJPERS, M. POZA, and R. H. MURGA, "Decomposing Bayesian networks: Triangulation of the moral graph with genetic algorithms," *Statistics and Computing*, vol. 7, no. 1, pp. 19–34, 1997, doi: 10.1023/A:1018553211613.
- [100] S. Bandyopadhyay and S. K. Pal, *Classification and learning using genetic algorithms: Applications in bioinformatics and Web intelligence ; with 43 tables*. Berlin u.a: Springer, 2007. [Online]. Available: http://deposit.d-nb.de/cgi-bin/dokserv?id=2874177&prov=M&dok_var=1&dok_ext=htm
- [101] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science (New York, N.Y.)*, vol. 220, no. 4598, pp. 671–680, 1983, doi: 10.1126/science.220.4598.671.
- [102] X.-D. Zhang, *A Matrix Algebra Approach to Artificial Intelligence*: Springer, 2020.
- [103] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems," in *Lecture Notes in Computer Science, Foundations of Fuzzy Logic and Soft Computing*, P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 789–798.
- [104] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artif Intell Rev*, vol. 42, no. 1, pp. 21–57, 2014, doi: 10.1007/s10462-012-9328-0.
- [105] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010, doi: 10.4249/scholarpedia.6915.
- [106] S. Sharma and P. Bhambu, "Artificial Bee Colony Algorithm: A Survey," *IJCA*, vol. 149, no. 4, pp. 11–19, 2016, doi: 10.5120/ijca2016911384.
- [107] L. Lipsky, *Queueing Theory: A Linear Algebraic Approach*. New York, NY: Springer Science+Business Media, LLC, 2009.
- [108] D. Baum, *Grundlagen der Warteschlangentheorie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [109] V. V. Kalashnikov, *Mathematical Methods in Queuing Theory*. Dordrecht: Springer Netherlands, 1994.

- [110] W. Domschke, A. Drexl, R. Klein, and A. Scholl, *Einführung in Operations Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
- [111] D. G. Kendall, "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain," *Ann. Math. Statist.*, vol. 24, no. 3, pp. 338–354, 1953, doi: 10.1214/aoms/1177728975.
- [112] U. Hedtstück, *Simulation diskreter Prozesse*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [113] M. Zukerman, "Introduction to Queueing Theory and Stochastic Teletraffic Models," Jul. 2013. [Online]. Available: <http://arxiv.org/pdf/1307.2968v23>
- [114] U. N. Bhat, *An introduction to queueing theory: Modeling and analysis in applications*. Boston, MA: Birkhäuser, 2015.
- [115] P. Robert, *Stochastic Networks and Queues*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.
- [116] Weitz, *Konkrete Mathematik (nicht nur) für Informatiker*. Springer Fachmedien Wiesbaden, 2018.
- [117] N. T. Thomopoulos, *Probability Distributions: With Truncated, Log and Bivariate Extensions*. Cham: Springer, 2018.
- [118] L. Papula, *Mathematik für Ingenieure und Naturwissenschaftler Band 3*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016.
- [119] K.-H. Waldmann and W. E. Helm, *Simulation stochastischer Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.
- [120] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," 2017.
- [121] C. Speicher, *NeWiP – Neue Wege der informationsgeführten Produktion*, 1st ed. Herzogenrath: Shaker, 2019.
- [122] S. Marx, "2D Camera Calibration," Saarbrücken, Apr. 29 2021.
- [123] Knebl, *Algorithmen und Datenstrukturen*. Springer Fachmedien Wiesbaden, 2019.
- [124] K. Weicker and N. Weicker, *Algorithmen und Datenstrukturen*. Wiesbaden: Springer Vieweg, 2013.
- [125] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.
- [126] "Dijkstra's Algorithm," in *Encyclopedia of Operations Research and Management Science*, S. I. Gass and M. C. Fu, Eds., Boston, MA: Springer US, 2013, p. 428.
- [127] K. Erciyes, *Guide to Graph Algorithms*. Cham: Springer International Publishing, 2018.
- [128] H. Yusnita, M. Zarlis, and Sawaluddin, "Performance of A Star with Dynamic Programming Algorithms in Determining the Shortest Route," in *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies*, Medan, Indonesia, 9162020, pp. 536–542.

- [129] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021, doi: 10.1109/ACCESS.2021.3070054.
- [130] R. Bellman, "On a routing problem," *Quart. Appl. Math.*, vol. 16, no. 1, pp. 87–90, 1958, doi: 10.1090/qam/102435.
- [131] M. Dietzfelbinger, K. Mehlhorn, and P. Sanders, *Algorithmen und Datenstrukturen: Die Grundwerkzeuge*. Berlin: Springer Vieweg, 2014.
- [132] P. Sanders, K. Mehlhorn, M. Dietzfelbinger, and R. Dementiev, *Sequential and parallel algorithms and data structures: The basic toolbox*. Cham, Switzerland: Springer, 2019.
- [133] M. Nebel and S. Wild, *Entwurf und Analyse von Algorithmen: Eine Einführung in die Algorithmik mit Java*, 2nd ed. Wiesbaden: Springer Vieweg, 2018. [Online]. Available: <http://www.springer.com/>
- [134] R. E. Korf, "Depth-first iterative-deepening," *Artificial Intelligence*, vol. 27, no. 1, pp. 97–109, 1985, doi: 10.1016/0004-3702(85)90084-0.
- [135] Bosch Rexroth, *ActiveAssist assistance system - modular, connected, interactive*. [Online]. Available: <https://www.boschrexroth.com/en/xc/products/product-groups/assembly-technology/news/activeassist-assistance-system/index> (accessed: Mar. 3 2022).
- [136] Armbruster Engineering, *ELAM START - IHR EINSTIEG IN ASSISTENZSYSTEME*. [Online]. Available: <https://www.armbruster.de/elam-start/>
- [137] L. Hörauf, *Forschungsberichte Montagetechnik und -organisation: Softwaregestützte Planung cyber-physischer Montagesysteme mittels durchgängiger Informationsnutzung*, 2019.
- [138] N. Privault, *Understanding Markov chains: Examples and applications*. Singapore: Springer, 2018.
- [139] B. Reuter, *Generalisierte Markov-Modellierung: Modellierung irreversibler - amyloid-peptid-dynamik unter mikrowelleneinfluss*. Wiesbaden: Springer, 2020.
- [140] J. E. Marsden, L. Sirovich, S. S. Antman, M. Golubitsky, W. Jäger, and P. Brémaud, *Markov Chains*. New York, NY: Springer New York, 1999.
- [141] S. Dörn, *Programmieren für Ingenieure und Naturwissenschaftler*. Berlin Heidelberg: Springer Berlin Heidelberg, 2018.
- [142] V. Krishnamurthy, "Structural Results for Partially Observed Markov Decision Processes," Dec. 2015. [Online]. Available: <http://arxiv.org/pdf/1512.03873v1>
- [143] M. Komorowski and J. Raffa, "Markov Models and Cost Effectiveness Analysis: Applications in Medical Research," in *Secondary Analysis of Electronic Health Records*, MIT Critical Data, Ed., Cham: Springer International Publishing, 2016, pp. 351–367.
- [144] G. Melli, *HMM Network Instance: Michael Littman's nifty explanatory grid*. [Online]. Available: https://www.gabormelli.com/RKB/HMM_Network_Instance (accessed: Sep. 13 2021).

- [145] N. Bäuerle and U. Rieder, *Markov Decision Processes with Applications to Finance*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [146] M. L. Littman, “A tutorial on partially observable Markov decision processes,” *Journal of Mathematical Psychology*, vol. 53, no. 3, pp. 119–125, 2009, doi: 10.1016/j.jmp.2009.01.005.
- [147] T. Schmidt, *Selbstverständnis und Ziele der Plattform*. [Online]. Available: plattform-lernende-systeme.de/selbstverstaendnis.html (accessed: 2021).
- [148] Normalized Nerd, *Forward Algorithm Clearly Explained | Hidden Markov Model / Part - 6*, 2021. Accessed: Jul. 29 2021. [Online]. Available: <https://www.youtube.com/watch?v=9-sPm4CfcD0>
- [149] K. Tumilaar, Y. Langi, and A. Rindengan, “Hidden Markov Model,” *dC*, vol. 4, no. 1, p. 86, 2015, doi: 10.35799/dc.4.1.2015.8104.
- [150] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989, doi: 10.1109/5.18626.
- [151] W. Khreich, E. Granger, A. Miri, and R. Sabourin, “On the memory complexity of the forward–backward algorithm,” *Pattern Recognition Letters*, vol. 31, no. 2, pp. 91–99, 2010, doi: 10.1016/j.patrec.2009.09.023.
- [152] T. Chis and P. G. Harrison, “Adapting Hidden Markov Models for Online Learning,” *Electronic Notes in Theoretical Computer Science*, vol. 318, pp. 109–127, 2015, doi: 10.1016/j.entcs.2015.10.022.
- [153] B. Pfister and T. Kaufmann, *Sprachverarbeitung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [154] F. Yang, S. Balakrishnan, and M. J. Wainwright, “Statistical and Computational Guarantees for the Baum-Welch Algorithm,” Dec. 2015. [Online]. Available: <http://arxiv.org/pdf/1512.08269v1>
- [155] B. Benyacoub, I. ElMoudden, S. ElBernoussi, A. Zoglat, and M. Ouzineb, “Initial Model Selection for the Baum-Welch Algorithm Applied to Credit Scoring,” in *Advances in Intelligent Systems and Computing, Modelling, Computation and Optimization in Information Systems and Management Sciences*, H. an Le Thi, T. Pham Dinh, and N. T. Nguyen, Eds., Cham: Springer International Publishing, 2015, pp. 359–368.
- [156] T. Schmidt, *Innerbetriebliche Logistik*. Berlin: Springer Vieweg, 2019.
- [157] W. Feller, *An introduction to probability theory and its applications: Volume I*. New Delhi: Wiley-India, 1968.
- [158] D. E. Knuth, *The art of computer programming*, 3rd ed. Upper Saddle River: Addison-Wesley, 2012, cop. 1998.
- [159] K.-H. Wehking, *Technisches Handbuch Logistik 2: Fördertechnik, Materialfluss, Intralogistik*, 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg; Imprint: Springer Vieweg, 2020.
- [160] M. Brookes, *The Matrix Reference Manual: Special Matrices*. [Online]. Available: <http://www.ee.imperial.ac.uk/hp/staff/dmb/matrix/intro.html> (accessed: Mar. 6 2022).

13 Wissenschaftliche Arbeiten

13.1 Eigene Veröffentlichungen

Müller, R., Vette, M., Hörauf, L., Speicher, C., Bashir, A., 2017. Werkerführung verteilt Infos. Handling, Heft 1-2, pp. 50-51

Müller, R., Vette-Steinkamp, M., Hörauf, L., Speicher, C. and Bashir, A., 2018. Intelligent and Flexible Worker Assistance Systems - Assembly Assistance Platform for Planning Assisted Assembly and Rework as Well as Execution of a Worker-Centered Assistance. Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.

Müller, R., Vette-Steinkamp, M., Hörauf, L., Speicher, C. and Bashir, A., 2018. Worker centered cognitive assistance for dynamically created repairing jobs in rework area. Procedia CIRP, 72, pp.141-146.

Müller, R., Müller-Polyzou, R., Hörauf, L., Bashir, A., Karkowski, M., Vesper, D. and Gärtner, S., 2018. Intuitive Bedienung laserbasierter Montageassistentz. Zeitschrift für wirtschaftlichen Fabrikbetrieb, 113(6), pp.363-368.

Müller, R., Hörauf, L., Bashir, A., Speicher, C., Vette-Steinkamp, M. and Mueller-Polyzou, R., 2019. Smart Devices Evaluation and Dynamic Cognitive Assistance System for Repair Processes in Production. Advances in Intelligent Systems and Computing, pp.65-77.

Müller, R., Hörauf, L. and Bashir, A., 2019. Cognitive Assistance Systems For Dynamic Environments. 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFAs).

Müller, R., Vette-Steinkamp, M., Hörauf, L., Speicher, C. and Bashir, A., 2019. Information and data structure to create flexible work plans for worker assistance system at rework site. Procedia CIRP, 79, pp.147-152.

Müller, R., Hörauf, L., Speicher, C. and Bashir, A., 2019. Situational cognitive assistance system in rework area. Procedia Manufacturing, 38, pp.884-891.

Müller, R., Hörauf, L., Bashir, A., Karkowski, M. and Eichenwald, M., 2020. Virtual Reality based Assembly Process Validation and Rework Assistance with consistent Data Exchange. Annals of Scientific Society for Assembly, Handling and Industrial Robotics, pp.1-10.

Müller, R., Hörauf, L. and Bashir, A., 2020. Assembly process prediction with digital assistance systems to ensure synchrony between digital and physical product state using recurrent neural networks. 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFAs).

Kuhn, D., Adelsbach, J., Bashir, A., Hörauf, L., and Müller, R., 2021. Zuweisung bester Abstellplätze im LIFO-Lager unter Berücksichtigung der Kommissionierungsreihenfolge mittels genetischem Algorithmus. In *Simulation in Produktion und Logistik 2021* (pp. 485–494). Cuvillier Verlag.

Muller, R., Horauf, L. and Bashir, A., 2021. An optimization method based on simulated annealing to improve stock reduction in rework with known operator skills. 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).

13.2 Betreute Arbeiten

Daniel Becker: “Entwicklung und Implementierung eines optischen Prüfmoduls zum Einlernen von Referenzbildern am Beispiel metallischer Bauteile in der Montage”, Bachelorarbeit, 2018

Florian Maurer: “Entwicklung einer „intelligenten“ Box (iBox) zur Vermeidung von Fehlern in der Montage von Kleinteilen”, Bachelorarbeit, 2018

Daniel Heil: “Implementierung eines Konzeptes zur Intensionserkennung im Nutzerkontext in der manuellen Montage”, F-und-E Arbeit, 2020

Handharbeni Kuku Prasetyo: “Entwicklung und Umsetzung einer Methode zur optimierten Zuweisung von Nacharbeitsaufträgen an Mitarbeiter in qualifikationsheterogenen Nacharbeitsteams”, Bachelorarbeit, 2021

Andre Schieler: “Entwicklung eines Modells zur Sicherstellung der Synchronisation von realen und digitalen Prozessdaten in einem Kognitiven Assistenzsystem”, Bachelorarbeit, 2021