

# **Das Digitale Kontinuum und die Gestaltung intralogistischer Lern- und Testumgebungen**

Zur Erlangung des akademischen Grades eines  
**Dr.-Ing.**  
von der Fakultät Maschinenbau  
der Technischen Universität Dortmund  
genehmigte Dissertation von

**Anike Murrenhoff, M.Sc.**  
aus  
Gladbeck, Deutschland

Tag der mündlichen Prüfung: 17.04.2026

Erster Gutachter: Prof. Dr. Dr. h. c. Michael ten Hompel  
Zweiter Gutachter: Prof. Dr.-Ing. Johannes Fottner

**Dortmund, 2026**



# Vorwort und Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Fraunhofer-Institut für Materialfluss und Logistik (IML). Bei ihrer Erstellung habe ich Unterstützung von vielen Menschen erhalten, ohne die sie in dieser Form nicht möglich gewesen wäre.

Zunächst möchte ich meinem Erstgutachter, Prof. Dr. Dr. h. c. Michael ten Hompel, für seine Unterstützung und Förderung dieser Arbeit, für die inspirierenden Diskussionen und für seine motivierende Zuversicht danken. Prof. Dr.-Ing. Johannes Fottner möchte ich danken für seine richtungsweisenden Vorarbeiten, die in diese Arbeit eingeflossen sind, für sein Interesse an meinem Forschungsthema und für die Übernahme des Zweitgutachtens.

Mein Dank gilt außerdem meinen Kolleginnen und Kollegen am Fraunhofer IML, am Lehrstuhl für Förder- und Lagerwesen (FLW) der TU Dortmund und am Lamarr-Institut für Maschinelles Lernen und Künstliche Intelligenz. Euer fachlicher Input und die vielen Diskussionen haben diese Arbeit vorangebracht und meine Motivation gestärkt. Besonders danken möchte ich meinen direkten Führungskräften während dieser Zeit, Detlef Spee, Jennifer Beuth und Kira Schmelzpfenning, für die Ermöglichung zeitlicher Gestaltungsspielräume, sowie Dr. Moritz Roidl, Oberingenieur des Lehrstuhls FLW, für seine stetige Ermutigung.

Ganz besonders und persönlich möchte ich mich bei meiner Familie und meinen Freunden bedanken, für ihr Verständnis für meine teils eingeschränkte zeitliche Verfügbarkeit. Vielen Dank, dass ihr mich über die gesamte Zeit mit eurem Rückhalt und eurer Motivation begleitet habt.

Und schließlich danke ich dir, Sebastian. Du hast mich in den vergangenen Jahren stets unterstützt und mir oft den Rücken freigehalten. Dafür danke ich dir sehr.

Dortmund, im Dezember 2025

*Anike Murrenhoff*



# Kurzfassung

Die Autonomie technischer Systeme gilt als zentraler Treiber industrieller Effizienz, da sie die Flexibilität und die Resilienz logistischer Prozesse erhöht. Fortschritte im Bereich der Künstlichen Intelligenz (KI) eröffnen dabei neue Möglichkeiten, Systeme von den heute vergleichsweise niedrigen Autonomiegraden hin zu weitgehend unabhängigen, selbstlernenden Einheiten in den Bereichen Wahrnehmung, Ausführung und Planung weiterzuentwickeln. Vor diesem Hintergrund stellt die vorliegende Arbeit das Leitbild des Digitalen Kontinuums erstmals systematisch und theoretisch fundiert auf. Das Leitbild zielt auf die durchgängige Integration aktiv handelnder KI-Agenten ab, um nahtlose Prozessketten für die Entwicklung und den Betrieb autonomer logistischer Systeme zu ermöglichen.

Für die Realisierung des Digitalen Kontinuums müssen bestehende Unterbrechungen in den Entwicklungs- und Betriebsprozessen autonomer Systeme in der Logistik geschlossen werden. Die Arbeit adressiert eine der zentralen Unterbrechungen am Beispiel Autonomer Mobiler Roboter (AMR) in Materialflusssystemen: die fehlende automatisierte Generierung geeigneter digitaler Lern- und Testumgebungen. Hierfür wird ein Verfahren entwickelt, das auf Methoden der Procedural Content Generation (PCG) zur Generierung der strukturellen Umgebung sowie auf einer serverbasierten Logik zur Erzeugung dynamischer Materialflüsse basiert. Das Verfahren wird hinsichtlich seiner Fähigkeit evaluiert, vielfältige Umgebungsstrukturen zu generieren, dynamische Materialflüsse abzubilden und Robotersteuerungen unmittelbar in die generierten Szenarien einzubinden.

Die Evaluation zeigt, dass das entwickelte Verfahren den Modellierungsprozess digitaler Lern- und Testumgebungen weitgehend automatisiert und als parametrisierbares Werkzeug in das Digitale Kontinuum integrierbar ist. Die Arbeit leistet damit einen konzeptionellen und methodischen Beitrag zur Entwicklung hochautonomer, adaptiver und kontinuierlich lernender logistischer Systeme.



# Abstract

The autonomy of technical systems is considered a central driver of industrial efficiency, as it increases the flexibility and resilience of logistics processes. Advances in artificial intelligence open up new possibilities for developing systems from today's comparatively low levels of autonomy into largely independent, self-learning units in the areas of perception, execution, and planning. In this context, this thesis systematically and theoretically establishes the conceptual framework of the Digital Continuum for the first time. This framework aims at the seamless integration of actively operating AI agents to enable continuous processes for the development and operation of autonomous logistics systems.

To realize the Digital Continuum, existing discontinuities in the development and operation processes of autonomous systems in logistics must be eliminated. This thesis addresses one of the central gaps using autonomous mobile robots (AMR) in material flow systems as an example: the lack of automated generation of suitable digital training and testing environments. To this end, a method is developed that combines procedural content generation (PCG) techniques for generating the structural layout of environments with a server-based logic for creating dynamic material flows. The method is evaluated with regard to its ability to generate diverse environment structures, model dynamic material flows, and directly integrate robot control systems into the generated scenarios.

The evaluation demonstrates that the developed method largely automates the modeling process of digital training and testing environments and can be integrated into the Digital Continuum as a parameterizable tool. The work thus provides both a conceptual and methodological contribution to the development of highly autonomous, adaptive, and continuously learning logistics systems.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b> .....	xiii
<b>Tabellenverzeichnis</b> .....	xvii
<b>Abkürzungs- und Akronymverzeichnis</b> .....	xix
<b>Symbolverzeichnis</b> .....	xxi
<b>1 Einleitung</b> .....	1
1.1 Ausgangssituation .....	1
1.2 Zielstellung und Vorgehen .....	2
<b>2 Grundlagen</b> .....	7
2.1 Materialflusssysteme .....	8
2.1.1 Begriffsabgrenzungen .....	8
2.1.2 Elemente und Struktur von Materialflusssystemen ....	14
2.2 Autonome mobile Roboter in Materialflusssystemen .....	25
2.2.1 Begriffsabgrenzungen .....	26
2.2.2 Anwendungsbereiche von AMR in Materialflusssystemen .....	30
2.2.3 Zu lösende Steuerungsentscheidungen in Materialflusssystemen .....	32
2.2.4 Komponenten und robotisches Steuerungskonzept von AMR .....	35
2.2.5 Steuerungsarchitektur von AMR in Materialflusssystemen .....	38
2.3 Agentenbasierte Künstliche Intelligenz .....	42
2.3.1 Begriffsabgrenzungen .....	43
2.3.2 Methoden für KI-basiertes Lernen .....	46
2.3.3 Entwicklungs- und Betriebsprozesse von KI-Systemen	55

2.4	Modellierung von Materialflusssystemen .....	58
2.4.1	Model-based Systems Engineering .....	58
2.4.2	SysML.....	59
2.5	Procedural Content Generation .....	61
2.5.1	Begriffsabgrenzungen .....	62
2.5.2	Verfahren der PCG .....	63
<b>3</b>	<b>Das Digitale Kontinuum und aktuelle Unterbrechungen .....</b>	<b>65</b>
3.1	Vorwort zum Digitalen Kontinuum .....	66
3.2	Motivation und Hintergrund .....	67
3.3	Kernkonzepte des Digitalen Kontinuums .....	69
3.3.1	KI-basierter Entwicklungszyklus .....	70
3.3.2	Dimensionen der Kontinuität.....	78
3.3.3	Schlüsseltechnologien .....	79
3.4	Existierende Unterbrechungen im Digitalen Kontinuum .....	81
3.4.1	Unterbrechungen in der horizontalen Kontinuität .....	81
3.4.2	Unterbrechungen in der vertikalen Kontinuität .....	82
3.4.3	Unterbrechungen in der kausalen Kontinuität .....	82
3.5	Fazit zum Digitalen Kontinuum und seinen Unterbrechungen .....	85
<b>4</b>	<b>Anforderungen an ein Verfahren zur Generierung von Lern- und Testumgebungen und Forschungslücke .....</b>	<b>87</b>
4.1	Ableitung der Anforderungen .....	88
4.1.1	Generelle Anforderungen aus dem Digitalen Kontinuum .....	88
4.1.2	Anforderungen aus dem Einsatz von AMR in Materialflusssystemen .....	89
4.1.3	Anforderungen aus dem Policy Learning in der Robotik .....	91
4.1.4	Zusammenfassung der Anforderungen .....	92
4.2	Relevante Vorarbeiten und ihre Grenzen .....	94
4.2.1	Generierung von Strukturmodellen von Materialflusssystemen .....	94
4.2.2	Generierung von Simulationsmodellen für Materialflusssysteme .....	98
4.2.3	Generierung von Lernumgebungen für Roboter .....	100
4.3	Forschungslücke.....	102
<b>5</b>	<b>Entwicklung eines Verfahrens zur automatisierten Generierung von Lern- und Testumgebungen für AMR in Materialflusssystemen.....</b>	<b>105</b>
5.1	Grundlegende Generierungsprinzipien .....	106
5.1.1	Übergeordneter Ablauf des Generierungsverfahrens ..	106
5.1.2	Auswahl geeigneter Generierungsansätze .....	107

5.2	Generierung der Struktur der Lern- und Testumgebungen ...	110
5.2.1	Überblick über die Strukturgenerierung .....	110
5.2.2	Generierung der Materialflusssysteme .....	113
5.3	Generierung der Materialflüsse in Lern- und Testumgebungen	137
<b>6</b>	<b>Validierung</b> .....	<b>141</b>
6.1	Ziel und Vorgehen bei der Validierung .....	142
6.2	Implementierung des Generierungsverfahrens .....	142
6.3	Darstellung der Ergebnisse .....	145
6.3.1	Strukturgenerierung .....	145
6.3.2	Materialflussgenerierung .....	148
6.3.3	Lern- und Testfähigkeit robotischer Steuerungsstrategien	150
6.4	Reflexion der Ergebnisse .....	155
6.4.1	Digitales Kontinuum .....	155
6.4.2	Strukturgenerierung .....	155
6.4.3	Materialflussgenerierung .....	156
6.4.4	Lern- und Testfähigkeit robotischer Steuerungsstrategien	157
6.4.5	Schlussfolgerung der Validierung .....	157
<b>7</b>	<b>Fazit und Ausblick</b> .....	<b>159</b>
7.1	Zusammenfassung .....	160
7.2	Ausblick .....	161
	<b>Literaturverzeichnis</b> .....	<b>163</b>
	<b>Anhang</b> .....	<b>179</b>
A.1	Literatur zu Steuerungsproblemen von AMR in der Intralogistik .....	180
A.2	Vorveröffentlichungen und Forschungsprojekte .....	181
A.3	An der Forschungsarbeit beteiligte Personen .....	182
A.4	Studentische Arbeiten im Kontext der Forschungsarbeit .....	183



# Abbildungsverzeichnis

1.1	Struktur der Arbeit .....	4
2.1	Grundbegriffe der Systemtheorie .....	9
2.2	Perspektiven der Systemtheorie .....	10
2.3	Hierarchie logistischer Systeme und Betrachtungsbereich dieser Arbeit .....	11
2.4	Funktionsbereiche in Materialflusssystemen .....	15
2.5	Typischer Aufbau eines Wareneingangs- und Warenausgangsbereichs .....	15
2.6	Typischer Aufbau eines manuellen Lagerungs- und Kommissionierbereichs .....	17
2.7	Schematischer Aufbau von Zeilen- und Blocklagern .....	19
2.8	Typischer Aufbau eines manuellen Verpackungsbereichs .....	20
2.9	Mögliche Gebäudetypen für Stückgutlagerung .....	23
2.10	Dimensionen der Autonomie im ALFUS-Modell .....	27
2.11	Grade der Autonomie .....	28
2.12	AMR-Typen und ihre Fähigkeiten .....	31
2.13	Steuerungsentscheidungen für AMR in Materialflusssystemen ...	33
2.14	Komponenten und Steuerungskonzept eines autonomen Systems.	36
2.15	Paradigmenwechsel von zentral gesteuerten zu dezentral gesteuerten, autonomen Materialflusssystemen .....	38
2.16	Grundstruktur eines Digitalen Zwillings .....	41
2.17	Die Interaktion zwischen Agent und Umgebung in einem MDP ..	47
2.18	Taxonomie von RL-Algorithmen .....	50
2.19	Klassifikation von Simulationsmethoden .....	52
2.20	Visualisierung in Materialfluss- und in Robotiksimulatoren .....	54
2.21	Erweiterung von DevOps zu MLOps .....	55
2.22	MLOps-Architektur und -Workflow .....	56
2.23	Typischer Workflow aus Simulation und ML .....	57
2.24	SysML-Diagrammtypen .....	60

2.25	SysML-Block .....	61
2.26	Beziehungen zwischen Blöcken .....	61
2.27	Ansätze der Procedural Content Generation.....	63
3.1	Entwicklung digitaler Leitbilder von der Industrie 4.0 zum Digitalen Kontinuum.....	67
3.2	Kernkonzepte des Digitalen Kontinuums .....	69
3.3	Phasen im Entwicklungszyklus des Digitalen Kontinuums .....	70
4.1	Zusammenfassung der Anforderungen an die Lern- und Testumgebungen.....	93
4.2	Übersicht über Teilschritte des FLP .....	95
5.1	Grundsätzlicher Ablauf des Generierungsverfahrens .....	106
5.2	Anordnungsarten von Objekten in einem Funktionsbereich .....	108
5.3	Semantisches Modell der Elemente der Lern- und Testumgebungen im Simulationsmodell .....	111
5.4	Ablauf der Strukturgenerierung .....	113
5.5	Semantisches Modell eines Materialflusssystems .....	114
5.6	Modellierung eines Funktionsbereichs und seiner technischen Konfiguration .....	115
5.7	Zufällige Platzierung und Expansion der Funktionsbereiche .....	117
5.8	Semantisches Modell des Gebäudes .....	119
5.9	Bildung der Hauptgänge im Materialflusssystem .....	121
5.10	Parameter für die Bestimmung der Position von Assets in Zeilen .	124
5.11	Technische Ausstattung von Wareneingang und Warenausgang in einer Lern- und Testumgebung .....	126
5.12	Schematische Darstellung der Funktionsbereiche Wareneingang und Warenausgang in einer Lern- und Testumgebung .....	126
5.13	Technische Ausstattung der Lagerung und Kommissionierung in einer Lern- und Testumgebung .....	130
5.14	Schematische Darstellung Lagerung und Kommissionierung in einer Lern- und Testumgebung .....	131
5.15	Technische Ausstattung Verpackung in einer Lern- und Testumgebung .....	133
5.16	Schematische Darstellung des Funktionsbereichs Verpackung in einer Lern- und Testumgebung .....	133
5.17	Technische Ausstattung der Ladestation in einer Lern- und Testumgebung .....	134
5.18	Schematische Darstellung des Funktionsbereichs Ladestation in einer Lern- und Testumgebung .....	135
5.19	Modellierung mobiler Objekte in einer Lern- und Testumgebung .	136

5.20 Modellierung von Transportaufträgen in einer Lern- und Testumgebung .....	138
6.1 Implementierungsarchitektur des Generierungsverfahrens in Nvidia Isaac Sim .....	143
6.2 Benutzeroberfläche der Generierungsanwendung in Nvidia Isaac Sim .....	144
6.3 Roboter in einem generierten Simulationsmodell in Nvidia Isaac Sim .....	145
6.4 Teilschritte der Strukturgenerierung in Nvidia Isaac Sim .....	146
6.5 Boxplots der Strukturgenerierung in Isaac Sim .....	147
6.6 Generierte Transportaufträge in zwei Materialflusssystemen zur Simulationszeit 160 Sekunden .....	148
6.7 Quellen und Senken der generierten Transportaufträge zur Simulationszeit 160 Sekunden .....	149
6.8 Histogramme der Zwischenankunftszeiten der Transportaufträge	150
6.9 Vergleich der gefahrenen und erkannten Wege mit dem Smac-Planner und dem Navfn-Planner, Teil 1 .....	151
6.10 Vergleich der gefahrenen und erkannten Wege mit dem Smac-Planner und dem Navfn-Planner, Teil 2 .....	152
6.11 Vergleich der Simulationszeiten mit dem Smac-Planner und dem Navfn-Planner, Teil 1 .....	153
6.12 Vergleich der Simulationszeiten mit dem Smac-Planner und dem Navfn-Planner, Teil 2 .....	154



# Tabellenverzeichnis

2.1	Basistechnologien für AMR .....	37
2.2	Vergleich zentraler und dezentraler Steuerungsarchitekturen in Materialflusssystemen .....	40
2.3	Vergleich Materialflusssimulation und Robotiksimulation .....	53
6.1	Materialflussmatrix zur Erzeugung der Transportaufträge .....	148
A.1	Literatur zu Steuerungsproblemen von AMR in Materialflusssystemen .....	180
A.2	An der Forschungsarbeit beteiligte Personen .....	182



# Abkürzungs- und Akronymverzeichnis

<b>ALFUS</b>	Autonomy Levels for Unmanned Systems
<b>AMR</b>	Autonome Mobile Roboter
<b>ASMG</b>	Automated Simulation Model Generation
<b>BDD</b>	Block Definition Diagram
<b>BGHW</b>	Berufsgenossenschaft Handel und Warenlogistik
<b>CAD</b>	Computer-aided Design
<b>CI/CD</b>	Continuous Integration, Continuous Delivery
<b>CPS</b>	Cyberphysisches System
<b>DevOps</b>	Development Operations
<b>DR</b>	Domain Randomization
<b>DSRM</b>	Design-Science-Research-Methode
<b>DQN</b>	Deep Q-Networks
<b>FLP</b>	Facility Layout Problem
<b>FTS</b>	Fahrerloses Transportsystem
<b>GAN</b>	Generative Adversarial Network
<b>IMU</b>	Inertial Measurement Unit
<b>IoT</b>	Internet of Things
<b>KI</b>	Künstliche Intelligenz
<b>KLT</b>	Kleinladungsträger
<b>LLM</b>	Large Language Model
<b>LORA</b>	Levels of Robot Autonomy
<b>LSTM</b>	Long Short-Term Memory
<b>MARL</b>	Multi-agent Reinforcement Learning
<b>MBSE</b>	Model-based Systems Engineering
<b>MDP</b>	Markov Decision Problem
<b>ML</b>	Maschinelles Lernen
<b>MLOps</b>	Machine Learning Operations
<b>PCG</b>	Procedural Content Generation
<b>PPO</b>	Proximal Policy Optimization

<b>POMDP</b>	Partially Observable Markov Decision Problem
<b>RFID</b>	Radio-frequency Identification
<b>RL</b>	Reinforcement Learning
<b>ROS</b>	Robot Operating System
<b>SARL</b>	Single-agent Reinforcement Learning
<b>USD</b>	Universal Scene Description
<b>VAE</b>	Variational Autoencoder
<b>VDMA</b>	Verein deutscher Maschinenbau-Anstalten

# Symbolverzeichnis

Symbol	Beschreibung
$A$	Menge an verfügbaren Aktionen eines RL-Agenten
$a_{fb}^{min}$	minimaler Abstand der Mittelpunkte der Funktionsbereiche in einer Lern- und Testumgebung
$a_t$	zum Zeitpunkt $t$ gewählte Aktion eines RL-Agenten
$b_A$	Breite eines zu platzierenden Assets in einer Lern- und Testumgebung
$b_B$	Breite eines mit Objekten zu füllenden Bereichs in einer Lern- und Testumgebung
$b_{fb}^{init}$	initiale Breite eines Funktionsbereichs in einer Lern- und Testumgebung
$b_G$	Breite des Ganges bei Anordnung von Objekten in Zeilen in einer Lern- und Testumgebung
$b_{HG}$	Breite des Hauptganges des Materialflusssystem in einer Lern- und Testumgebung
$b_{HG}^{max}$	maximale Breite des Hauptganges des Materialflusssystem in einer Lern- und Testumgebung
$b_{HG}^{min}$	minimale Breite des Hauptganges des Materialflusssystem in einer Lern- und Testumgebung
$b_{HG}$	Breite des Hauptganges des Materialflusssystem in einer Lern- und Testumgebung
$b_k$	Breite einer Gebäudekachel $k$ in einer Lern- und Testumgebung
$b_{offset}$	Abstand zwischen dem, Rand eines Funktionsbereich und dem Start des Equipments in einem Funktionsbereich in einer Lern- und Testumgebung
$b_{rg}$	Breite des Rangiergangs vor den Toren in einer Lern- und Testumgebung

<b>Symbol</b>	<b>Beschreibung</b>
$b_S$	Breite des Materialflusssystemes $S$ in einer Lern- und Testumgebung
$b_v$	verbleibende Breite eines Funktionsbereichs nach Platzierung der Assets in einer Lern- und Testumgebung
$\epsilon_{fb}$	Füllgrad eines Funktionsbereichs in einer Lern- und Testumgebung
$FB$	Menge der Funktionsbereiche in einer Lern- und Testumgebung
$\gamma_{fb}^{max}$	maximaler Wachstumsfaktor eines Funktionsbereich $fb$ in einer Lern- und Testumgebung
$\gamma_{fb}^{min}$	minimaler Wachstumsfaktor eines Funktionsbereich $fb$ in einer Lern- und Testumgebung
$\gamma_{fb}^x$	Wachstumsfaktor eines Funktionsbereich $fb$ in Richtung der $x$ -Achse in einer Lern- und Testumgebung
$\gamma_{fb}^y$	Wachstumsfaktor eines Funktionsbereich $fb$ in Richtung der $y$ -Achse in einer Lern- und Testumgebung
$g_x$	Ausrichtung der Gänge in einem Funktionsbereich entlang der $x$ -Achse in einer Lern- und Testumgebung
$g_y$	Ausrichtung der Gänge in einem Funktionsbereich entlang der $y$ -Achse in einer Lern- und Testumgebung
$J(\pi)$	erwartete Belohnung der RL-Strategie $\pi$
$K$	Menge der qualifizierten Assets für Gebäudekacheln in einer Lern- und Testumgebung
$\lambda$	Ratenparameter der Exponentialverteilung
$l_A$	Länge eines zu platzierenden Assets in einer Lern- und Testumgebung
$l_B$	Länge eines mit Objekten zu füllenden Bereichs in einer Lern- und Testumgebung
$l_{fb}^{init}$	initiale Länge eines Funktionsbereichs in einer Lern- und Testumgebung
$l_k$	Länge einer Gebäudekachel $k$ in einer Lern- und Testumgebung
$l_S$	Länge des Materialflusssystemes $S$ in einer Lern- und Testumgebung
$l_v$	verbleibende Länge eines Funktionsbereichs nach Platzierung der Assets in einer Lern- und Testumgebung
$LE$	Menge an qualifizierten Assets für Ladeeinheiten in einer Lern- und Testumgebung
$LM$	Menge an qualifizierten Assets für Lagermittel in einer Lern- und Testumgebung
$\mu_\theta$	deterministische RL-Strategie eines Agenten
$M$	Markov Decision Problem

<b>Symbol</b>	<b>Beschreibung</b>
$MO$	Menge an qualifizierten Assets zur Darstellung eines mobilen Objektes in einer Lern- und Testumgebung
$N_{FB}^S$	Anzahl der verschiedenen Typen an Funktionsbereichen in System $S$ einer Lern- und Testumgebung
$N_{MO}^S$	Anzahl und Typen an mobilen Objekten in System $S$ einer Lern- und Testumgebung
$\Omega$	Menge der Beobachtungen in einer Umgebung
$O$	Beobachtungsfunktion eines RL-Agenten
$p$	Stellplatz in einer Lern- und Testumgebung
$\mathcal{P}$	Menge der Übergangswahrscheinlichkeiten der Zustände in einem RL-Problem
$\pi_\theta$	stochastische Strategie eines RL-Agenten
$\pi^*$	optimale stochastische Strategie eines RL-Agenten
$R$	Belohnungsfunktion eines RL-Problems
$r_t$	Belohnung zum Zeitpunkt $t$ in einem RL-Problem
$S$	Menge an möglichen Zuständen einer Umgebung in einem Reinforcement Learning (RL)-Problem
$SP_{fb}$	Menge an Stellplätzen in einem Funktionsbereich $fb$ in einer Lern- und Testumgebung
$s_t$	Zustand der Umgebung zum Zeitpunkt $t$ in einem RL-Problem
$T$	Anzahl an Entscheidungsschritten in einer Episode eines RL-Problems
$\tau$	Sequenz der Zustände und Aktionen in einer Umgebung in einem RL-Problem
$\theta$	Parameterset der Strategie eines RL-Agenten
$V(s)$	Zustandswertfunktion einzelner Zustände in einem RL-Problem
$Q(s,a)$	Q-Wertfunktion der Zustands-Aktions-Paare in einem RL-Problem
$x_{fb}$	x-Koordinate des Mittelpunkts eines Funktionsbereichs $fb$ in einer Lern- und Testumgebung
$x_k$	x-Koordinate des Mittelpunkts einer Gebäudekachel $k$ in einer Lern- und Testumgebung
$x_S$	x-Koordinate des Mittelpunkts des Materialflusssystemes $S$ in einer Lern- und Testumgebung
$x_S^{max}$	maximale x-Koordinate des Materialflusssystemes $S$ in einer Lern- und Testumgebung
$x_S^{min}$	minimale x-Koordinate des Materialflusssystemes $S$ in einer Lern- und Testumgebung
$y_{fb}$	y-Koordinate des Mittelpunkts eines Funktionsbereichs $fb$ in einer Lern- und Testumgebung

<b>Symbol</b>	<b>Beschreibung</b>
$y_k$	y-Koordinate des Mittelpunkts einer Gebäudekachel $k$ in einer Lern- und Testumgebung
$y_S$	y-Koordinate des Mittelpunkts des Materialflusssystems $S$ in einer Lern- und Testumgebung
$y_S^{max}$	maximale y-Koordinate des Materialflusssystems $S$ in einer Lern- und Testumgebung
$y_S^{min}$	minimale y-Koordinate des Materialflusssystems $S$ in einer Lern- und Testumgebung

# Kapitel 1

## Einleitung

### 1.1 Ausgangssituation

Autonome Mobile Roboter (AMR) finden einen zunehmend verbreiteten Einsatz in intralogistischen Systemen. Sie gelten als vielversprechende Technologie zur Steigerung industrieller Flexibilität, Effizienz und Resilienz. Aufgrund ihrer Fähigkeit zur selbstständigen, dezentralen Entscheidungsfindung und zum eigenständigen Handeln werden sie als skalierbar, reaktionsschnell, leistungsfähig sowie als einfach integrierbar und in Betrieb zu nehmen eingeschätzt ([Fra+21, S. 1–3], [Fot+21, S. 2]). Im Vergleich zu stationären Robotern kommt hinzu, dass sie durch ihre Mobilität neben der dynamischen Gestaltung der Betriebssteuerung auch die Umgestaltung der Struktur der Betriebsumgebungen erlauben und so zusätzliche Flexibilität ermöglichen.

Aus diesen Gründen wird die Vision autonomer Systeme im industriellen Einsatz bereits seit der Vorstellung der Industrie 4.0 im Jahr 2011 verfolgt [KW22, S. 1]. Bislang bleibt die vollständige Umsetzung hochautonomer Systeme, einschließlich AMR, jedoch aus. Vollständige Autonomie umfasst die eigenständige Wahrnehmung, Analyse, Planung, Entscheidungsfindung, Ausführung und Kommunikation eines Systems weitgehend unabhängig von Unterstützung durch den Menschen (siehe Unterunterabschnitt 2.2.1.1). Eine Untersuchung von Fottner et al. zu Automatisierungsstufen intralogistischer Technologien und Prozesse zeigt, dass bisher in keinem der untersuchten Aufgabenbereiche die höchste Autonomiestufe erreicht wird [Fot+21].

Künstliche Intelligenz (KI) wird hierbei als entscheidende Schlüsseltechnologie angesehen, um hohe Autonomiegrade zu erzielen [Wah17]. In den vergangenen Jahren haben insbesondere generative KI-Modelle, großskalige Basismodelle (grundlegende, vortrainierte KI-Modelle, engl. Foundation Models) und ihre Fähigkeiten des logischen Schlussfolgerns (engl. Reasoning) erhebliche Fortschritte erzielt. So zeigen etwa Ferrag et al., wie sich große Sprachmodelle (engl. Large Language Model (LLM)) zu autonomen KI-Agenten entwickelt haben [FTD25]. Parallel dazu zeigen auch Liu et al.

Möglichkeiten auf, Gedächtnis-, Weltmodellierungs- und Selbstoptimierungsmechanismen digital abzubilden und damit eine Grundlage für intelligente Agentensysteme zu legen [Liu+25]. Durch die Fähigkeit dieser fortschrittlichen KI-Agenten, multimodale Eingaben zu verarbeiten, Planungs- und Entscheidungsprozesse durchzuführen und eigenständig mehrstufige Aufgabenketten zu bewältigen, eröffnen sie neue Möglichkeiten für die Umsetzung hochautonomer, reaktionsfähiger und robust agierender Systeme, auch in der Intralogistik.

Um die Potenziale dieser aktuellen Fortschritte im Bereich KI-basierter Methoden für die Realisierung hochautonomer logistischer Systeme voll auszuschöpfen, fehlen derzeit jedoch Konzepte zur vollständigen und geschlossenen Integration dieser Methoden in die Betriebs- und die Entwicklungsprozesse. Heutige Konzepte, wie beispielsweise Cyberphysische Systeme (CPS), und Vorgehensweisen, wie Machine Learning Operations (MLOps) und Maschinelles-Lernen-(ML-)Simulationsworkflows, die sich mit der Entwicklung und dem Betrieb autonomer Systeme beschäftigen, werden häufig noch an Stellen unterbrochen, die Automatisierungspotenzial haben. Es bleibt zu untersuchen, wie diese Prozessketten zu einem ununterbrochenen Digitalen Kontinuum verbunden werden können, das eine durchgängige Integration von physischer Realität, digitaler Entwicklungsumgebung und KI-gestützten Entscheidungsprozessen beschreibt (siehe Kapitel 3).

Eine der zentralen Unterbrechungen im angestrebten Digitalen Kontinuum zeigt sich in der bislang nicht automatisierten Bereitstellung vielfältiger, realistischer Lern- und Testumgebungen, die eine wichtige Grundlage für das Training und das Testen autonomer Steuerungsstrategien sind. Die Entwicklung der Strategien in virtuellen Umgebungen ermöglicht die Parallelisierung und Beschleunigung der Entwicklungsprozesse im Vergleich zum physischen System. Gleichzeitig sorgt ein Variantenreichtum in den virtuellen Umgebungen für robustere Steuerungsstrategien, auch in ihrer Übertragung auf die physischen Systeme. Für AMR werden diese virtuellen Lern- und Testumgebungen intralogistischer Systeme heute meist manuell erstellt, was zeit- und ressourcenintensiv ist. Vorhandene Ansätze zur automatisierten Generierung dieser Umgebungen sind häufig nicht ausreichend realistisch, bilden dynamische Aspekte des Verhaltens von Materialflusssystemen nur unvollständig ab oder erfüllen nicht die notwendigen Anforderungen an Systemgrenzen und den Detaillierungsgrad für AMR in der Intralogistik (siehe Kapitel 4).

## 1.2 Zielstellung und Vorgehen

Auf der Grundlage der beschriebenen Ausgangssituation lässt sich die zentrale Zielsetzung der Arbeit folgendermaßen ableiten: *Entwicklung eines Verfahrens zur automatisierten Generierung lauffähiger intralogistischer Systemmodelle*

*für das Lernen und Testen autonomer Steuerungsstrategien im Digitalen Kontinuum.*

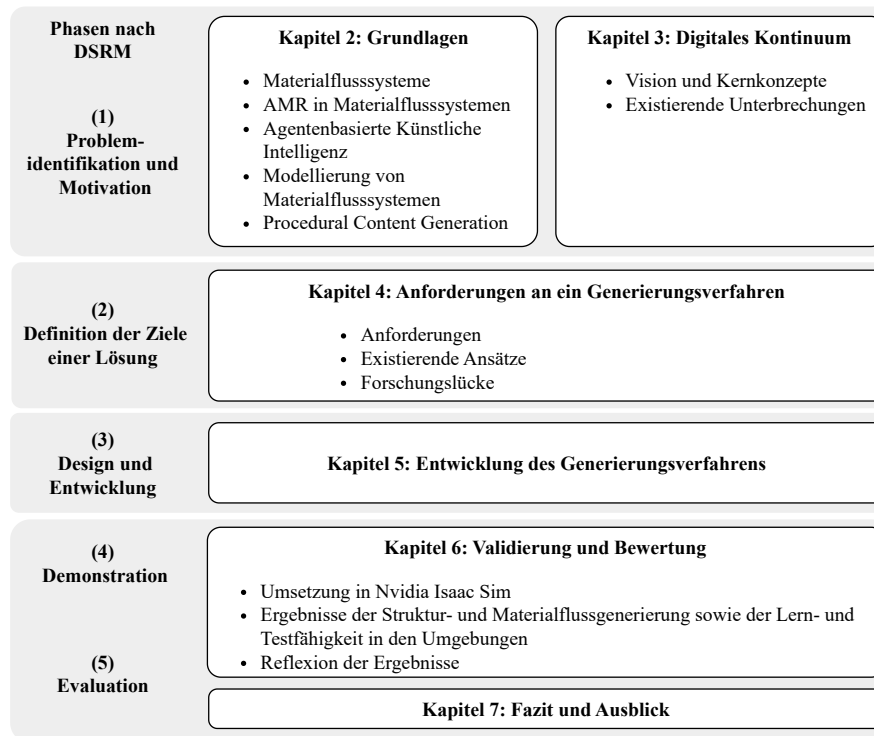
Zur Erreichung dieses Ziels verfolgt diese Arbeit zwei miteinander verknüpfte übergeordnete Forschungsfragen und ihre Teilfragen:

- Wie ist das Digitale Kontinuum als Leitbild definiert, das die Entwicklung und den Betrieb vollständig geschlossener Prozessketten in hochautonomen Systemen auf der Grundlage von Methoden der KI ermöglicht?
  - Welche Unterbrechungen gibt es aktuell in Entwicklungs- und Betriebsprozessen autonomer Systeme?
  - Wie können KI-basierte Lösungen dazu beitragen, sich dem Idealbild des Digitalen Kontinuums anzunähern?
- Wie kann die erforderliche Vielfalt an realistischen Lern- und Testumgebungen für autonome Steuerungsstrategien in der Intralogistik automatisch generiert werden?
  - Wie sehen Bausteine für die automatisierte Generierung von Simulationsumgebungen für intralogistische Systeme aus, in denen AMR eingesetzt werden?
  - Welcher Logik folgt die Zusammensetzung der Bausteine zu sinnvollen und realistischen Systemkonfigurationen?
  - Wie wird dynamisches Verhalten von Materialflusssystemen integriert?

Das Vorgehen zur Erreichung der zentralen Zielsetzung dieser Arbeit wird nach dem Ansatz der Design-Science-Research-Methode (DSRM) nach Pefers et al. strukturiert. Der Ansatz wurde entworfen, um die Entwicklung und Evaluation von Artefakten wie Modellen, Methoden oder Systemen zu unterstützen, die ein praktisches Problem lösen und gleichzeitig wissenschaftlich fundiert sind (siehe Abbildung 1.1) [Pef+07]. Der Ablauf der Methode umfasst sechs Phasen: (1) Problemidentifikation und Motivation, in der ein relevantes, praxisnahes Problem definiert und dessen Bedeutung begründet wird; (2) Definition der Ziele einer Lösung, die die Anforderungen und Erfolgskriterien festlegt; (3) Design und Entwicklung, in der das Artefakt konzipiert und realisiert wird; (4) Demonstration, in der die Funktionsfähigkeit in einer geeigneten Anwendungssituation gezeigt wird; (5) Evaluation, bei der die Zielerreichung überprüft wird; (6) Kommunikation der Ergebnisse an wissenschaftliche und praktische Zielgruppen.

Die sechste Phase ist im Kontext dieser Arbeit bereits durch begleitende Veröffentlichungen umgesetzt worden, die im Anhang dieser Arbeit aufgeführt und an den relevanten Stellen dieser Arbeit referenziert werden (siehe Abschnitt A.2).

Entsprechend der DSRM wird in Kapitel 2 und Kapitel 3 das in der zentralen Zielsetzung betrachtete Problem identifiziert und motiviert. Dazu



**Abb. 1.1** Struktur der Arbeit, orientiert an der DSRM nach [Pef+07]

werden zunächst die Grundlagen aus den Bereichen Materialflusssysteme, AMR in Materialflusssystemen, agentenbasierte KI, Modellierung von Materialflusssystemen und prozedurale Generierung (engl. Procedural Content Generation (PCG)) erläutert. Darauf aufbauend wird anschließend das Konzept des Digitalen Kontinuums entwickelt, eines Leitbilds für kontinuierliche, KI-getriebene Entwicklungs- und Betriebsprozesse für autonome Systeme in der Logistik, und in den Kontext bestehender Digitalisierungsansätze der Industrie 4.0 und des Industrial Metaverse sowie bestehender Entwicklungs- und Betriebsprozesse für KI-Systeme eingeordnet. Dabei werden existierende Unterbrechungen in aktuellen Prozessketten in Entwicklung und Betrieb autonomer Systeme in der Logistik identifiziert.

Eine der identifizierten Unterbrechungen, die automatisierte Generierung einer Vielzahl an variantenreichen Lern- und Testumgebungen für AMR in der Intralogistik, motiviert die zentrale Zielstellung dieser Arbeit. Für die Schließung dieser Unterbrechung werden in Kapitel 4 anschließend die Anforderungen abgeleitet, existierende Ansätze analysiert und die Forschungslücke für diese Arbeit konkretisiert.

Darauf aufbauend wird in Kapitel 5 ein Verfahren zur automatisierten Generierung von Lern- und Testumgebungen für AMR in der Intralogistik konzipiert. Der Ansatz umfasst

- die automatisierte Generierung der Systemstruktur, basierend auf definierten Bausteinen für Materialflusssysteme, und
- die automatisierte Erzeugung von Transportaufträgen, um realistische Materialflusssysteme abzubilden.

In Kapitel 6 wird die technische Umsetzung des entwickelten Ansatzes beschrieben, die in der Simulationsumgebung Isaac Sim (Nvidia Omniverse) erfolgt (siehe [Mak+21]). Zur Validierung werden Auswertungen durchgeführt, in denen der Variantenreichtum der modellierten Systemstrukturen, die Möglichkeit der dynamischen Generierung von Materialflüssen und das Testen von Navigationsalgorithmen in den generierten Umgebungen analysiert und bewertet werden.

Schließlich wird in Kapitel 7 das Fazit zu den Ausarbeitungen gezogen und ein Ausblick auf weiterführende Arbeiten gegeben.

Die angestrebten Beiträge dieser Arbeit liegen somit zusammenfassend in

- der Entwicklung des Digitalen Kontinuums als Leitbild für kontinuierliche, KI-getriebene Entwicklungs- und Betriebsprozesse für autonome Systeme in der Logistik und
- der Konzeption und Umsetzung eines Verfahrens zur automatisierten Erzeugung realitätsnaher Lern- und Testumgebungen für AMR in der Intralogistik.



## Kapitel 2

# Grundlagen

**Zusammenfassung** Dieses Kapitel bildet die Basis für die weiteren Ausführungen dieser Arbeit, die sich mit der Entwicklung eines Leitbilds für die KI-basierte Entwicklung und den Betrieb autonomer Systeme und einem seiner wesentlichen Teilschritte, der automatisierten Generierung variantenreicher Lern- und Testumgebungen intralogistischer Systeme für AMR, befassen. Zunächst werden die grundlegenden Strukturen und Prozesse in logistischen Systemen erläutert, mit besonderem Fokus auf Materialflusssysteme, die den Arbeitsraum der AMR im Sinne dieser Arbeit bilden. Darauf aufbauend werden AMR definiert und ihre Typen sowie Aufgaben in der Intralogistik beschrieben. Anschließend wird die Rolle agentenbasierter KI, des bestärkenden Lernens (engl. Reinforcement Learning (RL)) und der Domänenrandomisierung (engl. Domain Randomization (DR)) für die Autonomisierung dieser Systeme diskutiert. Zudem werden bestehende Vorgehensweisen für die Entwicklung und den Betrieb solcher Systeme vorgestellt. Darauf folgen die Grundlagen zur Modellierung von Materialflusssystemen in diesem Kontext. Abschließend werden die Grundlagen der PCG für die automatische Erzeugung digitaler Inhalte beschrieben.

## 2.1 Materialflusssysteme

In dieser Arbeit wird ein Verfahren entwickelt, mit dem digitale Modelle von Materialflusssystemen automatisiert generiert werden, um sie im Digitalen Kontinuum als Lern- und Testumgebungen für AMR einzusetzen. Materialflusssysteme bilden die Basis zahlreicher betrieblicher Prozesse und sind durch komplexe Strukturen und dynamische Abläufe gekennzeichnet. Um die Entwicklungen im Rahmen dieser Arbeit einordnen zu können, erfolgt im folgenden Abschnitt eine begriffliche Abgrenzung der Bezugsdomäne und des Betrachtungsbereichs dieser Arbeit.

Das Kapitel führt zunächst grundlegende Begriffe aus der Systemtheorie und aus den Bereichen logistischer Systeme und insbesondere der Materialflusssysteme ein. Anschließend werden typische Elemente und der strukturelle Aufbau von Materialflusssystemen beschrieben, um die zentralen Bestandteile zu identifizieren, die ein Verfahren zur Generierung von Lern- und Testumgebungen berücksichtigen sollte.

### 2.1.1 Begriffsabgrenzungen

Materialflusssysteme sind eine spezifische Ausprägung logistischer Systeme. Da der Begriff *Logistiksystem* je nach Kontext unterschiedliche Bedeutungen und Systemgrenzen aufweisen kann, werden im Folgenden zunächst grundlegende Konzepte von Systemen aus dem Bereich der Systemtheorie erläutert. Darauf aufbauend werden die Definition logistischer Systeme und anschließend die Eigenschaften von Materialflusssystemen dargestellt.

#### Systemtheorie

Der Begriff *System* stammt ursprünglich aus dem Griechischen *σύστημα*, das mit „das Zusammengesetzte“ oder „aus Teilen bestehende Einheit“ übersetzt werden kann [Dud25b]. Der Systembegriff findet in zahlreichen wissenschaftlichen Disziplinen Anwendung, darunter Naturwissenschaften, Wirtschaftswissenschaften, Philosophie und Technik.

Abbildung 2.1 veranschaulicht die zentralen Begriffe der Systemtheorie. Ein System besteht demnach aus einzelnen Elementen, die eine gemeinsame Funktion oder einen gemeinsamen Zweck erfüllen [Bla14, S. 3]. Die einzelnen Elemente besitzen spezifische Eigenschaften, beispielsweise physikalische Merkmale wie Abmessungen oder Farben. Sie sind über Beziehungen miteinander verknüpft, die verschiedenster Art sein und etwa Materialflüsse, Informationsflüsse oder Ursache-Wirkungs-Zusammenhänge abbilden können [Hab+21, S. 4]. Entsprechend seiner Funktion oder seinem Zweck wandelt ein System als Einheit aus einzelnen Elementen Eingangsgrößen (Input) in Ausgangsgrößen (Output) um [Hab+21, S. 9].

Wird ein einzelnes Element eines Systems auf einer detaillierteren Ebene betrachtet und wiederum in eigene Elemente und deren Beziehungen

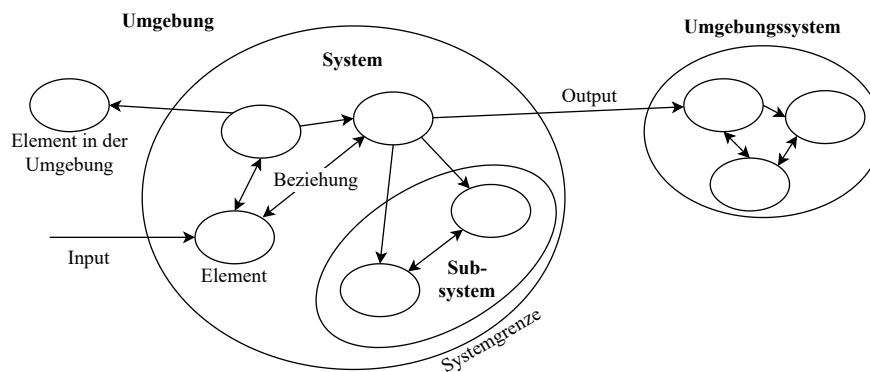


Abb. 2.1 Grundbegriffe der Systemtheorie (i. A. a. [Hab+21, S. 5, 9])

unterteilt, spricht man von einem Subsystem. Umgekehrt kann ein System auch Teil eines übergeordneten Supersystems sein, wenn es gemeinsam mit anderen Elementen eine größere funktionale Einheit bildet [Hab+21, S. 6]. Die Abgrenzung eines Systems von seiner Umwelt erfolgt in der Regel aufgaben-, sinn- oder zweckbezogen. Eine Systemgrenze ist somit nicht absolut, sondern kontextabhängig definiert [Hab+21, S. 9].

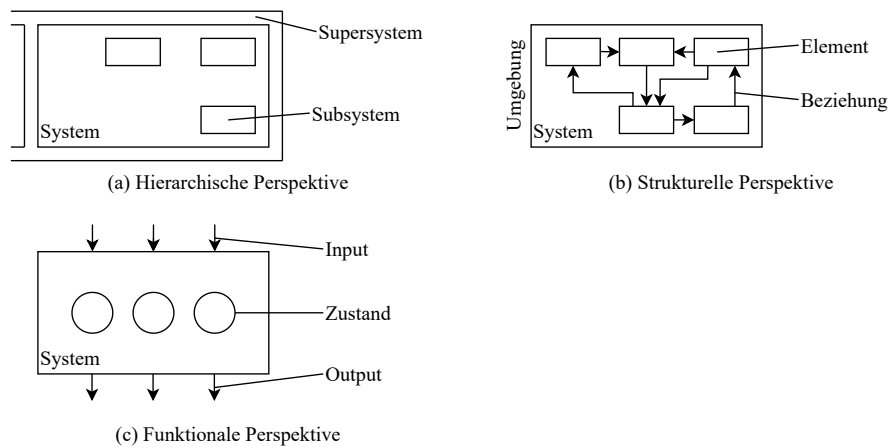
Damit ergibt sich folgendes Verständnis des Begriffs *System* für diese Arbeit:

**Definition: System**

Systeme sind technische Artefakte, die aus einem oder mehreren Elementen und Subsystemen zusammengesetzt sind. Diese Elemente sind aufgaben-, sinn- oder zweckgebunden miteinander verbunden und grenzen sich dadurch von ihrer Umgebung ab. Systeme verarbeiten Eingangsgrößen (Input) zu Ausgangsgrößen (Output), wobei die Art der Umwandlung durch die Funktion oder den Zweck des Systems bestimmt wird. Dabei kann es sich sowohl um physisch realisierte Systeme wie Maschinen oder Anlagen als auch um rein softwarebasierte Systeme handeln.

Abgeleitet von den Konzepten der Systemtheorie lassen sich nach Ropohl drei verschiedene Perspektiven auf Systeme definieren, die für die Ausarbeitungen in dieser Arbeit von Relevanz sind: die hierarchische, die strukturelle und die funktionale Perspektive (vgl. Abbildung 2.2) [Rop09, S. 76].

Die *hierarchische Perspektive* bezieht sich auf die Abgrenzung eines Systems gegenüber seiner Umgebung und damit auf die Festlegung des Detailgrads, auf dem ein System betrachtet wird (vgl. Abbildung 2.2 (a)). Sie basiert auf dem Prinzip, dass jedes System sowohl Teil eines Supersystems sein als auch



**Abb. 2.2** Perspektiven der Systemtheorie [Rop09, S. 76]

selbst aus mehreren Subsystemen bestehen kann. Die Analyse eines Systems auf verschiedenen Hierarchieebenen ermöglicht unterschiedliche Erkenntnisse: Eine Analyse auf niedrigeren Ebenen liefert ein tieferes Verständnis der Funktionsweise der Systemelemente, während die Betrachtung übergeordneter Ebenen zu einem besseren Verständnis des Gesamtzusammenhangs führt. Die hierarchische Perspektive ist sowohl für die Abgrenzung des Betrachtungsraums in dieser Arbeit relevant (vgl. Abbildung 2.3) als auch für die Modellierung der durch das entwickelte Verfahren generierten Modelle (vgl. Unterunterabschnitt 5.2.1.1).

Die *strukturelle Perspektive* leitet sich aus der etymologischen Bedeutung des Systembegriffs ab und betrachtet ein System als Ganzes, das aus mehreren miteinander in Beziehung stehenden Elementen besteht (vgl. Abbildung 2.2 (b)). Zentrales Merkmal dieser Perspektive ist das Prinzip der Emergenz, in dem das System als Ganzes als mehr als die Summe seiner Teile wahrgenommen wird. Die strukturelle Perspektive wird in dieser Arbeit für die Definition der Systemelemente und ihrer Beziehung zueinander herangezogen, die durch den in dieser Arbeit entwickelten Generierungsalgorithmus erstellt werden (vgl. Unterabschnitt 2.1.2).

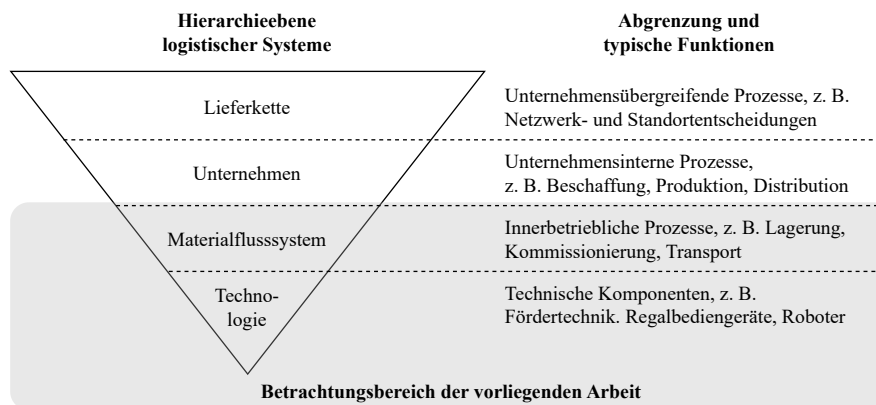
Im Gegensatz zur innengerichteten Betrachtung der strukturellen Perspektive richtet die *funktionale Perspektive* den Fokus auf das Verhalten des Systems in der Interaktion mit seiner Umgebung (vgl. Abbildung 2.2 (c)). Ein System wird hier als Black Box verstanden, deren Zustand über beobachtbare Inputs und Outputs und über interne Zustandsgrößen beschrieben wird. Diese Perspektive ist insbesondere dann hilfreich, wenn weniger die innere Struktur eines Systems, sondern vielmehr seine Funktion im Vordergrund steht. In dieser Arbeit wird die funktionale Perspektive für die Betrachtung von Subsystemen im Materialflusssystem herangezogen, die abstrahiert betrachtet werden.

### Logistische Systeme und Eingrenzung des Betrachtungsbereichs

Logistische Systeme als spezifische Kategorie von Systemen weisen aus der strukturellen Perspektive (vgl. Abbildung 2.2 (b)) typischerweise den Aufbau eines Netzwerks auf, das sich aus Knotenpunkten, wie Lager- und Produktionsstandorten, und Verbindungen zwischen diesen Knoten zusammensetzt. Sie dienen der Durchführung einer Vielzahl unterschiedlicher logistischer Prozesse. Unter logistischen Prozessen werden sämtliche Vorgänge der Beförderung, Lagerung und Handhabung von Waren verstanden, einschließlich des Be- und Entladens, des Ein- und Auslagerns und des Kommissionierens. Gemeinsames Merkmal aller logistischen Prozesse ist ihre Ausrichtung auf die bedarfsgerechte Bereitstellung von logistischen Objekten, bspw. von Ladeeinheiten wie Paletten und Kleinladungsträgern (KLT). [Fle19, S. 2]

Logistische Systeme gelten in der Forschung explizit als komplexe Systeme (vgl. z. B. [Gud12b, S. 37], [Arn08, S. 36]). Ein System wird als komplex bezeichnet, wenn es eine große Anzahl unterschiedlicher Elemente mit dynamischen Beziehungen zueinander aufweist. In derartigen Systemen existieren regelmäßig systemweite Wechselwirkungen zwischen den einzelnen Elementen [Hab+21, S. 11].

Die Systemgrenze eines logistischen Systems ist kontextabhängig und variiert je nach Betrachtungsebene der hierarchischen Perspektive (vgl. Abbildung 2.2 (a)) auf das System. Jedes logistische System kann sowohl in engere Subsysteme untergliedert werden als auch als Bestandteil eines übergeordneten Supersystems betrachtet werden. [Fle19, S. 2]



**Abb. 2.3** Hierarchie logistischer Systeme und Betrachtungsbereich dieser Arbeit (i. A. a. [Gud12a, S. 600])

Abbildung 2.3 zeigt das grundlegende Verständnis der Hierarchieebenen logistischer Systeme im Sinne dieser Arbeit (i. A. a. [Gud12a, S. 600]). Lieferketten bilden dabei die höchste Abstraktionsebene logistischer Systeme. Sie umfassen unternehmensübergreifende logistische Abläufe und Entscheidun-

gen, wie Netzwerk- und Standortentscheidungen. Die Lieferketten setzen sich aus unternehmensbezogenen logistischen Systemen zusammen, in denen die logistischen Abläufe innerhalb eines einzelnen Unternehmens abgebildet werden, bspw. Beschaffungs-, Produktions- und Distributionsprozesse. Auf einer darunterliegenden Ebene befinden sich die Materialflusssysteme, die die innerbetrieblichen Abläufe wie Lagerung, Kommissionierung, Umschlag und Transport umfassen (siehe Abbildung 2.1.1). Materialflusssysteme bestehen im Wesentlichen aus technischen Komponenten wie Fördertechnik, Lagertechnik und Kommissioniertechnik, die einzelnen Subsystemen des Materialflusssystems zugeordnet sind. Jede der Hierarchieebenen logistischer Systeme kann verschiedene technische Systeme, Subsysteme und Elemente enthalten. Ebenso können die Systeme, Subsysteme und Elemente je nach Betrachtungskontext verschiedenen Hierarchieebenen zugeordnet werden.

Der Betrachtungsbereich dieser Arbeit konzentriert sich auf die Hierarchieebenen der Materialflusssysteme sowie auf die darin enthaltenen Subsysteme und technischen Elemente.

Der Fokus liegt hier bewusst auf Materialflusssystemen, die auf vollständig manuelle Prozesse ausgelegt sind und in denen Stückgut abgewickelt wird. Diese Entscheidung basiert darauf, dass manuelle Systeme, im Gegensatz zu vollständig automatisierten Anlagen, auf die flexiblen Fähigkeiten des Menschen ausgerichtet und daher mit einer hohen funktionalen Offenheit gestaltet sind. Sie sind nicht auf starre Prozessstrukturen limitiert, sondern stellen variierende, häufig situationsabhängige Anforderungen an das Handeln, Entscheiden und Kooperieren in der Prozessausführung. Beispiele dafür sind Aufgabenfelder wie das gezielte Ansteuern und Greifen unterschiedlichster Ladeeinheiten, das situationsabhängige Navigieren in blockierten Lagergassen oder das flexible Reagieren auf variierende Auftragslastprofile des Materialflusssystems.

Durch ihre Offenheit und ihre Vielfalt an Aufgaben schaffen manuelle Materialflusssysteme zur Abwicklung von Stückgut somit die geeignete Grundlage für das Trainieren und das Testen sogenannter generalistischer Roboter (vgl. z. B. [McC+24, S. 4]). Denn das Ziel dieser Arbeit ist die Unterstützung der Entwicklung von autonomen Systemen, wie AMR, im Digitalen Kontinuum, die in der Lage sind, in offenen, dynamischen intralogistischen Umgebungen selbstständig zu lernen, zu planen und zu handeln. Die Roboter sollen dabei nicht auf einzelne Anwendungen in Materialflusssystemen spezialisiert sein, sondern sollen in der Lage sein, sich an unterschiedliche Anforderungen und Umgebungen anzupassen.

Der in dieser Arbeit verfolgte Modellierungsansatz zielt daher darauf ab, die strukturelle und funktionale Komplexität manueller Materialflusssysteme zur Abwicklung von Stückgut so aufzubereiten, dass sie als realitätsnahe Grundlage für das Training und das Testen flexibler, autonom agierender Roboter dienen kann.

## Materialflusssysteme

Der Betrachtungsbereich dieser Arbeit, Materialflusssysteme und ihre Subsysteme und Elemente, wird im Folgenden näher beschrieben. Ten Hompel und Heidenblut definieren den Begriff *Materialflusssystem* wie folgt:

„Materialflusssystem ist die Bezeichnung für ein technisches System zum Lagern, Verteilen, Zusammenführen oder Bewegen von Waren und Gütern. Es besteht aus Fördertechnik und Lagertechnik und einem Informations- und Steuerungssystem“ [HH11, S. 193]

Materialflusssysteme umfassen demnach die Prozesse und die technischen Einrichtungen für die innerbetriebliche Materialhandhabung, weshalb der Begriff *intralogistische Systeme* vermehrt synonym genutzt wird. Der Begriff der *Intralogistik* wurde erstmals auf der Pressekonferenz zur Ankündigung der CEMAT 2005 durch den Verein deutscher Maschinenbau-Anstalten (VDMA) in Hannover definiert:

„Die Intralogistik umfasst die Organisation, Steuerung, Durchführung und Optimierung des innerbetrieblichen Materialflusses, der Informationsströme sowie des Warenumschlags in Industrie, Handel und öffentlichen Einrichtungen.“ [Arn06, S. 1]

Im Wesentlichen verknüpft die Intralogistik den Wareneingang, interne Quellen und Senken und den Warenausgang in den Knoten eines logistischen Netzwerks, wie einem Produktionsstandort, Umschlagszentrum, Distributionszentrum o. Ä. [Gud12a, S. 5].

Im Rahmen der vorliegenden Arbeit wird der Begriff Materialflusssystem wie folgt verwendet:

### **Definition:** *Materialflusssystem*

Materialflusssysteme übernehmen die Aufgaben des innerbetrieblichen Materialflusses in Industrie und Handel und bedienen die Funktionen, Waren zu lagern, zu transportieren, zusammenzuführen und zu verteilen. Sie bestehen aus Förder-, Handhabungs- und Lagertechnik sowie zugehörigen Informations- und Steuerungssystemen, die die Organisation, Steuerung, Durchführung und Optimierung der Systeme übernehmen. Im Sinne dieser Arbeit sind die Systeme in abgeschlossenen Gebäuden untergebracht. Die Begriffe *Intralogistik* und *intralogistische Systeme* werden synonym verwendet.

Materialflusssysteme werden als ereignisdiskrete logistische Systeme durch die Anordnung ihrer Elemente (Struktur) und durch ihr Verhalten (Prozesse) charakterisiert (vgl. [Spr+20, S. 17], [Gud12a, S. 599]). Struktur und Prozesse stehen dabei in einem engen Zusammenhang, da die räumliche und funktionale Anordnung der Systemelemente die Voraussetzung für eine effiziente Durchführung logistischer Abläufe bildet.

Als Grundlage für das in dieser Arbeit entwickelte Verfahren zur Generierung intralogistischer Lern- und Testumgebungen werden im Folgenden die zentralen Elemente von Materialflusssystemen und deren strukturelle Anordnung näher betrachtet.

### **2.1.2 Elemente und Struktur von Materialflusssystemen**

Strukturell bestehen Materialflusssysteme aus einer Anordnung von mindestens zwei Gegenständen der Logistik [HSD18, S. 2]. Diese logistischen Gegenstände können einzelne Elemente oder Subsysteme sein, die in verschiedenen Kombinationen eine Vielzahl von intralogistischen Systemen bilden.

Zu den logistischen Elementen sind Güter, Personen, Informationen, Energie, Materialflussmittel, Produktionsmittel, Informationsflussmittel und Infrastruktur wie Gebäude, Flächen und Wege zu zählen [HSD18, S. 2 f.]. Im Allgemeinen kann zwischen den logistischen Elementen, die während ihres Aufenthalts im logistischen System transformiert werden, und solchen, die als Arbeitsmittel zusammen mit der notwendigen Infrastruktur die Transformation der Objekte bewirken, unterschieden werden [HSD18, S. 3].

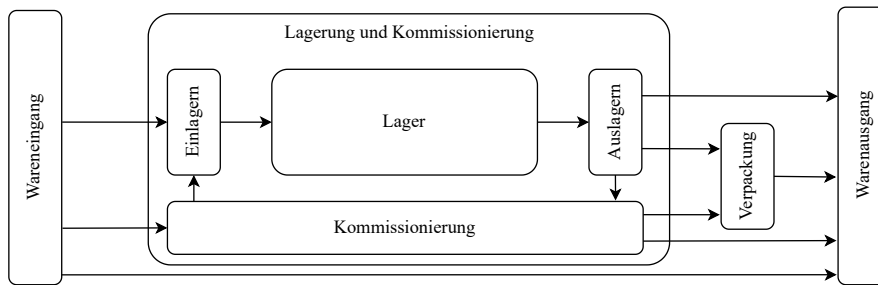
Die Transformationsprozesse in einem Materialflusssystem verändern den Zustand der durchlaufenden logistischen Objekte hinsichtlich Zeit, Ort, Menge, Zusammensetzung und Qualität und werden in den verschiedenen Subsystemen der Materialflusssysteme ausgeführt. Zu den Subsystemen der Intralogistik gehören Verpackungssysteme, Fördersysteme, Lagersysteme, Sortier- und Verteilsysteme, Kommissioniersysteme und Umschlagsysteme. [HSD18, S. 3]

Die Elemente und Subsysteme in Materialflusssystemen sind häufig hierarchisch strukturiert [HSD18, S. 2]. Die hierarchische Struktur der Elemente und Subsysteme äußert sich in einer physischen und logischen Einteilung des Materialflusssystems in Funktionsbereiche, denen die logistischen Elemente und Subsysteme zugeordnet sind.

#### **2.1.2.1 Funktionsbereiche**

VDI 4490 definiert für Materialflusssysteme die Funktionsbereiche Wareneingang, Einlagerung, Lagerung/Nachschub, Kommissionierung/Auslagerung und Versand bzw. Warenausgang [VDI 4490, S. 4]. Zusätzlich werden die Bereiche Qualitätssicherung und Leergutmanagement genannt, die als Querschnittsfunktionen in allen Bereichen des Materialflusssystems integriert sind.

Die Funktionsbereiche, die in dem in dieser Arbeit entwickelten Verfahren betrachtet werden, orientieren sich an [HSD18, S. 55] und [VDI 4490, S. 4] und sind in Abbildung 2.4 dargestellt. Verbunden sind die Funktionsbereiche in einem Materialflusssystem durch Fördersysteme.

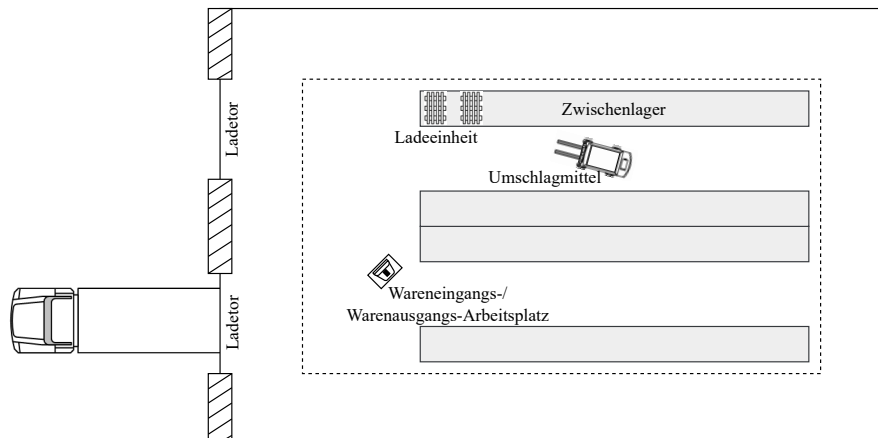


**Abb. 2.4** Funktionsbereiche in Materialflusssystemen (i. A. a. [HSD18, S. 55] und [VDI 4490, S. 4])

Um die Elemente und Subsysteme zu definieren, die als Strukturbausteine der zu generierenden Lern- und Testumgebungen von manuellen Materialflusssystemen zur Auswahl stehen, werden die Funktionsbereiche inklusive ihrer typischen Prozessabläufe und in deren Rahmen eingesetzten Technologien sowie die verbindenden Fördersysteme im Folgenden kurz vorgestellt.

### Wareneingang und Warenausgang

Der Wareneingang und der Warenausgang bilden jeweils eine Schnittstelle des Materialflusssystems zu seiner Systemumwelt, durch die Waren in das System eintreten bzw. austreten. Abbildung 2.5 stellt den Aufbau eines Wareneingangs- bzw. eines Warenausgangsfunktionsbereichs schematisch dar.



**Abb. 2.5** Typischer Aufbau eines Wareneingangs- und Warenausgangsbereichs (i. A. a. [Gud12a, S. 644])

### *Prozessbeschreibung*

Im Wareneingang finden die ersten Prozesse des innerbetrieblichen Materialflusses statt. Zunächst werden ankommende Ladeeinheiten von Transportfahrzeugen, die an den Ladetoren des Materialflusssystemes bereitstehen, entladen. Bei innerbetrieblichen Wareneingängen, beispielsweise aus einer angeschlossenen Produktion, werden die Güter direkt durch interne Fördermittel angeliefert. Anschließend werden die Lieferungen auf Vollständigkeit und Unversehrtheit geprüft und IT-systemseitig vereinnahmt. Falls erforderlich, werden Waren zur Vorbereitung auf die Einlagerung zu neuen Ladeeinheiten umgepackt. Anschließend werden die vereinnahmten Ladeeinheiten für die Einlagerung bereitgestellt. [HSD18, S. 55]

Analog zur Funktion des Wareneingangs stellt der Warenausgang die Schnittstelle zur externen Systemumwelt dar, durch die Güter das Materialflusssystem verlassen. Der Warenausgangsprozess ist aus prozessualer Sicht der finale Schritt im innerbetrieblichen Materialfluss. Der Transport von der Bereitstellung des vorhergehenden Prozesses, z. B. des Konsolidierens und Verpackens, zum Warenausgangsbereich ist die erste Tätigkeit, die dem Prozess zugeordnet wird. Der Warenausgangsprozess gilt als beendet, sobald alle zugehörigen Ladeeinheiten verladen sind.

### *Technische Systeme*

Die Realisierung der Wareneingangsfunktion erfolgt größtenteils durch manuelle oder mechanisierte Prozesse, die Umschlagmittel wie Flurförderzeuge, Rollenbahnen und ähnliche Technologien zu Hilfe nehmen [HSD18, S. 330 ff.]. Für die Zwischenlagerung und Bereitstellung der Waren für die Einlagerung werden Lagersysteme verwendet (siehe Abschnitt Lagerung und Kommissionierung). In der Praxis wird dabei meist die Bodenlagerung umgesetzt [Gud12a, S. 644]. Die informationstechnische Unterstützung der Prozesse erfolgt in der Regel über Arbeitsplätze mit Zugriff auf das verwaltende IT-System. Diese Arbeitsplätze können sowohl stationär als auch mobil ausgestaltet sein.

Die strukturelle und funktionale Ausgestaltung des Warenausgangs gleicht derjenigen des Wareneingangs. [Sch18, S. 50]

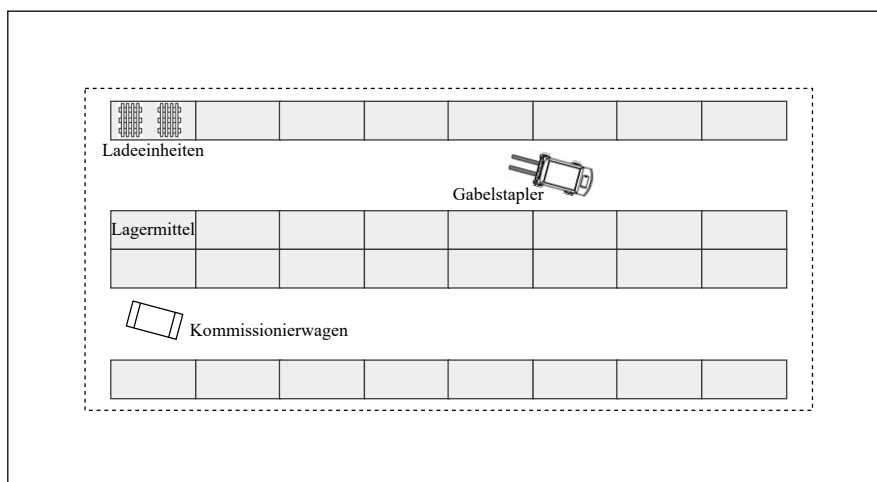
Wareneingang und Warenausgang können in Materialflusssystemen in verschiedenen Kombinationsmöglichkeiten auftreten: Sie können vollständig voneinander getrennt, nebeneinander oder in einem Funktionsbereich kombiniert liegen [Gud12a, S. 645].

### **Lagerung und Kommissionierung**

Nach dem Wareneingang werden die Ladeeinheiten typischerweise in einen vorgesehenen Funktionsbereich zur Lagerung überführt. Eine Ausnahme bilden sogenannte Cross-Docking-Prozesse, bei denen die Ladeeinheiten

ohne Zwischenlagerung vom Wareneingang zum Warenausgang weitergeleitet werden [HSD18, S. 271]. Gemäß VDI-Richtlinie 2411 ist Lagern definiert als das geplante Liegen von Arbeitsgegenständen im Materialfluss ([Arn08, S. 645], [VDI 2411]). Ein Lager ist ein Raum oder eine Fläche, die zum Aufbewahren von Stück- oder Schüttgut in Form von Rohstoffen, Zwischenprodukten oder Fertigwaren genutzt wird [HSD18, S. 51].

In manuell betriebenen Materialflusssystemen findet auch die konventionelle Kommissionierung im Lagerbereich statt [HSB11, S. 66 f.], sodass in der hier vorliegenden Arbeit die Funktionen Lagerung und Kommissionierung dem gleichen Funktionsbereich zugeordnet werden. Das Ziel der Kommissionierung besteht darin, aus einer Gesamtmenge von Gütern Teilmengen aufgrund der Anforderungen der Kommissionieraufträge zusammenzustellen [VDI 3590-1, S. 2]. Abbildung 2.6 zeigt beispielhaft den typischen Aufbau eines manuellen Lagerungs- und Kommissionierbereichs.



**Abb. 2.6** Typischer Aufbau eines manuellen Lagerungs- und Kommissionierbereichs in Zeilenanordnung

### *Prozessbeschreibung*

Der erste Prozess, der dem Funktionsbereich Lagerung und Kommissionierung zugeordnet ist, ist die Einlagerung einer Ladeinheit. Dabei wird die Ladeinheit einem Lagerplatz zugeführt, auf dem sie verweilt, bis sie entweder vollständig ausgelagert wird oder innerhalb des Lagersystems umgelagert wird oder bis einzelne Artikel im Rahmen der Kommissionierung entnommen werden.

Die konventionelle Kommissionierung in manuellen Lagersystemen folgt dem Person-zu-Ware-Prinzip. Dabei bewegen sich die kommissionierenden Personen auf festgelegten Wegen ebenerdig durch die Lagergassen,

entnehmen die jeweiligen Auftragspositionen manuell von den statisch bereitgestellten Ladeeinheiten und legen diese in einen Sammelbehälter. Nach vollständiger Bearbeitung aller Positionen eines Kommissionierauftrags wird der Sammelbehälter an einem definierten Übergabepunkt übergeben, beispielsweise im angrenzenden Verpackungsbereich [HSB11, S. 67].

Der abschließende Prozess im Funktionsbereich Lagerung und Kommissionierung besteht somit entweder in der Auslagerung einer vollständigen Ladeeinheit oder im Abschluss der Kommissionierung. Mit jeder Veränderung des Lagerplatzes oder der Warenmenge auf einer Ladeeinheit erfolgt eine entsprechende Aktualisierung im übergeordneten IT-System, um die Transparenz und Nachverfolgbarkeit des Materialflusses durchgängig sicherzustellen.

### *Technische Systeme*

Für die Durchführung der Prozesse im Funktionsbereich Lagerung und Kommissionierung werden Lager- und Kommissioniersysteme eingesetzt. **Lagersysteme** sind die technischen Subsysteme eines Materialflusssystems im Funktionsbereich Lagerung und Kommissionierung, deren Funktion in der bedarfsgerechten Vorhaltung von Waren und Gütern besteht. Technisch gesehen bestehen sie aus einer Kombination von Lagermitteln, Fördermitteln und ihrem zugehörigen Informationssystem [HSD18, S. 51].

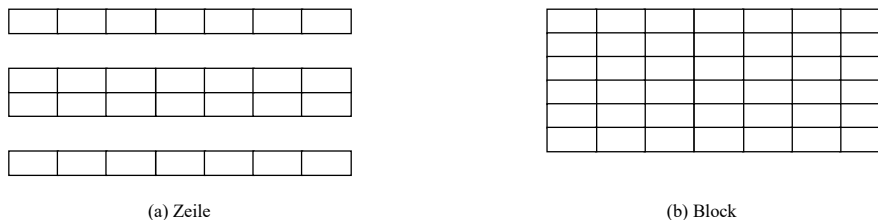
Bei **Kommissioniersystemen** kommen zusätzlich zu diesen technischen Elementen noch die Organisationssysteme hinzu, bestehend aus Aufbau-, Ablauf- und Betriebsorganisation [VDI 3590-1, S. 2]. Durch Variationen in der geometrischen Anordnung, den eingesetzten Technologien und den organisatorischen Abläufen können Kommissioniersysteme auf unterschiedlichste Arten gestaltet werden. Die Bandbreite erstreckt sich von manuellen bis zu vollautomatisierten Lösungen und von einer auftragsorientierten (einstufigen) bis zu einer artikelorientierten (zweistufigen) Kommissionierung. Aufgrund der Vielfalt der technischen Umsetzungsmöglichkeiten wird die Gestaltung von Kommissioniersystemen als äußerst anspruchsvoll angesehen. Übersichten über die verschiedenen Realisierungsformen sind unter anderem in [VDI 3590-1, S. 6], [Gud12a, S. 728], [HSD18, S. 270 ff.] und [Sch20, S. 92] zu finden. Für die Kommissionierung in manuellen Lagersystemen wird im Folgenden die konventionelle Kommissionierung betrachtet (vgl. [HSB11, S. 66 f.]).

Für den im Rahmen dieser Arbeit entwickelten Generierungsalgorithmus ist insbesondere der physische Aufbau manueller Lager- und Kommissioniersysteme von Bedeutung. Daher werden die physischen Elemente zur Durchführung der beschriebenen Lager- und Kommissionierprozesse im Folgenden weiter vorgestellt. Zunächst werden verschiedene manuelle Lagersysteme anhand der verwendeten **Lagermittel** beschrieben. Weiterführende Systematiken zur Klassifikation von Lagersystemen nach Lagermitteln für

Stückgüter finden sich in [HSD18, S. 58] und [Mar16, S. 349]. Anschließend werden die für diesen Kontext relevanten **Fördermittel** näher erläutert.

Grundsätzlich lassen sich manuelle Lagersysteme anhand der **Lagermittel** in Lager mit und ohne Regallagerung unterscheiden [HSD18, S. 58].

Bei der Lagerung ohne Regale werden die Ladeeinheiten direkt auf dem Boden abgestellt. Diese sogenannten Bodenlager lassen sich in Block- und Zeilenlager unterteilen (siehe Abbildung 2.7), die jeweils gestapelt oder ungestapelt ausgeführt sein können. In der Bodenblocklagerung werden Ladeeinheiten direkt nebeneinander, hintereinander und, sofern die Ladeeinheiten stapelbar sind, übereinander gelagert, was eine kompakte Lagerstruktur mit hohem Raumnutzungsgrad ermöglicht. Im Gegensatz dazu sind in der Bodenzeilenlagerung die Ladeeinheiten so angeordnet, dass jeder Stapel direkt an einen Bediengang angrenzt. Dadurch verbessert sich der Zugriff auf einzelne Ladeeinheiten, allerdings auf Kosten des Raumnutzungsgrads.



**Abb. 2.7** Schematischer Aufbau von Zeilen- und Blocklagern

Bei der Regallagerung wird zwischen statischer und dynamischer Lagerung unterschieden. Wie bei der Bodenlagerung ist die statische Regallagerung in Block- und Zeilenlagerung unterteilt, die sich auch hier analog im Raumnutzungsgrad und den Zugriffsmöglichkeiten auf die eingelagerten Ladeeinheiten unterscheiden. Dynamische Regallagerung unterscheidet sich von statischer Regallagerung durch die Beweglichkeit der gelagerten Ladeeinheiten oder der Regale selbst. Die für manuelle Lagersysteme relevante Kategorie ist die der bewegten Ladeeinheiten und feststehenden Regale: Hier werden Durchlauf- und Einschubregale eingeordnet, bei denen die Ladeeinheiten entweder durch Schwerkraft oder durch angetriebene Fördermittel in Kanäle geschoben werden.

Neben den Lagermitteln bestimmen die zu lagernden Ladeeinheiten, bestehend aus Lagergut und Ladehilfsmittel, die physische Struktur des Lagersystems. Klassischerweise lassen sich in der Praxis Palettenlager, Kleinteilelager und Sperrgutlager unterscheiden. In einem Palettenlager werden die Ladeeinheiten gelagert, die Palettengröße aufweisen. Das können Europaletten, Gitterboxen oder sonstige Ladehilfsmittel inklusive Lagergut sein, deren Grundmaß dem eines Palettenmaßes entspricht. In einem Kleinteilelager finden die Lagergüter Platz, die in Paketen, Kleinladungsträgern, Tüten oder

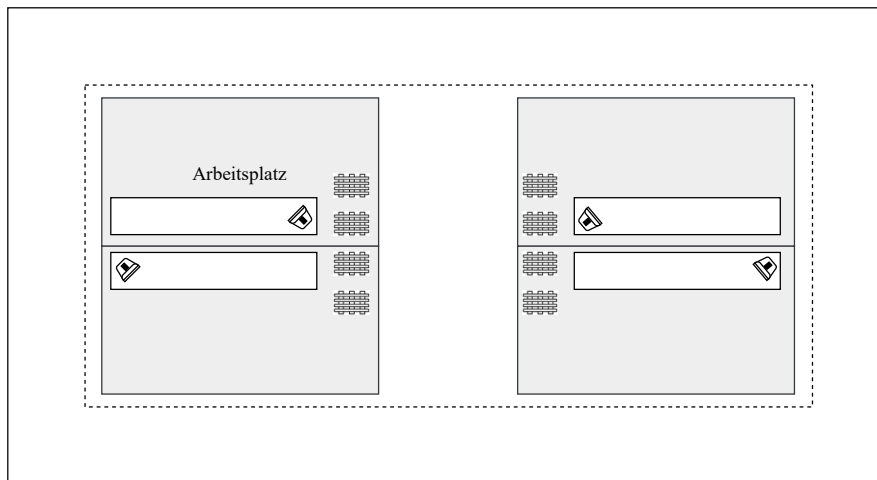
lose gelagert werden. Sperrigteile, wie Rohre oder kundenspezifische Konstruktionen, die die Größe von Paletten überschreiten, werden häufig in separaten Sperrigteillagern in Kragarmregalen oder auf dem Boden gelagert.

Neben Lagermitteln sind auch **Fördermittel** Teil eines Lager- und Kommissioniersystems. Die Fördermittel sind für die Bewegungen von Ladeeinheiten im Lager zur Ein-, Um- und Auslagerung zuständig. In manuellen Lagersystemen erfolgt die Bewegung mit einem Unstetigfördermittel. In Palettenlagern werden dazu beispielsweise Gabelstapler oder Handgabelhubwagen eingesetzt. In manuell bedienten Fachbodenregalen für Kleinteile werden die Ladeeinheiten häufig mit einem Einlager- oder Kommissionierwagen bedient (siehe Abbildung 2.6).

Die für einen Kommissionierauftrag zusammengestellten Artikel werden aus dem Funktionsbereich Lagerung und Kommissionierung anschließend in den Funktionsbereich der Verpackung oder direkt in den Warenausgang gebracht, wo sie potenziell mit weiteren Ladeeinheiten aus anderen Lager- und Kommissionierbereichen für die jeweilige Sendung konsolidiert werden.

### Verpackung

Im Verpackungsbereich eines Materialflusssystemes werden kommissionierte Güter zu versandfertigen Ladeeinheiten zusammengestellt und verpackt. Abbildung 2.8 zeigt exemplarisch den Aufbau eines Verpackungsbereiches.



**Abb. 2.8** Typischer Aufbau eines manuellen Verpackungsbereiches

### *Prozessbeschreibung*

DIN 55405 definiert Verpacken als „Vorgang als Teil des Verpackungsprozesses, bei dem das Packgut mit der Verpackung (Packmittel und Packhilfsmittel) nach einem Verpackungsverfahren mittels Verpackungsmaschinen, -einrichtungen und -geräten oder von Hand zu einer Packung vereinigt wird“ [DIN 55405, S. 2]. Je nach Ausprägung des Packprozesses kann das Befüllen und Etikettieren von Versandkartons sowie das Bilden, Sichern und Etikettieren von Versandpaletten oder speziellen Transportbehältern Teil des Prozesses sein. Das Verpacken fällt häufig mit der Konsolidierung von Gütern aus verschiedenen Kommissionierbereichen zusammen.

### *Technische Systeme*

Die Bandbreite technischer Ausführungen im Bereich der Verpackung reicht von vollständig manuellen Prozessen bis hin zu Anlagen mit automatisierter Sortier- und Verteiltechnik zur Konsolidierung sowie automatischen Systemen zur Sicherung und Etikettierung von Ladeeinheiten [HSD18, S. 5 ff.].

Manuelle Verpackungsbereiche sind in der Regel mit spezifisch ausgestatteten Arbeitsplätzen versehen. Diese umfassen Arbeitstische, Vorrichtungen zur Bereitstellung von Verpackungsmaterialien wie Kartonagen, Füllmaterial und Klebeband sowie technische Ausstattungen wie Bildschirme mit Zugang zum Informationssystem, Drucker, Scanner oder Waagen.

Für die räumliche Anordnung der Arbeitsplätze innerhalb des Funktionsbereichs Verpackung bestehen unterschiedliche Möglichkeiten. Ein Arbeitsplatz, an dem alle für einen bestimmten Verpackungsprozess erforderlichen Tätigkeiten von einer einzelnen Person ausgeführt werden, wird als Einzelverpackungsarbeitsplatz bezeichnet. Häufig sind mehrere dieser Arbeitsplätze parallel zueinander angeordnet. Eine gängige und kompakte Form der Arbeitsplatzgestaltung stellen sogenannte Doppelpackplätze dar (siehe Abbildung 2.8) [Dze08, S. 20].

Nach dem Verpacken werden die versandfertigen Ladeeinheiten in den Funktionsbereich Warenausgang überführt, wo sie die finalen Prozessschritte im Materialflusssystem durchlaufen, bevor sie das System verlassen (siehe Abschnitt Wareneingang und Warenausgang).

### **Ladebereich für Fördersysteme**

Fördersysteme stellen einen wesentlichen Bestandteil von Materialflusssystemen dar. Sie können entweder eigenständig agieren und verschiedene Subsysteme und Bereiche in einem Materialflusssystem verbinden oder Teil anderer Systeme wie Lager-, Handhabungs- und Kommissioniersysteme sein (siehe z. B. Abschnitt Lagerung und Kommissionierung).

### *Prozessbeschreibung*

Fördersysteme dienen dem Verteilen, Sammeln, Sortieren, Puffern und Zwischenlagern von Gütern. Zu einem Fördervorgang gehören die Aufnahme einer Transporteinheit, das Fördern von einem Quellort zu einem Zielort und die Abgabe der Transporteinheit. Das eigentliche Fördern wird gemäß DIN 30781, Teil 1 als der Transport innerhalb einer örtlich begrenzten und zusammenhängenden Betriebseinheit definiert, wie zum Beispiel innerhalb eines Werks, eines Flughafens, einer Grube, eines Lagers oder eines Büros [DIN 30781-1, S. 2].

### *Technische Systeme*

Ein Fördersystem besteht in der Regel aus einem oder mehreren Fördermitteln. Der Begriff *Fördermittel* umfasst dabei sämtliche mechanischen Hilfsmittel zur Förderung von Gütern. Förderzeuge, die in Materialflusssystemen eingesetzt werden, in denen hauptsächlich manuelle Prozesse durchgeführt und Stückgüter abgewickelt werden, lassen sich den flurgebundenen, frei verfahrbaren Unstetigförderern zuordnen. Dazu zählen beispielsweise Hubwagen, Stapler, Kommissionierfahrzeuge und Wagen [HSD18, S. 129].

Im Bereich der intralogistischen Fördersysteme ist in den letzten Jahren ein Trend zum Einsatz von AMR erkennbar. AMR kombinieren die Mobilität flurgebundener Fördermittel mit autonomen Navigations- und Entscheidungsfähigkeiten, sodass sie flexibel auf Änderungen in der Umgebung reagieren und Aufgaben ohne direkte menschliche Steuerung ausführen können (siehe Abschnitt 2.2). AMR können gemeinsam mit manuellen Fördersystemen in Materialflusssystemen betrieben werden. Für die in dieser Arbeit entwickelte Methodik zur Generierung von Lern- und Testumgebungen ist die Abbildung sowohl autonomer als auch manueller Fördersysteme von Bedeutung, um potenziell Interaktionen und mögliche Konfliktsituationen zwischen den verschiedenen Fördermitteltypen abbilden zu können.

Beim Einsatz elektrisch betriebener Flurförderzeuge ist ein entsprechender Ladebereich innerhalb des Materialflusssystems integriert. Dieser Bereich dient der regelmäßigen Aufladung der Fahrzeugbatterien und muss in die räumliche und infrastrukturelle Planung eines Materialflusssystems integriert werden.

#### **2.1.2.2 Logistikgebäude**

Materialflusssysteme befinden sich in der Regel in Industriegebäuden, deren bauliche Ausführung bestimmten Normen und Vorgaben folgt. Dadurch weisen sie häufig ähnliche Elemente auf. Für die Generierung variantenreicher und realistischer Lern- und Testumgebungen ist eine Abbildung dieser Elemente von Bedeutung. Die Aufgaben, die AMR in den simulierten Umgebungen erlernen sollen, basieren auf Sensordaten (siehe Abschnitt 2.2),

die nicht nur die Elemente im Inneren des Materialflusssystem, sondern auch die Ausstattung des umgebenden Gebäudes erfassen. Eine realitätsnahe Abbildung des Gebäudes ist daher essenziell, um die Übertragbarkeit digital entwickelter Steuerungsstrategien auf reale Einsatzumgebungen zu verbessern, insbesondere bei Aufgaben wie der Ortung und der Navigation.

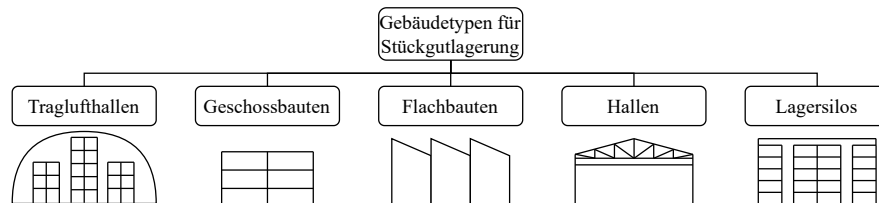


Abb. 2.9 Mögliche Gebäudetypen für Stückgutlagerung [Mar16, S. 349]

Für die Lagerung von Stückgut lassen sich fünf typische Gebäudetypen unterscheiden: Traglufthallen, Geschossbauten, Flachbauten, Hallen und Lagersilos (siehe Abbildung 2.9) [Mar16, S. 349]. In der Praxis dominieren Flachbauten und Hallen, die sich im Wesentlichen durch ihre Raumhöhe unterscheiden [LR11, S. 67 ff.]. Flachbauten weisen in der Regel Höhen bis etwa fünf Meter auf, während Hallen meist mit Höhen von zwischen fünfzehn und fünfundzwanzig Metern gebaut werden. Da Palettenregale in den in dieser Arbeit betrachteten Materialflusssystemen eine Rolle spielen und eine Höhe von bis zu sechzehn Metern erreichen können [HSD18, S. 68], liegt der Fokus im Folgenden auf dem Gebäudetyp Halle.

Hallen werden meist mit einem rechteckigen Grundriss geplant, da sie aus einzelnen Hallenschiffen zusammengesetzt werden [LR11, S. 71]. Die Elemente im Inneren einer Halle, die durch die Sensorik von Robotern wahrgenommen werden können und daher von Relevanz für die Generierung von Lern- und Testumgebungen sind, lassen sich den Kategorien Tragwerkselemente, Gebäudehüllelemente, Ausbauelemente, Haustechnik und Brandschutz zuordnen [LR11, S. 73-79].

### Tragwerkselemente

Zu den sichtbaren Tragwerkselementen im Inneren einer Halle gehören die Elemente der Skelettkonstruktion wie Stützen, Binder, Träger, Rahmen, Bogen, Schalen und Fachwerkträger sowie Elemente der Aussteifung und Verbindungen wie Nieten und Schrauben [LR11, S. 74].

### Gebäudehülle

Die von innen sichtbaren Bestandteile der Gebäudehülle umfassen Wand-, Decken- und Bodenflächen, bei deren Konstruktion häufig Materialien wie Beton, Glas, Metall oder Kunststoff verwendet werden [LR11, S. 75 f.]. Besonders der Boden muss in Materialflusssystemen spezifische Anforderungen

erfüllen, etwa hinsichtlich Ebenheit, Belastbarkeit und Rutschfestigkeit, um einen sicheren Betrieb flurgebundener Technologien zu gewährleisten (vgl. [VDI 2510-1, S. 4 ff.], [ASR A1.5]).

### **Ausbauelemente**

Zu den Ausbauelementen zählen Öffnungen in Dach und Wänden, Türen, Tore, Treppen, Rampen und Stege [LR11, S. 76]. Neben Fenstern ist insbesondere die Konfiguration der Ent- und Beladebereiche von Bedeutung, da sie die Schnittstellen zum externen Transport darstellen. Diese Zonen befinden sich in der Regel im Wareneingang und Warenausgang eines Materialflusssystemes und werden durch Rampen- und Torausstattung definiert. Rampen ermöglichen eine niveaugerechte Übergabe zwischen Lagerbereich und Transportfahrzeug. Bei ihrer Realisierung sind Vorschriften wie die ArbStättV §21, VBG §32, VDI 2199 und ZH 1 156 zu berücksichtigen [HSD18, S. 325], die ihre Konfiguration und ihr Erscheinungsbild beeinflussen.

Die Ent- und Beladebereiche können in einer Reihe unterschiedlicher Ausführungsformen umgesetzt werden. Rampen können als Seitenrampen, Kopframpen, Rampen in Sägezahnform oder als Dockrampen ausgeführt sein. Für den Höhenausgleich zwischen Boden und Fahrzeug kommen Überladebleche, Überladebrücken oder Hebebühnen zum Einsatz. Bei den Toren unterscheidet man zwischen Dreh-, Falt-, Schiebe-, Roll- und Sektionaltoren, wobei die Auswahl von baulichen Gegebenheiten und gewünschten Eigenschaften wie Schnelligkeit oder Sicherheit abhängt. Eine detaillierte Beschreibung findet sich in [HSD18, S. 326 ff.].

### **Haustechnik**

Auch die sichtbaren Elemente der technischen Gebäudeausstattung sind für die Generierung realistischer Umgebungen von Bedeutung. Dazu zählen unter anderem Heizungs- und Lüftungssysteme, Beleuchtung, Rohre, Stromversorgung, Kommunikations- und Netzwerkinfrastruktur sowie Versorgungsleitungen für Druckluft und weitere Medien ([LR11, S. 77]; vgl. auch [Mar16, S. 124]). Häufig verlaufen diese Installationen entlang von Decken und Wänden und stellen charakteristische Merkmale dar, die durch Sensoren erkannt und zur Orientierung genutzt werden können.

### **Brandschutz**

Schließlich sind auch sicherheitsrelevante Systeme Teil der Gebäudeinfrastruktur. Dazu zählen Brandmeldeanlagen, Feuerlöschsysteme, Notbeleuchtung und Alarmtechnik, die dem Schutz von Personal, Anlagen und Gütern dienen [LR11, S. 79]. Auch automatische Löschsyste me wie Deckensprinkleranlagen gehören zur baulichen Ausstattung einer Logistikhalle, abhängig von der Brandlast des im Materialflusssystem gelagerten Guts [Gud12a, S. 638, 646].

Die konkrete Ausprägung der baulichen und technischen Merkmale einer Logistikhalle ist in der Realität mit den zuständigen Behörden abzustimmen und folgt daher nicht immer vorhersehbaren Anforderungen. Für die Generierung variantenreicher Lern- und Testumgebungen stellen diese baulichen Merkmale jedoch ein Potenzial dar, gezielte Variationen einzuführen. Dadurch können in den generierten Modellen erlernte Verhaltensweisen robuster gegenüber den in der Realität auftretenden Bandbreiten physischer Gebäudeausprägungen werden.

## 2.2 Autonome mobile Roboter in Materialflusssystemen

In dem vorangegangenen Abschnitt werden Materialflusssysteme beschrieben, in denen überwiegend manuelle Prozesse durchgeführt werden. Aufgrund ihrer hohen Freiheitsgrade in der Prozessgestaltung und Entscheidungsfindung bieten diese Systeme eine umfassende Grundlage für das Erlernen von Verhaltensweisen in virtuellen Lern- und Testumgebungen für autonome robotische Systeme.

In der Praxis haben sich die in Materialflusssystemen eingesetzten Technologien bereits deutlich über rein manuell durchgeführte Prozesse hinaus entwickelt. In den vergangenen Jahrzehnten ist ein stetiger Wandel von manuell geprägten Systemen hin zu automatisierten und zunehmend autonomen technischen Lösungen zu beobachten [Fra+21, S. 1]. Während zu Beginn vor allem menschliche Arbeitskraft den Materialfluss innerhalb intralogistischer Umgebungen ausführte, setzte mit dem Einzug automatisierter Förder- und Lagersysteme ab den 1970er-Jahren eine erste Welle der Technisierung ein. Diese automatisierten Technologien sind heute in vielen Materialflusssystemen in der Industrie integriert. Sie zeichnen sich durch fest installierte Systeme aus, in denen klar definierte Prozesse durch Maschinen ausgeführt werden, die jedoch in ihrer Anpassungsfähigkeit an neue Abläufe, Störungen und variierende Leistungsanforderungen stark begrenzt sind.

Mit den Fortschritten in Robotik, Sensorik, KI und drahtloser Kommunikation vollzieht sich derzeit ein nächster Entwicklungsschritt: die Integration autonomer robotischer Systeme in intralogistische Prozesse. Diese Systeme sollen in der Lage sein, sich in dynamischen Umgebungen selbstständig zu orientieren, Aufgaben zu planen und auszuführen sowie mit anderen Systemkomponenten zu interagieren. Im Gegensatz zu klassisch automatisierten Anlagen verfügen autonome Systeme über eine höhere Flexibilität, Skalierbarkeit und Selbstständigkeit. [Fra+21]

Der zunehmende Einsatz autonomer Systeme in Materialflusssystemen lässt sich auf mehrere strukturelle und wirtschaftliche Entwicklungen zurückführen. Dazu zählen zum Beispiel der anhaltende Fachkräftemangel und steigende Anforderungen an Effizienz und Robustheit. Autonome Tech-

nologien wie AMR, kollaborative Greifarme und autonome Serviceroboter übernehmen bereits heute teils Aufgaben, die bisher durch Menschen ausgeführt wurden.

Dennoch erreichen technische Systeme bislang keine vollständige Autonomie in allen ihren Anwendungs- und Aufgabenbereichen [Fot+21]. An dieser Stelle wird KI als Schlüsseltechnologie gesehen, um höhere Grade an Autonomie erreichen zu können. Für den Einsatz KI-basierter Methoden in diesem Kontext wird eine Vielzahl an variablen, virtuellen Lern- und Testumgebungen benötigt.

In diesem Abschnitt werden daher die grundlegenden Eigenschaften und Einsatzmöglichkeiten von AMR im Kontext von Materialflusssystemen dargestellt, um darauf aufbauend in Kapitel 4 Anforderungen an virtuelle Lern- und Testumgebungen für diese Systeme ableiten zu können. Es werden zunächst der Begriff der Autonomie im Kontext technischer Systeme definiert und verschiedene Grade der Autonomie aufgezeigt. Anschließend werden typische intralogistische Anwendungsgebiete aufgezeigt, in denen AMR-Systeme zum Einsatz kommen, und die Steuerungsentscheidungen erläutert, die sie in diesen Anwendungsbereichen treffen müssen. Darauf aufbauend werden der physische Aufbau und die Steuerungsmechanismen von AMR erläutert.

### 2.2.1 Begriffsabgrenzungen

Im Folgenden wird der Begriff der Autonomie im Kontext technischer Systeme definiert und das Konzept verschiedener Grade der Autonomie erläutert.

#### 2.2.1.1 Autonome Systeme

Der Begriff *Autonomie* stammt aus dem Altgriechischen und bedeutet „Eigengesetzlichkeit“ oder „Selbstständigkeit“ [Dud25a]. In verschiedenen fachlichen Kontexten wird er unterschiedlich konkretisiert, wodurch sich eine Vielzahl an Definitionen ergibt, auch in Bezug auf technische Systeme.

Eine bedeutende Initiative zur begrifflichen Klärung wurde Anfang der 2000er-Jahre in den Vereinigten Staaten durch die Initiative *Autonomy Levels for Unmanned Systems (ALFUS)* angestoßen, deren Ziel es war, eine domänenübergreifende Definition für Autonomie im Kontext unbemannter technischer Systeme zu entwickeln (vgl. u. a. [Hua+04a; Hua+04b; Hua+05; Hua+07]). Die Definition wurde u. a. von Beer et al. aufgegriffen und explizit für Roboter im Kontext der Mensch-Roboter-Interaktion übernommen [BFR14]. Diese Definition bildet die konzeptionelle Grundlage für das Verständnis von Autonomie in dieser Arbeit:

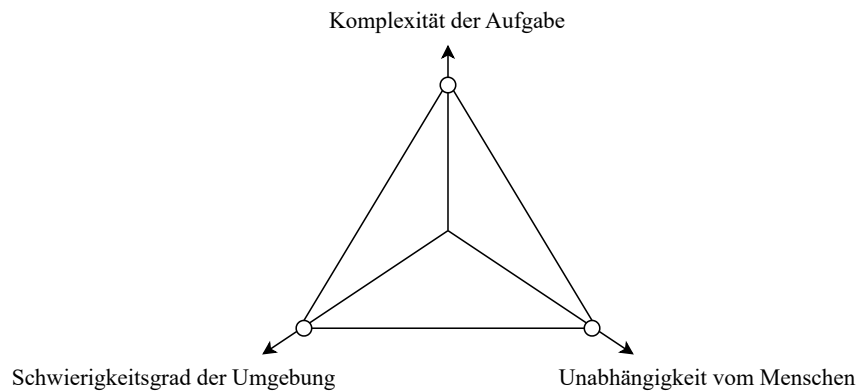
**Definition:** *Autonomie, autonomes System*

Autonomie bezeichnet die Fähigkeit eines technischen Systems, ein ihm zugewiesenes Ziel eigenständig zu erreichen. Die Zielvorgabe kann dabei durch Menschen oder durch ein anderes technisches System über eine Kommunikationsschnittstelle erfolgen. Um das Ziel zu erfüllen, ist das System in der Lage, seine Umgebung selbstständig wahrzunehmen, Informationen zu analysieren, mit anderen Einheiten zu kommunizieren, Handlungen zu planen, Entscheidungen zu treffen und diese eigenständig auszuführen, ohne kontinuierliche externe Steuerung.

Entsprechend der Definition des Begriffs *System* kann ein technisches System in diesem Zusammenhang sowohl ein physisch handelndes, robotisches System als auch ein digital handelndes Softwaresystem auf einem Server sein.

**2.2.1.2 Grade der Autonomie**

Autonome Systeme können unterschiedliche Grade an Autonomie erreichen. Im Rahmen des ALFUS-Modells werden Autonomiestufen definiert, anhand derer sich der Grad der Autonomie technischer Systeme einordnen lässt. Dabei wird die Autonomiefähigkeit eines Systems als abhängig von drei zentralen Faktoren betrachtet: der Komplexität der auszuführenden Aufgabe, dem Schwierigkeitsgrad der Umgebung und dem Grad an Unabhängigkeit vom Menschen (siehe Abbildung 2.10).



**Abb. 2.10** Dimensionen der Autonomie im ALFUS-Modell [Hua07, S. 48]

Die *Komplexität einer Aufgabe* kann unter anderem anhand ihrer strukturellen Zusammensetzung, der erforderlichen Kommunikation und Zusammenar-

beit zwischen beteiligten Systemelementen und dem notwendigen Maß an Situationsbewusstsein für ihre erfolgreiche Ausführung beurteilt werden [Hua+04b, S. 6]. So ist es beispielsweise deutlich einfacher für einen einzelnen Transportroboter, sich autonom von Punkt A nach Punkt B zu bewegen, als für ein gesamtes Materialflusssystem, eigenständig eine Neugestaltung der Einlagerungsstrategie vorzunehmen.

Auch der *Schwierigkeitsgrad der Umgebung* beeinflusst den Autonomiegrad eines technischen Systems. Aufgaben in einer statischen und kontrollierten Umgebung lassen sich in der Regel einfacher bewältigen als solche in dynamischen, unstrukturierten und unsicheren Umgebungen, in denen mit einer Vielzahl ungeplanter Ereignisse gerechnet werden muss [Hua+04b, S. 9].

Ein weiterer zentraler Faktor für die Bestimmung des Autonomiegrads eines technischen Systems ist die *Unabhängigkeit vom Menschen*. Sie lässt sich unter anderem anhand der Häufigkeit und Dauer menschlicher Eingriffe, des erforderlichen Qualifikationsniveaus des Bedienpersonals sowie des Verhältnisses zwischen der Anzahl an menschlichen Operatoren und autonomen Systemen quantifizieren [Hua+04b, S. 11].

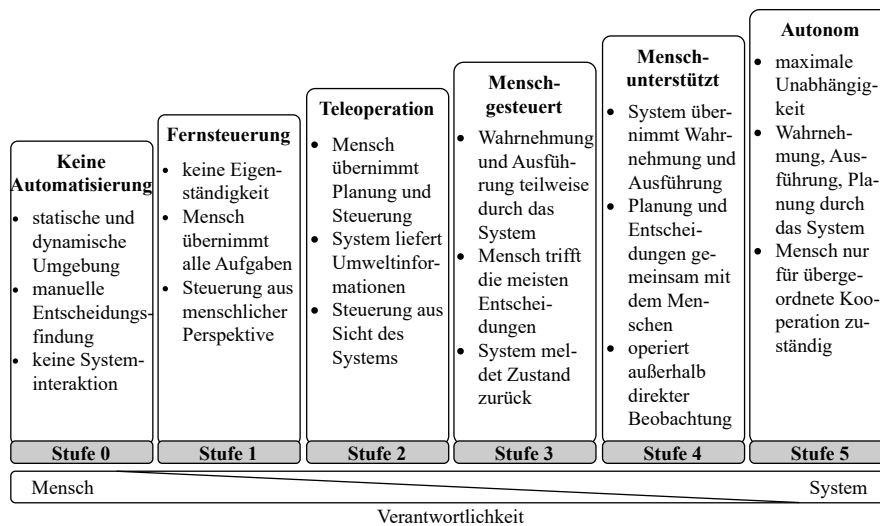


Abb. 2.11 Grade der Autonomie (vgl. [Hua+07, S. 7–8])

Das ALFUS-Modell sieht vor, die drei Dimensionen für ein autonomes System einzeln zu bewerten und aus dem erreichten Durchschnittswert die Einordnung in eine der Autonomiestufen in Abbildung 2.11 vorzunehmen [Hua+07, S. 7–8].

- *Stufe 0*: Auf Stufe 0 finden sich Systeme, die rein vom Menschen ohne technische Unterstützung betrieben werden.

- *Stufe 1:* Auf Stufe 1 werden ferngesteuerte Systeme eingestuft, bei denen ein technisches System als Hilfsmittel eingesetzt wird, jedoch alle Aufgaben von der Wahrnehmung bis zur Ausführung durch den Menschen übernommen werden, der das System aus seiner eigenen Perspektive steuert.
- *Stufe 2:* In Stufe 2 (Teleoperation) befindet sich das System außerhalb der direkten Sicht des Menschen, die Steuerung erfolgt daher aus der Perspektive des Systems. Wahrnehmung, Analyse, Planung und Entscheidung werden fast vollständig vom Menschen übernommen und das System führt Handlungen nach direkter Anweisung aus.
- *Stufe 3:* Bei den menschengesteuerten Systemen in Stufe 3 werden Wahrnehmung und Ausführung zwischen Mensch und System geteilt. Die Planung und die Entscheidungsfindung liegen überwiegend beim Menschen, während das System eigenständig Umgebungsinformationen liefert.
- *Stufe 4:* Systeme in Stufe 4 werden als menschenunterstützt eingestuft. Hier werden Wahrnehmung und Ausführung überwiegend durch das System übernommen, Analyse, Planung und Entscheidungsfindung erfolgen gemeinsam durch den Menschen und das technische System.
- *Stufe 5:* Auf Stufe 5 erreichen Systeme den höchsten Grad an Autonomie. Systeme auf dieser Stufe handeln mit maximaler Unabhängigkeit und Selbstbestimmung innerhalb ihrer technischen Grenzen. Die Wahrnehmung und die Ausführung erfolgen vollständig durch das System, die Planung und die Entscheidungsfindung erfolgen ebenfalls überwiegend systemseitig. Der Mensch ist nur in übergeordneten Aufgaben wie Kooperation und Verhandlung involviert.

Ausgehend von dem ALFUS-Modell wurden weitere Modelle zur Einstufung der Autonomie eines Systems entwickelt. Levels of Robot Autonomy (LORA) wurde von [BFR14] für den Kontext der Mensch-Roboter-Interaktion entwickelt. Es enthält zehn Autonomiestufen und bewertet die Autonomie in den primitiven Roboterfunktionen Wahrnehmen, Planen und Ausführen. Fottner et al. übertragen die Autonomielevel von ALFUS und LORA auf die Intralogistik und ergänzen die Dimension Selbstoptimierung als Merkmal der Autonomie [Fot+21].

Allen diesen Modellen ist gemein, dass der Autonomiegrad eines Systems umso höher eingestuft wird, je besser das System in der Lage ist, eigenständig Aufgaben zu übernehmen, die bisher dem Menschen vorbehalten waren oder es noch sind [Hua06, S. 48].

### 2.2.1.3 Autonome mobile Roboter

Eine autonome Technologie, die in den letzten Jahren zunehmend Einzug in Materialflusssysteme hält, sind AMR. Der Übergang von klassischen

Fahrerlosen Transportsystemen (FTS) zu AMR ist eine bedeutende Entwicklung im Bereich der innerbetrieblichen Materialflüsse. Während FTS auf eine zentralisierte Steuerung und fest definierte Fahrwege angewiesen sind, zeichnen sich AMR durch Hardware- und Softwarekomponenten aus, die eine dezentrale, autonome Navigation und generelle Entscheidungsfindung ermöglichen. Dadurch können sie sich flexibel in dynamischen und unstrukturierten Umgebungen bewegen, mit anderen Systemen kommunizieren und sich selbstständig an veränderte Bedingungen anpassen [Fra+21, S. 1–2].

Diese Eigenschaften erweitern den Einsatzbereich von AMR in Materialflusssystemen deutlich über klassische Transportaufgaben hinaus. Neben der Handhabung von Ladeeinheiten übernehmen sie zunehmend kollaborative Aufgaben im Zusammenspiel mit menschlichen Mitarbeitenden oder agieren als eigenständige Serviceeinheiten, zum Beispiel bei der Inventur (vgl. Unterabschnitt 2.2.2).

Im Folgenden wird die folgende Definition von AMR in Anlehnung an [Fra+21, S. 3] verwendet:

**Definition:** *Autonome Mobile Roboter*

AMR sind Roboter, die sich unabhängig von direkter menschlicher Steuerung in ihrer Umgebung bewegen, ihre Umgebung wahrnehmen, Entscheidungen treffen und mindestens in Teilaufgaben eigenständig auf Autonomiestufe 4 handeln. Je nach Einsatzkontext können sie mit anderen technischen Systemen oder Menschen interagieren. AMR dienen als Plattform für Materialtransport, kollaborative Aufgaben oder die Erbringung von Dienstleistungen innerhalb eines räumlich begrenzten Einsatzbereichs.

### 2.2.2 Anwendungsbereiche von AMR in Materialflusssystemen

Um in den generierten Lern- und Testumgebungen alle relevanten Einsatzbereiche von AMR innerhalb von Materialflusssystemen abzudecken, werden im folgenden Unterkapitel die typischen Prozesse und Aufgaben beschrieben, die diese Systeme übernehmen.

Die von AMR unterstützten Prozesse lassen sich in die Kategorien *Transport*, *Lagerung* und *Handhabung* einordnen. Die Handhabung schließt auch Kommissionierung und Verpackung ein ([Fot+21, S. 3], [Fra+21, S. 6]). Je nach Anwendungsbereich unterscheiden sich die Konfigurationen der Roboter (siehe Abbildung 2.12).

Der häufigste Einsatzbereich von AMR in der Intralogistik ist der *Transport* von Ladeeinheiten. Dabei werden Ladeeinheiten an einer Quellposition aufgenommen, unter autonomer Navigation zu einer Zielposition transportiert

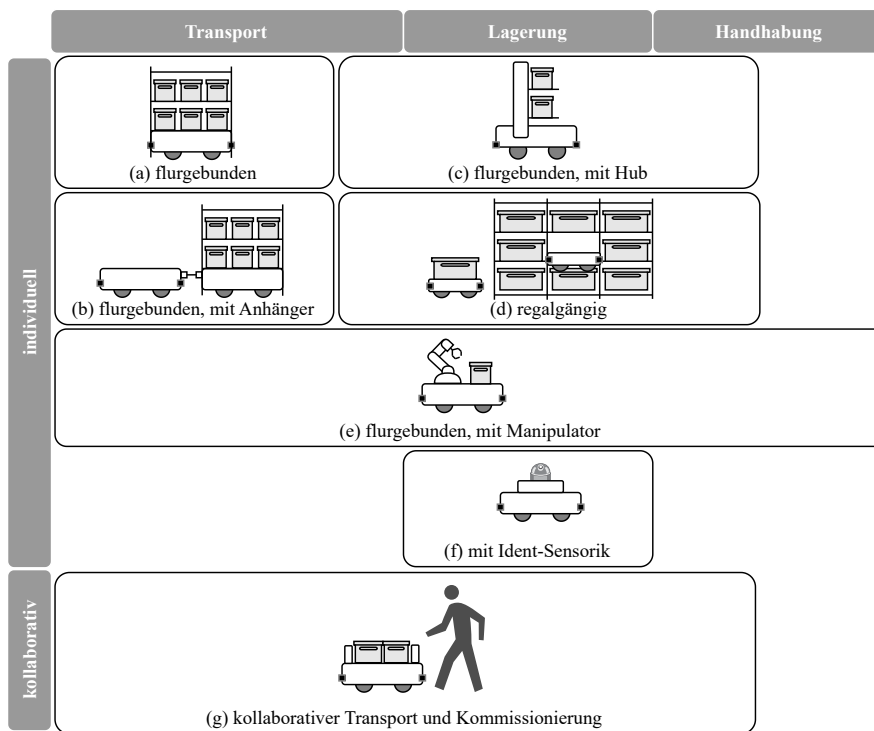


Abb. 2.12 AMR-Typen und ihre Fähigkeiten

und dort abgesetzt. Eine gängige Variante für diesen Einsatzbereich sind flurgebundene Roboter, die Ladeeinheiten durch Unterfahren aufnehmen (Abbildung 2.12 (a)). Diese Roboter nutzen spezielle Aufnahmemittel oder transportieren aufgeständerte Ladeeinheiten, zum Beispiel mobile Regale [EW11] oder Ladeeinheiten von Aufnahmestationen [Hom+20].

Eine weitere Möglichkeit für den Transport bieten flurgebundene Roboter mit Anhängern (Abbildung 2.12 (b)). In diesem Fall übernimmt der Roboter die aktive Bewegung, während der Anhänger als passives Trägerelement für zu transportierende Ladeeinheiten dient [Hom+20, S. 12]. Mehrere Anhänger können hintereinander gekoppelt werden, um die Transportkapazität zu erhöhen.

Einige AMR sind so konstruiert, dass sie direkt auf Lagerfächer zugreifen. Sie können so zusätzlich zum Transport die Prozesse der Einlagerung und der Kommissionierung im Bereich *Lagerung* unterstützen (Abbildung 2.12 (c)-(d)). Dies geschieht etwa durch Roboter mit Hubvorrichtungen für Paletten oder Kleinladungsträger ([Tsi+21], [Klo+23], [Ame23]) oder durch Fahrzeuge, die an der Regalkonstruktion hochklettern, um Lagerplätze zu erreichen [Bio23]. Bei der Einlagerung übernehmen die Roboter Ladeeinheiten an Übergabepunkten, fahren in die Regalgassen und positionieren die Einheiten

auf freien Lagerplätzen. Im Rahmen der Kommissionierung entnehmen sie Ladeeinheiten aus den Regalen und bringen sie zu Abgabepositionen. Dabei erfolgt der Zugriff auf ganze Ladeeinheiten, nicht auf Einzelartikel.

Anders ist dies bei Robotern mit integrierten Manipulatoren oder Greifern (Abbildung 2.12 (e)). Diese ermöglichen zusätzlich den Zugriff auf einzelne Artikel in Ladeeinheiten [Tha+18]. Sie können damit alle Prozesse der Kategorien *Transport*, *Lagerung* und *Handhabung* abdecken. Die Güter werden über den Manipulator gegriffen und direkt auf der Plattform oder einem Anhänger bewegt, eingelagert oder weiterverarbeitet.

Ein weiterer Anwendungsbereich von AMR in Materialflusssystemen liegt in der Unterstützung der Inventur (Abbildung 2.12 (f)). Hierfür sind die Roboter mit optischer oder funkbasierter Sensorik wie Kameras oder Radio-frequency Identification (RFID) ausgestattet. Sie bewegen sich durch das Lager und erfassen Informationen über den Lagerbestand [Cas+20]. Diese Systeme können als flurgebundene Fahrzeuge oder auch als Drohnen im dreidimensionalen Raum realisiert sein [FF20].

Darüber hinaus arbeiten AMR zunehmend kollaborativ mit Menschen zusammen (Abbildung 2.12 (g)). Ein typisches Beispiel ist die gemeinsame Kommissionierung: Der Roboter trägt einen Sammelbehälter und positioniert sich an der richtigen Stelle im Lager, während die Person den Artikel entnimmt und an den Roboter übergibt. Danach bewegen sich Roboter und Mensch unabhängig weiter. Ist die Bestellung vollständig, bringt der Roboter den Behälter zum Verpackungs- oder Konsolidierungsbereich, bevor er für den nächsten Auftrag eingesetzt wird [Fra+21, S. 2].

Die Einsatzbereiche von AMR in Materialflusssystemen erstrecken sich somit allein durch ihre Transportfunktion über sämtliche typischen Funktionsbereiche (siehe Unterunterabschnitt 2.1.2.1). Darüber hinaus führen Aufgaben in den Bereichen Handhabung und Service zu zusätzlichen Anforderungen an virtuelle Lern- und Testumgebungen: Neben der reinen statischen Abbildung von Elementen, etwa für die Navigation, muss auch die Interaktion mit Objekten in den generierten Lern- und Testumgebungen berücksichtigt werden.

### 2.2.3 Zu lösende Steuerungsentscheidungen in Materialflusssystemen

AMR können intralogistische Prozesse wie Transport, Lagerung und Handhabung entweder eigenständig oder in Zusammenarbeit mit Menschen unterstützen. Für den effizienten Betrieb von AMR in Materialflusssystemen müssen verschiedene Steuerungsentscheidungen getroffen werden (siehe Abbildung 2.13). In klassisch gesteuerten Materialflusssystemen werden diese Entscheidungen zentral getroffen. Je nach Steuerungsarchitektur und Autonomiegrad können diese Entscheidungen in autonomen, dezentral gesteuerten

Systemen durch die autonomen Entitäten selbst getroffen werden, sodass ein fließender Übergang zwischen der Steuerung des Materialflusssystem und der Steuerung der einzelnen Roboter entsteht. Im Folgenden werden die relevanten Steuerungsentscheidungen vorgestellt, um ihre Abdeckung in den generierten Lern- und Testumgebungen für zunehmend generalistische AMR in der Intralogistik sicherzustellen (siehe Kapitel 4).

Steuerungsebene	Transport	Lagerung	Handhabung	übergreifend
Materialfluss- steuerung          robotische Steuerung	<ul style="list-style-type: none"> <li>• Flottenmanagement</li> <li>• Leerfahrzeugmanagement</li> <li>• Scheduling</li> <li>• Dispatching</li> <li>• Routing</li> <li>• Navigation</li> <li>• Deadlock-Vermeidung</li> </ul>	<ul style="list-style-type: none"> <li>• Lagerplatzzuweisung</li> <li>• Zonung</li> <li>• Lagerplatzbelegung</li> <li>• Inventur</li> </ul>	<ul style="list-style-type: none"> <li>• Behälterpackproblem</li> <li>• Identifikation einer Einheit</li> <li>• Greifen</li> <li>• Bewegen einer Einheit</li> </ul>	<ul style="list-style-type: none"> <li>• Robustheit</li> <li>• Mensch-Maschine-Interaktion</li> <li>• Kollisionsvermeidung</li> </ul>

**Abb. 2.13** Steuerungsentscheidungen für AMR in Materialflusssystemen (vgl. [Fot+21, S. 3], [Fra+21, S. 6], [WHK22])

### Transport

Im Bereich Transport umfasst das *Flottenmanagement* die Entscheidungen über die benötigte Anzahl und Art der eingesetzten Roboter [Fra+21, S. 7].

Weitere relevante Steuerungsentscheidungen im Bereich Transport liegen im *Leerfahrzeugmanagement*. Hier muss die Frage beantwortet werden, an welcher Stelle im Materialflusssystem sich Roboter aufhalten sollen, solange sie keinen aktiven Transportauftrag haben. Eine vorausschauende Auswahl der Warteposition kann die Effizienz künftiger Transportprozesse verbessern [Fot+21, S. 13].

Um die im Materialflusssystem anstehenden Transportaufträge auf die zur Verfügung stehenden AMR zu verteilen, müssen Entscheidungen zum *Scheduling* und *Dispatching* getroffen werden. *Scheduling* plant die zeitliche Reihenfolge der Transportaufträge, während *Dispatching* die konkrete Zuweisung der Roboter zu diesen Aufträgen übernimmt ([Fra+21, S. 10 f.], [Fot+21, S. 12]). Sind die Aufträge zugewiesen, müssen die Roboter das *Routing* (die Pfadplanung) und die *Navigation* (die Pfadverfolgung) eigenständig umsetzen [Fot+21, S. 13]. Im Gegensatz zu festgelegten Routen in FTS bewegen sich autonome Roboter frei in der Fläche und nutzen Sensoren zur Lokalisierung sowie zur Hinderniserkennung und zur Anpassung ihrer Bewegungsgeschwindigkeit [Fra+21, S. 2].

Ein weiteres Steuerungsproblem ist die *Deadlock-Vermeidung*. Diese Blockaden entstehen, wenn sich Roboter gegenseitig den Weg versperren und jeweils darauf warten, dass ein anderer Roboter eine Ressource freigibt. Die

autonomen Systeme müssen Strategien entwickeln, um solche Situationen erkennen und auflösen zu können [Fot+21, S. 14].

### **Lagerung**

Im Bereich der Lagerung müssen Entscheidungen zur Einlagerung und Auslagerung von Ladeeinheiten getroffen werden. Wesentliche Teilprobleme sind die Lagerplatzzuweisung, die Zonierung und die konkrete Lagerplatzbelegung. Sie bestimmen gemeinsam, auf welchem Platz eine bestimmte Ladeeinheit abgelegt oder entnommen wird [Fot+21, S. 11].

Die *Lagerplatzzuweisung* umfasst die Auswahl der Art und der Anzahl geeigneter Lagerplätze auf Basis von Eigenschaften wie Abmessung oder Gewicht. Die *Zonierung* strukturiert das Lager, zum Beispiel durch Gruppierung nach Zugriffshäufigkeit. Bei der *Lagerplatzbelegung* steht dann die konkrete Entscheidung der Zuordnung einer bestimmten Ladeeinheit zu einem bestimmten Lagerplatz an, auf den die Ladeeinheit eingelagert wird [Fot+21, S. 15 f.].

In die Kategorie Lagerung fällt ebenfalls die Aufgabe der *Inventur*. Die Probleme, die AMR bei der Inventur lösen müssen, bestehen vor allem in ihrer eigenen Lokalisierung sowie der Lokalisierung, Identifikation und Zählung der erkannten Ware, der Kommunikation der Ergebnisse und der Planung der Abdeckung des Lagerbereichs [Cas+20].

### **Handhabung**

Die AMR-spezifischen Aufgaben im Bereich der Kommissionierung, Verpackung und Handhabung ähneln sich, sodass sie hier unter dem Begriff Handhabung zusammengefasst sind. Für alle drei Aufgaben muss zunächst die Einheit identifiziert werden, die aufgenommen werden soll (einzelner Artikel, Paket auf einer Palette, eine Palette etc.). Anschließend muss der AMR entscheiden, wie die Einheit gegriffen und von ihrem Lagerort genommen werden kann. Es folgt eine kurze Bewegung der Einheit zur bestimmten Zielposition, zum Beispiel dem Kommissionierwagen oder einer Verpackungseinheit, wo sie abgelegt wird. Um eine gute Zielposition für die gegriffene Einheit zu bestimmen, muss der AMR das Behälterpackproblem lösen, das sich damit beschäftigt, wie Objekte verschiedener Größen in Zielbehältern positioniert werden können, sodass möglichst wenige Behälter benötigt werden. [Fot+21, S. 16]

### **Übergreifend**

Neben den spezifischen Teilproblemen zur Durchführung der grundlegenden Prozesse in Materialflusssystemen gibt es übergreifende Aufgaben, die die AMR im Laufe des Betriebs beachten und lösen müssen. Dazu gehört beispielsweise das Aufrechterhalten eines *robusten Systems*, eines der entscheidenden Vorteile von autonomen Systemen gegenüber zentral gesteuerten

Systemen. Die autonomen Systeme müssen Strategien entwickeln, um im Falle eines Fehlers während der Lösung oder Ausführung einer ihrer Teilaufgaben in der Lage zu sein, diesen zu erkennen und das normale Systemverhalten wiederherzustellen, ohne dass Menschen diesen Prozess überwachen oder eingreifen müssen. [Fra+21, S. 13]

In hybriden, kollaborativen Systemen kommen zudem noch generelle Aufgaben im Bereich der *Mensch-Maschine-Interaktion* hinzu. Hier müssen die AMR Kommunikationsaufgaben dadurch lösen, dass sie optische Signale senden, verbale Befehle verstehen und selbst über Worte kommunizieren, oder Gesten deuten und entsprechend agieren können. [Fot+21, S. 9]

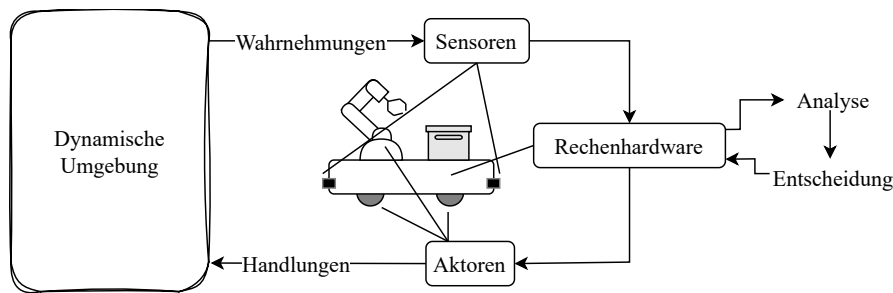
Ebenso ist die *Kollisionsvermeidung* ein Teilproblem, das in allen Kategorien der Aufgaben im Materialflusssystem eine Rolle spielt und das Risiko eines Zusammenstoßes mit anderen statischen oder dynamischen Objekten verringern soll. Diese Objekte können selbst mobil und autonom gesteuert sein, sodass die Kollisionsvermeidung häufig mit der Einschätzung der Intention und Bewegungsrichtung anderer Objekte einhergeht [WHK22, S. 6].

Die Autoren Fragapane et al. sowie Wesselhöft et al. untersuchen eine Vielzahl von Studien, die sich mit unterschiedlichen Steuerungs- und Planungsentscheidungen für AMR befassen [Fra+21; WHK22]. Sie identifizieren dabei den Trend, dass vermehrt mehrere der Entscheidungsprobleme von AMR integriert betrachtet werden, was einen Wandel von der sequenziellen Lösung einzelner Teilprobleme hin zu ganzheitlichen Steuerungsansätzen erkennen lässt. Methoden der KI spielen dabei eine zentrale Rolle, da sie in der Lage sind, komplexe Abhängigkeiten zwischen den einzelnen Entscheidungsprozessen zu erfassen und integriert zu optimieren. Diese Entwicklungen deuten auf einen Trend hin, Steuerungsentscheidungen nicht mehr isoliert, sondern als Teil eines lernfähigen Gesamtsystems zu verstehen.

Insgesamt lässt sich für die Lern- und Testumgebungen für AMR im Digitalen Kontinuum ableiten, dass in ihnen sowohl die Elemente als auch das Verhalten von Materialflusssystemen so abgebildet werden müssen, dass sie die Grundlage zur Lösung der Steuerungsaufgaben in Abbildung 2.13 bilden können.

#### **2.2.4 Komponenten und robotisches Steuerungskonzept von AMR**

Um die in virtuellen Lern- und Testumgebungen bestimmten Steuerungsstrategien von AMR auf physische Systeme übertragen zu können, müssen die physischen Steuerungsmechanismen in den virtuellen Umgebungen realitätsnah abgebildet werden. Um die Anforderungen an Lern- und Testumgebungen für KI-basierte AMR im Digitalen Kontinuum dahingehend ableiten zu können, werden im Folgenden der physische Aufbau und das robotische Steuerungskonzept heutiger AMR erläutert.



**Abb. 2.14** Komponenten und Steuerungskonzept eines autonomen Systems am Beispiel eines AMR ([BK23, S. 36], [Sie11, S. 10])

Die Grundkomponenten, die autonome Systeme generell besitzen, sind Sensorik zur Wahrnehmung der Umgebung, Aktorik zur Ausführung von Handlungen und eine Recheneinheit zur Verarbeitung von Informationen [Wah17, S. 411]. Bei AMR sind diese Komponenten als physische Hardware auf einer mobilen Plattform integriert (siehe Abbildung 2.14). In Anwendungen, bei denen AMR mit anderen autonomen Systemen oder mit Menschen zusammenarbeiten, wird diese Infrastruktur zusätzlich durch Kommunikationsschnittstellen ergänzt.

Das Ziel eines autonomen Systems besteht darin, Aufgaben vom Menschen unabhängig auszuführen. Dazu erfasst es mithilfe von *Sensoren* kontinuierlich Informationen über seine Umgebung (externer Zustand) und über den eigenen Systemzustand (interner Zustand).

Diese Daten werden von der *Recheneinheit* des eingebetteten Computersystems verarbeitet. Eingebettete Rechenhardware bildet dabei die technische Grundlage der AMR, da sie die notwendige Rechenleistung bereitstellt, um Wahrnehmung, Planung und Steuerung in Echtzeit zu ermöglichen. Ihre Aufgabe besteht in der Analyse des aktuellen Systemzustands basierend auf dem Sensorinput und der Auswahl passender Handlungen auf dieser Grundlage. AMR nutzen hierzu nicht nur klassische Analyse- und Regelungstechniken, sondern integrieren zunehmend modellbasierte und datengetriebene Verfahren zur Optimierung ihrer Bewegungs- und Entscheidungsstrategien, insbesondere Methoden der KI (siehe Abschnitt 2.3).

Die geplanten nächsten Handlungen führt das autonome System anschließend durch *Aktoren* aus, zum Beispiel das Aktivieren eines Motors zur Fortbewegung entlang eines geplanten Pfads.

Für die Gestaltung der Lern- und Testumgebungen im Digitalen Kontinuum ist es entscheidend, diese zentralen Funktionsprinzipien realitätsnah abzubilden. Die virtuelle Umgebung muss sowohl die sensorische Wahrnehmung als auch die Entscheidungs- und Handlungsmechanismen eines AMR realitätsnah simulieren können. Nur so kann sichergestellt werden,

dass in der virtuellen Umgebung erlernte Verhaltensweisen auch auf reale, physische Systeme übertragbar sind.

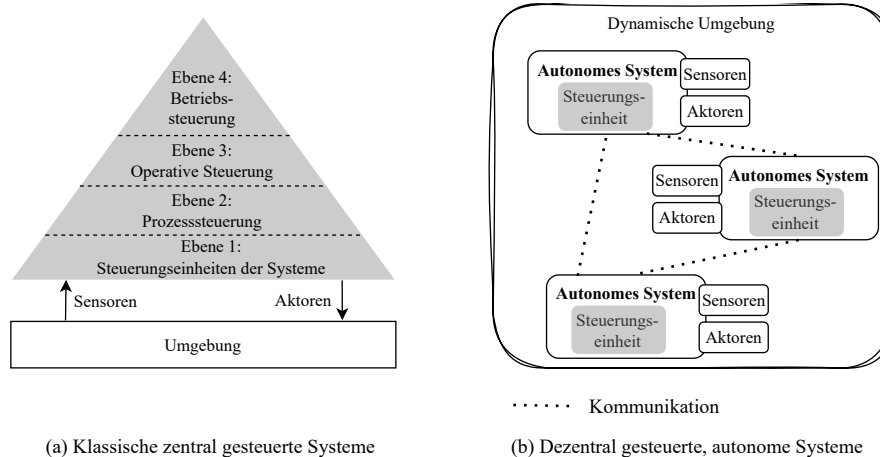
Die in Tabelle 2.1 aufgelisteten Basistechnologien geben einen Überblick über typische Technologien, die für die Grundkomponenten von AMR eingesetzt werden.

**Tabelle 2.1** Basistechnologien für AMR [Fot+21, S. 10 f.]

<b>Sensorik</b>	<b>Kategorie</b>	<b>Beispiele</b>
Identifikation	optisch	Barcode, Kamera
	funkbasiert	RFID, NFC
Lokalisierung	markerbasiert	induktive, magnetische oder optische Sensoren
	Orientierungspunkte der Umgebung	Ultraschall-, Kamera- oder Lasersensoren
	funkbasiert	GPS, Mobilfunk, WLAN, Bluetooth, UWB, RFID, LPWAN, LoRaWAN, NB-IoT
Betriebssicherheit	intern	Odometer, Beschleunigungsmesser, Trägheitssensoren, Inkrementalgeber
	taktil	Berührungssensoren, Stoßdämpfer
	optisch	Lichtschranken, Infrarot, Laser, Kamera
Zustand	andere	Induktionsschleifen, kapazitive Sensoren
	Zustandssensor	Last-, Kraft-, Drehmoment-, Schwingungs- und Energieverbrauchssensoren
<b>Rechenhardware</b>	<b>Kategorie</b>	<b>Beispiele</b>
Informationsverarbeitung	Berechnung	SPS, Mikrocontroller, Industrie-PCs
<b>Aktorik</b>	<b>Kategorie</b>	<b>Beispiele</b>
Bewegung und Handhabung	elektrische Antriebe	rotierende/lineare elektrische Maschinen, Synchronmaschinen, Magnetlager, Zugmagnete, Linearmotoren
	Fluidtechnik	Pneumatik, Hydraulik und Vakuumtechnik
	Mechanik	mechanische Manipulatoren
<b>Kommunikation</b>	<b>Kategorie</b>	<b>Beispiele</b>
Maschine–Maschine	Feldbus	PROFIBUS, CAN, ASI, InterBus-S
	Industrial Ethernet	Profinet, Ethernet/IP, EtherCAT, Powerlink
Mensch–Maschine	funkbasiert	Bluetooth, WLAN, ZigBee, SigFox, LPWAN, LoRaWAN, NB-IoT, Mobilfunk
	optisch	Displays, Projektoren, Lichter
	Sprache	Lautsprecher, Mikrofone
	Gesten	Kameras

## 2.2.5 Steuerungsarchitektur von AMR in Materialflusssystemen

Auch die übergeordnete Steuerungsarchitektur von AMR ist ein Aspekt, der in den generierten Lern- und Testumgebungen Beachtung finden muss, um die Ergebnisse aus den digitalen Umgebungen auf die physische Realität zu übertragen. Mit dem Wandel von automatisierten hin zu autonomen Systemen vollzieht sich auch ein Paradigmenwechsel in der Steuerungsarchitektur von klassisch zentral und hierarchisch gesteuerten Materialflusssystemen hin zu dezentral gesteuerten, autonomen Systemen (siehe Abbildung 2.15).



(a) Klassische zentral gesteuerte Systeme

(b) Dezentral gesteuerte, autonome Systeme

**Abb. 2.15** Paradigmenwechsel von zentral gesteuerten zu dezentral gesteuerten, autonomen Materialflusssystemen (i. a. A. [Hün19, S. 5] und [Fot+21, S. 3])

Materialflusssysteme werden klassischerweise durch zentral organisierte, hierarchisch aufgebaute Steuerungssysteme koordiniert (siehe Abbildung 2.15 (a)). Die dabei zu treffenden Steuerungsentscheidungen sind nach Entscheidungsebenen strukturiert. Je höher die Ebene, desto stärker ist die Aggregation der zugrunde liegenden Informationen und desto strategischer ist der Entscheidungscharakter.

Auf der untersten Ebene führen die physischen Systeme ihre Aufgaben mithilfe von Aktoren aus und erfassen durch Sensoren Informationen über ihre Umgebung. Diese Informationen werden an die nächsthöhere Ebene übergeben, auf der die Informationen aller physischen Systeme zusammenlaufen und zentral Entscheidungen zur Prozesssteuerung getroffen werden, zum Beispiel zur Routenwahl. Die folgende Ebene nutzt diese Entscheidungen als Grundlage für die operative Steuerung, etwa für Scheduling und Dispatching. Auf der obersten Ebene, der Betriebssteuerung, werden schließlich langfristige Entscheidungen geplant, beispielsweise zur Kapazitätsplanung oder zum Personaleinsatz. [Fot+21, S. 3]

Im Vergleich dazu agieren autonome Systeme eigenständiger und stärker lokalisiert. Mindestens ein Teil der Steuerungsentscheidungen wird dabei dezentral von den autonomen Akteuren innerhalb des Materialflusssystems getroffen (siehe Abbildung 2.15 (b)). Diese Systeme nehmen über ihre Sensorik ihre unmittelbare Umgebung wahr und treffen auf Grundlage dieser begrenzten Information eigenständig Entscheidungen über ihre nächsten Handlungen. Die klassischen Ebenen der Steuerung verschwimmen dabei zunehmend zu integrierten, ganzheitlichen Entscheidungen, bei denen mehrere Aufgaben gleichzeitig betrachtet und gelöst werden [Fra+21, S. 18].

Grundsätzlich unterscheiden sich zentrale und dezentrale Steuerungsarchitekturen nicht in den Arten der zu lösenden Aufgaben zur Steuerung des Gesamtsystems (vgl. Unterabschnitt 2.2.3), sondern darin, wo in der Rechenarchitektur diese Entscheidungen getroffen werden und auf welcher Informationsbasis sie beruhen.

In dezentralen Architekturen wird das komplexe Steuerungsproblem in viele kleinere Teilprobleme zerlegt, die jeweils lokal von autonomen Einheiten bearbeitet werden. Diese Einheiten treffen Entscheidungen auf Grundlage der ihnen individuell zur Verfügung stehenden Informationen. Auf diese Weise wird die Komplexität einzelner Entscheidungsprozesse reduziert, was eine höhere Reaktionsfähigkeit des Gesamtsystems auf dynamische Veränderungen im Betrieb ermöglicht.

Diese lokale Entkopplung hat jedoch zur Folge, dass sich das Verhalten des Gesamtsystems schwerer vorhersagen und steuern lässt. Mit zunehmender Anzahl von Entitäten im System wächst der Lösungsraum, was die Vorhersage globaler Systemeffekte erschwert, zum Beispiel bei Kreuzungsverkehren oder Ressourcenkonflikten ([HK15, S. 179], [Fra+21, S. 7]). Eine analytische Modellierung solcher Systeme ist kaum mehr möglich. Daher kommt der Simulation eine zentrale Rolle zu: Sie ist ein entscheidendes Werkzeug, um die Wirkzusammenhänge verteilter Entscheidungsprozesse abzubilden und das Gesamtsystemverhalten zu bewerten, insbesondere im Kontext der in dieser Arbeit entwickelten Lern- und Testumgebungen für AMR (siehe Unterunterabschnitt 2.3.2.3).

Gleichzeitig sinkt durch die dezentrale Entscheidungsfindung tendenziell die Wahrscheinlichkeit, globale Optima zu erreichen. In bestimmten Fällen kann es daher notwendig sein, einzelne komplexe Entscheidungen an leistungsfähigere zentrale Recheneinheiten wie Server oder Cloud-Plattformen auszulagern.

Tabelle 2.2 fasst den Vergleich der Steuerungsarchitekturen zusammen. Die Frage, welche Steuerungsarchitektur unter welchen Bedingungen geeignet ist, stellt ein eigenständiges Forschungsfeld dar. Die im Rahmen dieser Arbeit entwickelten Lern- und Testumgebungen sollen daher so konzipiert sein, dass sie unterschiedliche Steuerungsansätze realitätsnah abbilden und

**Tabelle 2.2** Vergleich zentraler und dezentraler Steuerungsarchitekturen in Materialflusssystemen

Merkmale	Zentral gesteuerte Systeme	Dezentral gesteuerte Systeme
<b>Entscheidungsebene</b>	Entscheidungsfindung auf höheren Ebenen und Weitergabe an untergeordnete Ebenen	Entscheidungsfindung lokal durch autonome Einheiten auf Basis ihrer eigenen Wahrnehmung
<b>Informationsfluss</b>	Vertikal strukturiert: Informationen werden von unten nach oben aggregiert und zentral ausgewertet.	Horizontal verteilt: Informationsaustausch erfolgt zwischen autonomen Einheiten auf lokaler Ebene.
<b>Reaktionsfähigkeit</b>	geringere Reaktionsgeschwindigkeit auf unerwartete Ereignisse durch zentrale Entscheidungs- und Kommunikationswege	hohe Reaktionsfähigkeit durch un-mittelbare, lokale Entscheidungsfindung
<b>Komplexität der Entscheidungsfindung</b>	hohe Komplexität zentraler Entscheidungen durch umfangreiche Informationslage	geringere Komplexität einzelner lokaler Entscheidungen, aber komplexes Gesamtsystemverhalten durch emergente Effekte
<b>Robustheit gegenüber Störungen</b>	anfällig gegenüber Ausfällen zentraler Komponenten	hohe Robustheit durch verteilte Entscheidungslogik

analysierbar machen, von vollständig zentral bis zu weitgehend dezentral organisiert.

Im Kontext des Einsatzes von KI-Methoden für die Steuerung von AMR, die auf leistungsfähige, zentrale Recheneinheiten für das Erlernen von Steuerungsstrategien angewiesen sind, stehen AMR und ihre Steuerungsarchitekturen in engem Zusammenhang mit dem Konzept der Cyberphysischen Systeme (CPS) und Digitalen Zwillinge, die in den folgenden Abschnitten erläutert werden.

### 2.2.5.1 Cyberphysische Systeme

Cyberphysische Systeme (CPS) sind ein grundlegender Bestandteil der Industrie 4.0, in der intelligente Maschinen und Objekte sowohl miteinander als auch mit Menschen kommunizieren und interagieren [BK23, S. 32 ff.]. Laut acatech [Aca11, S. 13] sind CPS solche, die

- mit eingebetteter Software ausgestattet sind,
- über Sensoren Daten des physischen Systems erfassen und durch Aktoren auf physische Vorgänge einwirken,
- erfasste Daten auswerten und speichern,
- aktiv oder reaktiv mit der physischen und der digitalen Welt interagieren,
- über digitale Kommunikationseinrichtungen untereinander und in globalen Netzen verbunden sind,
- weltweit verfügbare Daten und Dienste nutzen und
- über Mensch-Maschine-Schnittstellen verfügen.

Im Grunde bezeichnen CPS somit eingebettete Systeme, die mit dem Internet verbunden sind und auf diesem Weg die physische mit der digitalen Welt verknüpfen ([Alu15, S. 1], [DWF18, S. 17]).

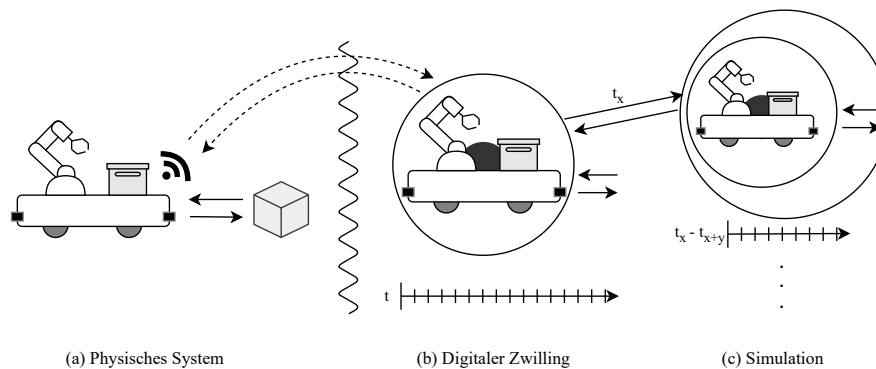
Viele der Eigenschaften von CPS werden durch AMR erfüllt (siehe Unterabschnitt 2.2.4). Erweitert wird die Definition durch den Fokus auf die Vernetzung der Systeme mit der digitalen Welt über globale Netzwerke wie das Internet. Durch diese Verbindung können CPS ihre selbst erzeugten Daten durch weitere externe Daten anreichern und für rechenintensive Dienste Kapazitäten in angebundenen Rechenzentren nutzen, für die die eingebettete Hardware nicht genügend Kapazitäten aufweist.

Die digitalen Informationen eines physischen CPS bilden eine digitale Repräsentation des realen Objekts. CPS sind daher eng mit dem Konzept Digitaler Zwillinge verknüpft, die es ermöglichen, Informationen des physischen Objekts, die in der digitalen Welt bereitgestellt werden, abzubilden, zu überwachen oder zu optimieren.

### 2.2.5.2 Digitale Zwillinge

Das Konzept Digitaler Zwillinge stammt aus dem Bereich des Product Lifecycle Managements und wurde Anfang der 2000er-Jahre von Grieves und Vickers entwickelt ([Gri15], [Gri16]). Es basiert auf der Idee, dass zu einem physischen System ein digitales Gegenstück, ein *Zwilling*, geschaffen wird, das dessen relevante Informationen über den gesamten Lebenszyklus hinweg abbildet und mit ihm kontinuierlich verknüpft bleibt [GV17, S. 92].

Drei grundlegende Elemente sind Teil des Konzeptes Digitaler Zwillinge: das physische Objekt, das digitale Objekt und die Kommunikation zwischen diesen Objekten (siehe Abbildung 2.16 (a) und (b)).



**Abb. 2.16** Grundstruktur eines Digitalen Zwillinges (i. A. a. [RM22, S. 202 ff.])

Das digitale Objekt wird als Modell des physischen CPS geschaffen und weist die Merkmale eines Modells nach der allgemeinen Modelltheorie von Stachowiak auf (siehe Abschnitt 2.4). Das digitale Objekt erfasst daher nicht

alle Eigenschaften des Originalobjekts, sondern beschränkt sich auf die relevanten Eigenschaften, die durch ein Metamodell spezifiziert werden [RM22, S. 198]. Dabei ist die Hauptaufgabe des digitalen Zwillinges, die Zwillingseigenschaft zwischen physischem und digitalem Objekt durch den automatisierten Abgleich des Zustands des physischen Objektes mit seinem digitalen Abbild permanent und echtzeitnah aufrechtzuerhalten [Roi22, S. 23].

Ein Digitaler Zwilling besteht somit aus zwei Recheneinheiten: Eine Einheit ist im physischen CPS eingebettet. Dieses besitzt Sensoren zur Wahrnehmung der Umgebung und die Fähigkeit zur Kommunikation. Die zweite Recheneinheit führt das virtuelle Gegenstück als digitales Abbild des realen Objekts aus. Die eingebettete Recheneinheit wird als ressourcenbeschränkt angenommen, während die zweite Recheneinheit theoretisch auf uneingeschränkte Ressourcen zugreifen kann [Roi22, S. 24].

Ein zentraler Mehrwert Digitaler Zwillinge ist es, auf Basis ihrer echtzeitnahen Daten Experimente zu Veränderungen im physischen System in der digitalen Welt schneller, risikofreier und ohne Eingriff in den laufenden Betrieb durchzuführen. Dazu können Digitale Zwillinge als Basis für Simulationsstudien dienen (siehe Abbildung 2.16 (b) und (c)) [Sha+10, S. 7]. Die im Digitalen erlangten Erkenntnisse können dann in das physische System übertragen werden. Da das digitale Modell ein Abbild der relevanten Aspekte des physischen Systems ist, besteht die Erwartung, dass die gleichen positiven Veränderungen, die in der digitalen Welt erreicht werden, auch in der Realität eintreten. Das gilt für existierende physische Systeme wie für digitale Prototypen noch nicht physisch umgesetzter Systeme [GV17, S. 102].

Die Entwicklung und der Betrieb von CPS und damit von AMR werden durch den Einsatz von Digitalen Zwillingen beschleunigt und erleichtert. Durch die Abbildung der relevanten Eigenschaften des physischen Objekts in der digitalen Welt können Anpassungen und Reaktionen auf prognostizierte Systemzustände parallelisiert, beschleunigt und sicher ausgetestet und bewertet werden, bevor sie in die Realität übertragen werden. Dadurch sind Digitale Zwillinge ein zentraler Aspekt der Motivation für die automatisierte Generierung von virtuellen Lern- und Testumgebungen, um die Entwicklung und den Betrieb von autonomen Systemen zu unterstützen (siehe Kapitel 3).

### 2.3 Agentenbasierte Künstliche Intelligenz

Aus den Ausführungen zu autonomen Systemen (Abschnitt 2.2) geht hervor, dass diese Systeme eigenständig Zielvorgaben erreichen, indem sie ihre lokale Umgebung wahrnehmen, diese Informationen teils auf eingebetteten Recheneinheiten verarbeiten, um Entscheidungen zu treffen, Handlungen zu planen, diese Handlungen anschließend auszuführen sowie in bestimmten Anwendungsbereichen mit anderen Systemen und Menschen kommuni-

zieren. Dabei werden sie als umso autonomer eingestuft, je komplexer die Aufgabe ist, die sie bearbeiten, je unstrukturierter, dynamischer und unsicherer die Umgebung ist, in der sie agieren, und je unabhängiger vom Menschen sie arbeiten.

Um die Anforderungen hoher Stufen der Autonomie zu erfüllen, müssen die autonomen Einheiten also mit einer hohen Reaktionsgeschwindigkeit komplexe Zusammenhänge abbilden, die ihre wahrgenommenen Umgebungsinformationen auf sinnvolle, durch sie auszuführende Handlungen abbilden. Dies erfordert den Einsatz probabilistischer, evidenzbasierter und entscheidungstheoretischer Methoden zur Wissensrepräsentation, zum Lernen und zur Inferenz. Diese Methoden müssen zudem mit verschiedenen Ressourcenbeschränkungen umgehen können, etwa in Bezug auf Zeit, Speicher, Rechenleistung oder Energie [Wah17, S. 410].

Klassische Steuerungssysteme, die auf festen Regeln und zentralen Entscheidungslogiken beruhen, stoßen hierbei an ihre Grenzen. In der Grundlagenforschung der KI wurden in den vergangenen Jahren jedoch bedeutende Fortschritte erzielt, die in diesen Bereichen Erfolge zeigen, beispielsweise im Bereich des Deep Learnings mit tiefen Neuronalen Netzen [Wah17, S. 412].

KI gilt daher als wesentliche Voraussetzung für die Realisierung hochautonomer Systeme. Hierzu zählen einerseits physische Systeme wie AMR, die im Kontext dieser Arbeit betrachtet werden und für die in dieser Arbeit automatisiert Lern- und Testumgebungen generiert werden. Andererseits zählen auch autonome Softwareagenten dazu, die den Entwicklungs- und Betriebsprozess solcher Systeme unterstützen (siehe Kapitel 3).

Um eine fundierte Grundlage für die Entwicklung geeigneter Entwicklungs- und Betriebsprozesse sowie Lern- und Testumgebungen für KI-basierte autonome Systeme zu schaffen, werden im weiteren Verlauf dieses Abschnitts die relevanten Begriffe aus dem Bereich der KI definiert und zentrale Lernkonzepte im Zusammenhang mit dieser Arbeit erläutert.

### 2.3.1 Begriffsabgrenzungen

Das Forschungsgebiet der KI besteht seit den 1950er-Jahren [Str21]. Es verfolgt das Ziel, Computersysteme zu entwickeln, die in der Lage sind, Fähigkeiten auf menschlichem Niveau oder darüber hinaus zu demonstrieren. Trotz der langen Historie existiert keine einheitlich anerkannte Definition für KI. Für die vorliegende Arbeit wird eine Definition verwendet, die sich an der Formulierung der *Strategie Künstliche Intelligenz* der Bundesregierung orientiert [Bun18, S. 4]:

**Definition:** *Künstliche Intelligenz*

Künstliche Intelligenz bezeichnet ein Set an Methoden und Algorithmen aus Mathematik und Informatik, die das Ziel verfolgen, Computersysteme durch selbstoptimierende Verfahren mit den gleichen intellektuellen Fähigkeiten auszustatten, wie der Mensch sie hat, oder diese zu übertreffen.

Der Aspekt der Selbstoptimierung ist verbunden mit der Fähigkeit des Lernens. Das Maschinelle Lernen (ML) ist der Teilbereich der KI, der in den letzten Jahren enorme Fortschritte gemacht hat und neue Anwendungen ermöglicht, die bisher nicht erreichbar waren. Beim ML werden Algorithmen auf der Basis von Daten und Erfahrungen trainiert, aus denen sie die Approximation komplexer Zusammenhänge lernen, sich stets verbessern und an Änderungen anpassen [Moc21, S. 134]. Als datenbasierter Ansatz steht ML im Kontrast zu explizit programmierten Anwendungen, die auf festen Regeln beruhen. Der Einsatz von Methoden des ML eignet sich insbesondere in Situationen, in denen die Algorithmen mit Unsicherheiten umgehen und sich dynamisch an Änderungen anpassen müssen, die Zusammenhänge, die abgebildet werden, komplex sind und Daten vorliegen, aus denen die Zusammenhänge geschlossen werden können. Daher bilden die Methoden des ML eine geeignete Grundlage für die Realisierung autonomer Systeme.

**Definition:** *Maschinelles Lernen*

Maschinelles Lernen ist ein Teilbereich der KI, der sich darauf konzentriert, Computersysteme zu befähigen aus Daten zu lernen und sich zu verbessern, ohne dass sie explizit dafür programmiert werden. Es ermöglicht Systemen, Muster und Zusammenhänge in Daten zu erkennen und basierend auf diesen Erkenntnissen Vorhersagen oder Entscheidungen zu treffen.

Ähnlich wie in der Intralogistik ein Wandel von automatisierten hin zu zunehmend autonomen Systemen zu beobachten ist (siehe Abschnitt 2.2), vollzieht sich auch im Bereich der KI eine Entwicklung hin zu autonomen KI-Systemen. Besonders deutlich wird dieser Fortschritt im Feld der KI-Agenten. Aufbauend auf den Konzepten verteilter KI wurde das Agentenparadigma als Alternative zu zentralisierten, monolithischen KI-Systemen eingeführt.

Der agentenbasierte Ansatz basiert auf dem Prinzip autonom agierender, handlungsfähiger Einheiten, sogenannter Agenten, die ihre Umgebung wahrnehmen, Entscheidungen treffen und selbstständig Ziele verfolgen ([RND10, S. 34], [Liu+25, S. 16]). Damit entspricht das Konzept im Wesentlichen der

Definition autonomer Systeme, erweitert um die Spezifizierung, dass in der Steuerungslogik der Agenten explizit Methoden der KI eingesetzt werden.

**Definition:** *KI-Agent*

Ein KI-Agent ist eine handlungsfähige Softwareeinheit, die mithilfe von Methoden der KI, insbesondere des ML, befähigt wird, ihre Umgebung wahrzunehmen, eigenständig Entscheidungen zu treffen, Handlungen auszuführen und dabei definierte Ziele zu verfolgen. KI-Agenten bilden die Grundlage KI-basierter autonomer Systeme und können in unterschiedlichen Ausprägungen auftreten. Physisch eingebettete Agenten werden direkt auf robotischen Systemen ausgeführt und steuern dort physische Prozesse. Softwarebasierte KI-Agenten hingegen operieren auf verteilten oder zentralen Recheneinheiten und agieren innerhalb digitaler Umgebungen. Sie sind in der Lage, digitale Werkzeuge zu parametrisieren und auszuführen, Anwendungen zu steuern und dadurch eigenständig Entwicklungs-, Analyse- und Optimierungsaufgaben zu übernehmen.

Ein wesentlicher Treiber für die aktuellen Fortschritte im Bereich autonomer KI-Agenten ist die generative KI.

**Definition:** *Generative KI*

Generative KI bezeichnet KI-Modelle, die auf Basis gelernter Wahrscheinlichkeitsverteilungen neue Inhalte erzeugen können, beispielsweise Texte, Bilder, Audio, Videos oder Code. Die erzeugten Inhalte sind dabei statistisch konsistent mit den Daten, auf denen das Modell trainiert wurde.

Insbesondere große Sprach- und Multimodalmodelle, die auf sehr großen Datenmengen trainiert wurden, ermöglichen neue Anwendungen im Bereich der generativen KI. Sie ermöglichen unter anderem neuartige Fähigkeiten im Bereich des Reasonings und statten technische Systeme so mit der Fähigkeit des logischen Schlussfolgerns aus [FTD25].

**Definition:** *Reasoning*

Reasoning ist die Fähigkeit, aus vorhandenem Wissen oder gegebenen Informationen logische Schlussfolgerungen zu ziehen, Zusammenhänge zu erkennen und Probleme zu lösen. Es geht dabei nicht nur um das Wiedergeben gelernter Muster, sondern um die Fähigkeit zur eigenständigen Argumentation und Entscheidungsfindung.

Die Fähigkeiten des Reasonings ermöglichen es KI-Modellen, komplexe Aufgaben zu bearbeiten, die über reine Mustererkennung hinausgehen. Damit eröffnen sie neue Potenziale für die Planung, Problemlösung und Entscheidungsfindung in autonomen Systemen und die Erschließung neuer Autonomiestufen. Eingebettet in KI-Agenten ist es möglich, diese Fähigkeiten mit aktiven Handlungen zu verbinden, entweder durch Roboter in der physischen Welt oder Softwarekomponenten in der digitalen Welt.

### 2.3.2 Methoden für KI-basiertes Lernen

Für die Entwicklung einer Vision zur Entwicklung und zum Betrieb von Systemen mit hoher Autonomiestufe ist die Definition zentraler Begriffe aus dem Bereich der KI ausreichend, wie sie im vorangegangenen Unterabschnitt aufgestellt wurden. Für die Generierung konkreter Lern- und Testumgebungen für AMR ist zusätzlich eine detailliertere Betrachtung geeigneter KI-Methoden erforderlich, um die spezifischen Anforderungen der eingesetzten Lernverfahren in das Design der Umgebungen und den Generierungsprozess einfließen zu lassen.

Ein vielversprechender Ansatz zum Lernen von Steuerungsentscheidungen durch autonome Roboter ist das bestärkende Lernen (engl. Reinforcement Learning (RL)). Dieses Verfahren hat sich im Bereich der Multi-Roboter-Systeme bewährt [Aru+17; EI21] und wurde bereits erfolgreich auf verschiedene Steuerungsaufgaben von AMR in intralogistischen Systemen angewandt (vgl. Unterabschnitt 2.2.3, Literaturübersicht in Tabelle A.1), sodass RL als Wegbereiter für autonome Transportroboter in der Intralogistik gehandelt wird [Cha+20]. Besonders relevant in diesem Zusammenhang ist das End-to-End-RL, bei dem ein Agent lernt, direkt von rohen Sensordaten (z. B. Kamerabildern) auf konkrete Steuerbefehle (z. B. Bewegungen von Rädern oder Greifarmen) zu schließen, ohne Zwischenverarbeitungsschritte.

Da das RL-basierte Training typischerweise in virtuellen Lernumgebungen stattfindet, muss der sogenannte Sim-to-Real-Gap überbrückt werden, um die erlernten Strategien in realen Systemen anzuwenden. Eine etablierte Methode zur Erhöhung der Robustheit gegenüber realen Umgebungsvariationen ist die Domänenrandomisierung (engl. Domain Randomization (DR)).

Im folgenden Abschnitt werden daher diese beiden für die Generierung von Lern- und Testumgebungen relevanten Lernkonzepte vorgestellt.

#### 2.3.2.1 Reinforcement Learning

RL ist neben dem überwachten und dem unüberwachten Lernen einer der drei Hauptansätze des ML. Beim RL lernen Agenten nach dem Prinzip des *Versuchs und Irrtums*, indem sie ihre Umgebung beobachten, daraufhin Aktionen auswählen und ausführen sowie Belohnungen für ihr Verhalten erhalten [SB18, S. 1 f.]. RL verfolgt das Ziel, eine Strategie zu erlernen, die es dem

Agenten erlaubt, in einer gegebenen Situation jene Aktion auszuwählen, die in einer Episode zur Maximierung der kumulierten Belohnung führt.

### Problemdefinition

Abhängig von der Anzahl der beteiligten Agenten lässt sich zwischen Single-agent Reinforcement Learning (SARL) und Multi-agent Reinforcement Learning (MARL) unterscheiden. Da die Elemente, die zur Abbildung der Probleme in virtuellen Lern- und Testumgebungen beachtet werden müssen, für beide Arten von Problemen die gleichen sind, werden im Folgenden grundlegende Begriffe und Elemente des SARL eingeführt.

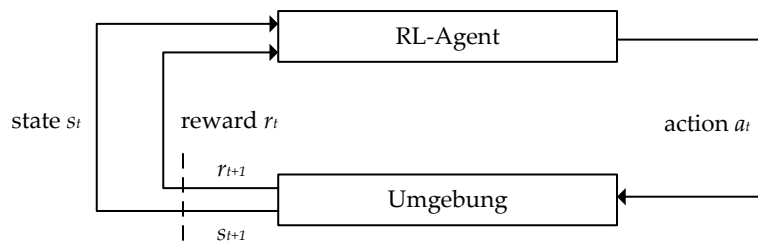


Abb. 2.17 Die Interaktion zwischen Agent und Umgebung in einem MDP [SB18, S. 48]

Das SARL-Problem wird als System aus einem RL-Agenten und seiner Umgebung modelliert (siehe Abbildung 2.17). Dieses System folgt einem stochastischen Prozess, der die sogenannte Markov-Eigenschaft erfüllt. Diese ist gegeben, wenn die bedingten Wahrscheinlichkeiten zukünftiger Zustände der Umgebung ausschließlich vom aktuellen Zustand abhängen und nicht von vergangenen Zuständen. Das RL-Problem lässt sich daher formal als Markov-Entscheidungsproblem (engl. Markov Decision Problem (MDP)) beschreiben [WHK22]. Ein MDP ist durch folgendes Quadrupel definiert:

$$M = (S, A, R, \mathcal{P}) \quad (2.1)$$

Die Umgebung wird als Menge von Zuständen (engl. States)  $S$  repräsentiert, und der Agent hat in jedem Zustand eine Menge von verfügbaren Aktionen (engl. Actions)  $A$  zur Auswahl. Für jede Aktion  $a_t \in A$ , die der Agent in einem Zustand  $s_t \in S$  ausführen kann, existiert eine Übergangswahrscheinlichkeit der Umgebung vom Zustand  $s_t$  zum nächsten Zustand  $s_{t+1} : P(s_{t+1} | s_t, a_t)$ . Die Menge der Übergangswahrscheinlichkeiten wird mit  $\mathcal{P}$  bezeichnet. Die Veränderungen der Zustände der Umgebung ergeben sich dabei zum einen durch die Aktion des Agenten und können sich andererseits ebenfalls durch die Umgebung selbst ergeben. Nach jeder ausgeführten Akti-

on erhält der Agent ein Feedback in Form einer Belohnung (engl. Reward)  $r_t = R(s_t, a_t, s_{t+1})$ . Die Belohnungsfunktion  $R$  kann in verschiedenen Formen ausgedrückt werden, häufig wird sie in Abhängigkeit von der gewählten Aktion  $R(a_t)$  dargestellt oder als Funktion von Aktions-Zustands-Paaren  $R(a_t, s_t)$ , die bewertet, wie gut oder schlecht eine gewählte Aktion in einem gegebenen Zustand der Umgebung ist [AG21]. Das Ziel des Agenten ist es, die Entscheidungen für Aktionen so zu treffen, dass die kumulierte Belohnung, die über  $T$  aufeinanderfolgende Schritte  $\sum_{i=0}^T r_i$  erreicht wird, maximiert wird.

Die Entscheidungsfindung für eine Aktion wird im RL durch eine Strategie (engl. Policy) abgebildet. Die Strategie erfasst die Zuordnung von Zuständen der Umgebung zu Aktionen des Agenten und bildet die Regeln ab, nach denen ein Agent entscheidet, welche Aktion zu wählen ist. Es gibt deterministische und stochastische Strategien. Deterministische Strategien  $\mu$  sind wie folgt definiert [Ope18a]:

$$a_t = \mu_\theta(s_t) \quad (2.2)$$

Ist eine Strategie  $\pi$  stochastisch, folgt also einer Wahrscheinlichkeitsfunktion, wird sie wie folgt definiert [Ope18a]:

$$a_t \sim \pi_\theta(\cdot | s_t) \quad (2.3)$$

Die Strategie  $\pi_\theta(a|s)$  in Formel 2.3 beschreibt die Wahrscheinlichkeit, dass der Zustand der Umgebung  $s \in S$  der Aktion  $a \in A$  des Agenten zugeordnet ist.  $\theta$  bezeichnet das Set der Parameter, die die Funktion der Strategie definieren und veränderbar sind, beispielsweise Gewichte und Biases eines Neuronalen Netzes. Durch die Veränderung dieser Parameter wird die Strategie und damit das Verhalten des Agenten verändert [Ope18a].

Ein weiteres wichtiges Element in RL ist der Verlauf bzw. die Trajektorie (engl. Trajectory)  $\tau$ . Die Trajektorie ist die Sequenz der Zustände und Aktionen in der Umgebung, die aufeinander folgen [Ope18a]:

$$\tau = (s_0, a_0, s_1, a_1, \dots) \quad (2.4)$$

Die Zustände der Umgebung  $s_t$  der Trajektorie können dabei einer deterministischen oder einer stochastischen Funktion folgen. Die Aktionen  $a_t$  werden durch den Agenten anhand seiner Strategie bestimmt [Ope18a]. Die erwartete Belohnung ist dann:

$$J(\pi) = E_{\tau \sim \pi} [R(\tau)] \quad (2.5)$$

Und das grundlegende Optimierungsproblem in RL zur Bestimmung der optimalen Strategie  $\pi^*$  zur Maximierung der Belohnung kann dann wie folgt ausgedrückt werden [Ope18a]:

$$\pi^* = \arg \max_{\pi} J(\pi) \quad (2.6)$$

Eine Erweiterung des MDP, das für RL-Probleme im Kontext autonomer Transportroboter in der Intralogistik relevant ist, ist das partiell beobachtbare MDP (engl. Partially Observable Markov Decision Problem (POMDP)). Bei dieser Erweiterung des MDP nimmt ein Agent nur Teile der Umgebung wahr und kennt nicht alle Zustände der gesamten Umgebung zu dem Zeitpunkt, zu dem er die Entscheidung für eine Aktion trifft. Da AMR in intralogistischen Systemen ihre Umwelt über eigene Sensorik wahrnehmen, kennen sie nur ihre lokale Umgebung und nicht den Zustand des gesamten Systems. Ein POMDP wird definiert als

$$M = (S, A, R, \mathcal{P}, \Omega, O) \quad (2.7)$$

$S$ ,  $A$ ,  $R$  und  $\mathcal{P}$  beschreiben das MDP,  $\Omega$  ist die Menge der Beobachtungen (engl. Observations) und  $O: S \times A \rightarrow \Pi(\Omega)$  ist die Beobachtungsfunktion, die eine Wahrscheinlichkeitsverteilung über mögliche Beobachtungen für jede durchgeführte Aktion und den vom Agenten erreichten Zustand liefert. Auch beim POMDP ist das Ziel des Agenten die Maximierung der zukünftigen Belohnung [WHK22].

Generell werden Strategien durch den erwarteten Wert der zukünftigen Belohnung bestimmt, auch Wertfunktion (engl. Value function)  $V(s)$  genannt, indem numerische Werte für einzelne Zustände bestimmt werden ([MKV20, S. 2], [SB18, S. 58]):

$$V^{\pi}(s) = E_{\tau \sim \pi} [R(\tau) | s_0 = s] \quad (2.8)$$

Im Vergleich zur Zustandswertfunktion  $V(s)$  definiert die Q-Wertfunktion  $Q(s, a)$  den Wert von Zustands-Aktions-Paaren, also den Wert jeder Aktion in jedem Zustand, und funktioniert daher auch, wenn die Wahrscheinlichkeit von Übergängen zwischen den Zuständen unbekannt ist [SB18, S. 58]:

$$Q^{\pi}(s, a) = E_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a] \quad (2.9)$$

### Reinforcement-Learning-Algorithmen

Es gibt verschiedene Algorithmen, die das RL-Problem in Gleichung 2.6 für SARL- und MARL-Umgebungen lösen (siehe Abbildung 2.18). Sie können in modellfreie und modellbasierte Algorithmen unterschieden werden. Modellbasierte Algorithmen nutzen ein Modell zur Vorhersage der Reaktion der Umgebung auf eine Aktion. Der Begriff *Modell* bezeichnet hierbei beispielsweise eine Funktion, die die Zustandsübergangswahrscheinlichkeiten und Belohnungen vorhersagt.

Fast alle Algorithmen nutzen die Wertfunktion  $V^\pi(s)$  bzw.  $Q^\pi(s, a)$  die abschätzt, wie gut es für den Agenten ist, sich in einem bestimmten Zustand zu befinden, oder wie gut es ist, eine bestimmte Handlung in einem bestimmten Zustand durchzuführen [SB18, S. 58]. Häufig verwendete Algorithmen sind insbesondere Q-Learning-Ansätze und Proximal Policy Optimization (PPO). Während Q-Learning [SB18, S. 5] und Deep Q-Networks (DQN) bei SARL sehr beliebt sind, sind Algorithmen wie PPO und A3C oft besser geeignet für komplexere Szenarien, einschließlich MARL.

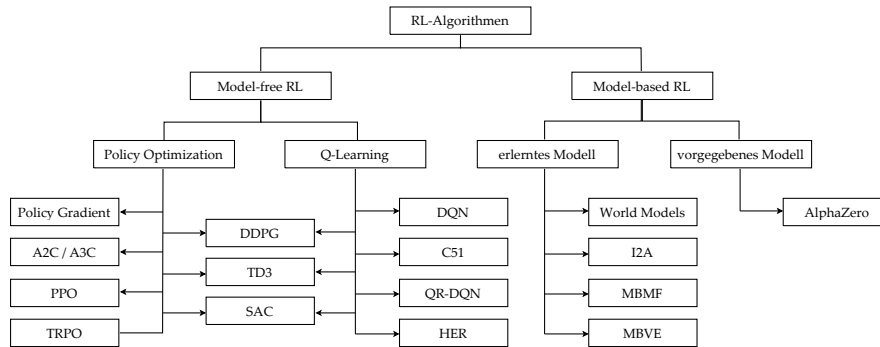


Abb. 2.18 Taxonomie von RL-Algorithmen [Ope18b]

### 2.3.2.2 Domain Randomization

RL ist ein vielversprechender Ansatz zur Lösung komplexer Entscheidungsprobleme von AMR in der Intralogistik. Da das Training eine große Anzahl an Interaktionen zwischen den Agenten und ihrer Umwelt erfordert, ist das Lernen direkt am realen, physischen Roboter mit erheblichen Herausforderungen verbunden. Dazu zählen hohe Kosten pro Trainingsdurchlauf, Risiken für die Sicherheit von Menschen und Maschinen sowie Einschränkungen bei Beobachtbarkeit und Parallelisierbarkeit der Lernprozesse.

Aus diesem Grund werden RL-Strategien für autonome Roboter typischerweise in virtuellen Simulationsumgebungen trainiert (siehe Unterunterabschnitt 2.3.2.3). Diese Lernumgebungen für die autonomen Agenten der Roboter müssen dabei die Kernkomponenten einer RL-Aufgabe abbilden: die Wahrnehmung von Umgebungszuständen, die Ausführung von Aktionen basierend auf diesen Wahrnehmungen und die Vergabe von Belohnungen in Abhängigkeit von der Qualität der gewählten Aktion. Die daraus abgeleitete Strategie, also die Abbildung von Zuständen auf Handlungen gemäß der Belohnungsfunktion, kodiert die Steuerungsstrategie der AMR und ist das zentrale Ergebnis des Trainings, das später im physischen System eingesetzt wird.

Ein wesentliches Problem bei der Übertragung der in virtuellen Umgebungen erlernten Strategien auf reale Systeme ist jedoch die Abweichung

zwischen der simulierten und der realen Umgebung, die als Sim-to-Real-Gap bezeichnet wird. Diese Unterschiede können dazu führen, dass eine in der Simulation erfolgreich erlernte Steuerung in der Realität nicht wie erwartet funktioniert.

Eine zentrale Methode zur Reduktion dieses Sim-to-Real-Gaps ist die Domain Randomization (DR). Dabei handelt es sich um eine Technik zur Verbesserung der Generalisierungsfähigkeit von RL-Modellen, bei der systematisch Variationen in der Simulationsumgebung eingeführt werden, um die Generalisierungsfähigkeit des erlernten Verhaltens auf reale Einsatzbedingungen zu erhöhen. Das grundlegende Prinzip ist, die Strategie nicht auf einzelne, idealisierte Simulationsmodelle zu trainieren, sondern RL-Agenten mit einer Vielzahl möglicher Umgebungsbedingungen zu konfrontieren. Studien zeigen, dass eine große Vielfalt an Trainingsumgebungen die Generalisierungsfähigkeit von Agenten verbessert [RT20] und in rekurrenten neuronalen Netzen sogar Meta-Lernen fördern kann, das eine Anpassung an zuvor unbekannte Situationen ermöglicht. Die Randomisierung von Umgebungsaspekten während des Trainings ist somit auch ein bewährter Ansatz, um die Übertragbarkeit auf die reale Welt zu verbessern [Eße+22; Mur+22; ZQW20].

Dazu werden zum einen physikalische Parameter wie Objektpositionen, Größen, Farben, Texturen oder Lichtverhältnisse systematisch verändert [Meh+20]. Zum anderen umfasst DR die strukturellen Variationen der Umgebung, etwa durch veränderte Anordnungen von Regalen, dynamische Layouts von Lagerbereichen oder zufällige Positionierung von Hindernissen. Durch diese Form der strukturellen Randomisierung werden autonome Systeme wie AMR unterstützt, robuste Steuerungsstrategien zu erlernen, die wahrscheinlicher in realistischen Einsatzszenarien funktionieren.

### 2.3.2.3 Simulation

Der Einsatz von RL für die Entwicklung und den Betrieb der Steuerung von AMR erfordert virtuelle Umgebungen, die dynamisches Verhalten abbilden können. Daraus ergibt sich, dass die durch das in dieser Arbeit entwickelte Verfahren generierten Lern- und Testumgebungen für AMR in Materialflusssystemen Simulationsmodelle sind.

Laut VDI 3633 Blatt 1 ist Simulation das „Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind“ [VDI 3633-1, S. 3]. In Simulationsstudien geht es insbesondere um die Beobachtung und Bewertung der abgebildeten Prozesse über die Zeit des Simulationslaufs, die Abbildung stochastischer Einflüsse, die Analyse von Wirkzusammenhängen zwischen eigenständigen Systemen sowie die Bildung von Kennzahlen zur Bewertung von Steuerungsalternativen [Wen18, S. 2].

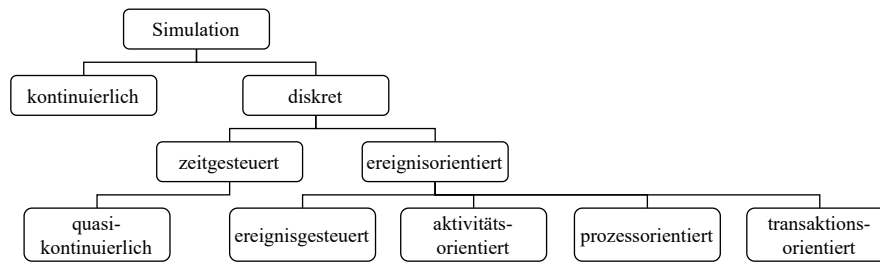


Abb. 2.19 Klassifikation von Simulationsmethoden nach [MM89, S. 202]

Bei einer Simulation können verschiedenen Methoden zum Einsatz kommen (siehe Abbildung 2.19). Die Simulationsmethode definiert die Art und Weise, wie das Zeitverhalten in der Simulation berücksichtigt wird (vgl. [Wen18, S. 9 ff.]). Mit dem Fortschreiten der Simulationszeit verändert sich der Zustand des Modells, der durch die Zustände der einzelnen Modellelemente beschrieben wird. Die Zeit kann dabei kontinuierlich oder diskret fortgeschrieben werden. Bei der *kontinuierlichen Simulation* werden die Zustandswerte über Differenzialgleichungen in einem stetigen Verlauf über die Zeit abgebildet. Bei der *diskreten Simulation* erfolgen die Zustandsänderungen zu bestimmten Zeitpunkten. Die Zeitpunkte werden entweder an Ereignissen festgemacht (*ereignisorientierte Simulation*) oder über feste Zeitintervalle gesteuert (*zeitgesteuerte Simulation*). Werden die Zeitinkremente bei der zeitdiskreten Simulation sehr klein gewählt, nähert sie sich der kontinuierlichen Simulation an und wird als *quasi-kontinuierlich* bezeichnet. Bei der ereignisorientierten Simulation können die Zustandsänderungen durch verschiedene Mechanismen ausgelöst werden. Die *ereignisgesteuerte Simulation* beschreibt das System als eine Menge von Ereignissen, die zu bestimmten Zeitpunkten eintreten und Zustandsänderungen bewirken. Bei der *aktivitätsorientierten Simulation* wird eine Menge von möglichen Aktivitäten zyklisch daraufhin überprüft, welche der Aktivitäten ausgelöst werden kann. Der *prozessorientierte* Ansatz organisiert das Modell in interagierende parallele Prozesse, die Zustände verändern können. Dieser Ansatz wird im *transaktionsorientierten* Modell dahingehend erweitert, dass es mobile und dynamische Objekte in dem Modell gibt. [Arn08, S. 79 f.]

Für die Entwicklung und den Betrieb von AMR in Materialflusssystemen sind zwei verschiedene Arten der Simulation relevant: die Materialflusssimulation und die Robotiksimulation bzw. agentenbasierte Simulation (siehe Tabelle 2.3). In der Logistik wird klassischerweise die ereignisorientierte Simulationsmethode genutzt [Wen18, S. 11]. Dabei erstrecken sich die Anwendungsbereiche von der Betrachtung der innerbetrieblichen Materialflüsse über die Auftragsabwicklung in der Produktion bis hin zur Beschaffungs- und Distributionslogistik [Wen18, S. 4]. Im Sinne der Hierarchiepyramide logistischer Systeme Abbildung 2.3 werden insbesondere die Entscheidungen

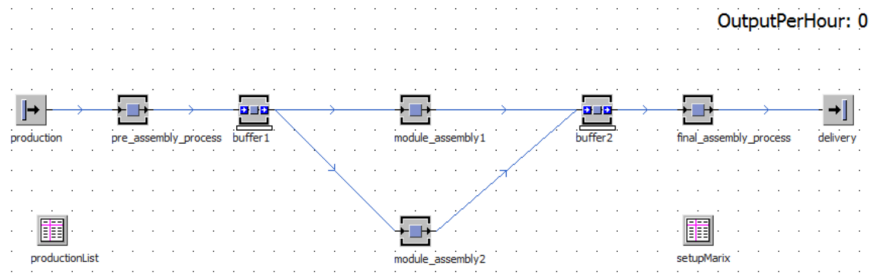
auf Ebene der Materialflusssysteme bis zur Ebene der Lieferkette betrachtet. Simulationsstudien in der Logistik verfolgen klassischerweise die Ziele, Anlagenplanungen durch Funktions- und Leistungsnachweise in der Planungsphase abzusichern oder das Systemverhalten durch Betrachtung von Steuerungsalternativen in der Betriebsphase der Materialflusssysteme zu verbessern ([Wen18, S. 5], [Wen+19, S. 2190]).

Insgesamt geht es bei der Materialflusssimulation insbesondere um die Abbildung und Untersuchung von Prozessen und des Verhaltens einzelner Komponenten und des gesamten Materialflusssystems im Zeitverlauf, etwa mit dem Ziel der Engpassanalyse oder Optimierung. Daher werden die Systemelemente, die das Verhalten der Komponenten simulieren, meist abstrahiert in Form von Blöcken dargestellt (siehe Abbildung 2.20 (a)). Da es vor allem um die Abbildung von Prozesszeiten wie Bearbeitungs- und Wartezeiten geht, wird die Physik der realen Welt in diesen Simulationen gemäß dem Verkürzungsmerkmal von Modellen (siehe Abschnitt 2.4) nicht abgebildet.

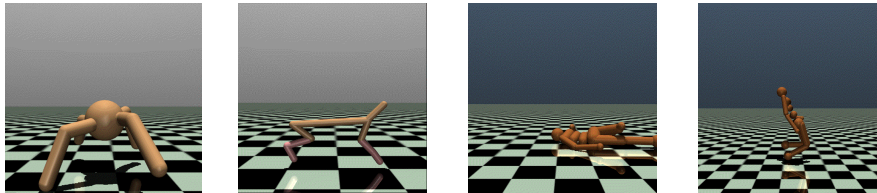
**Tabelle 2.3** Vergleich Materialflusssimulation und Robotiksimulation

	<b>Materialflusssimulation</b>	<b>Robotiksimulation</b>
<b>Simulationsmethode</b>	ereignisorientiert	zeitgesteuert
<b>Steuerungs- entscheidungen</b>	Ebene Materialflusssysteme und höher in Abbildung 2.3	Ebene Technologie in Abbildung 2.3
<b>Typische Ziele</b>	Analyse von Materialflüssen Absicherung Anlagenplanung Funktions- und Leistungsnachweise Verbesserung Systemverhalten	Hardware Prototyping Entwicklung von Steuerungsstrategien für Roboter Testen von Steuerungsstrategien für Roboter
<b>Visualisierung</b>	abstrahierte Objekte, Blöcke	möglichst realitätsgetreue Objekte
<b>Simulierte Elemente</b>	Quellen, Senken, Server, Warteschlangen	Objekte inkl. ihrer physikalischen Eigenschaften, Sensoren, Aktoren, Recheneinheiten, Kommunikationsinfrastruktur
<b>Physik</b>	nicht abgebildet	abgebildet

Dahingegen spielt die Abbildung von physikalischen Eigenschaften in Robotiksimulationen eine wichtige Rolle. Im Bereich der Entwicklung und des Betriebs von Robotern wie AMR werden Simulationen eingesetzt, um Hardwarekomponenten wie Sensoren, Aktoren, Rechen- und Kommunikationsinfrastruktur (siehe Tabelle 2.1) zu testen, bevor sie in einen realen Roboter eingebaut werden, und um Steuerungsmethoden für das Verhalten der Roboter zu entwickeln und zu testen, bevor es in der Realität umgesetzt wird [Cho+21, S. 2]. Die Abbildung der Physik ist dabei von Bedeutung, um



(a) Beispielhafte Visualisierung einer Materialflusssimulation



(b) Beispielhafte Visualisierung einer Robotersimulation

**Abb. 2.20** Visualisierung in Materialfluss- und in Robotersimulatoren ([Wen+19, S. 2195], [Ope22])

die Bedingungen der Umgebungen, in denen die realen Roboter arbeiten werden, möglichst genau abzubilden und den Sensor-Input, die Berechnung der daraus folgenden Handlung und die Umsetzung dieser durch die Aktorik so zu simulieren, dass die Ergebnisse in die Realität übertragen werden können ([LN21, S. 45 ff.], [Dur+11]). Je genauer die Abbildung erfolgt, desto geringer wird der Sim-to-Real-Gap angenommen, der die Robotikentwicklung bei der Übertragung der Ergebnisse vor Herausforderungen stellt. Die sehr realitätsnahen Simulationen werden auch High-Fidelity-Simulationen genannt. Nach aktuellen Studien verbessert eine höhere Realitätstreue jedoch nicht für jede Aufgabe der Robotik die Übertragbarkeit in die Realität, und das richtige Level an Abstraktion zu finden, kann von Aufgabe zu Aufgabe unterschiedlich sein [Tru+22]. Die Abbildung der Physik ist der Grund, aus dem in Robotersimulatoren der zeitgesteuerte Ansatz angewandt wird, um quasi-kontinuierliche Physikzustände zu simulieren, wie sie in der Realität bestehen. Die Visualisierung der Objekte in Robotersimulatoren erfolgt basierend auf ihrer realen, physischen Form, häufig in 3D (siehe Abbildung 2.20 (b)), wobei für einige Aufgaben auch 2D-Simulatoren zum Einsatz kommen können (siehe z. B. [Sur+20]).

Bei der Entwicklung von Robotern für den Einsatz in Materialflusssystemen, die auf hohen Autonomiestufen agieren, muss eine Kombination dieser beiden Ansätze gefunden werden. AMR selbst sind Roboter; die Umgebungen, mit denen sie interagieren, sind Materialflusssysteme.

### 2.3.3 Entwicklungs- und Betriebsprozesse von KI-Systemen

Wie in den vorangegangenen Abschnitten erläutert, nehmen KI und insbesondere ML und RL in Simulationsumgebungen eine wichtige Rolle auf dem Weg zur vollständigen Autonomisierung von Systemen ein. Der folgende Abschnitt beschreibt die organisatorischen und technischen Konzepte, mit denen KI-Systeme heute entwickelt, getestet, bereitgestellt und im Betrieb kontinuierlich verbessert werden. Im Fokus stehen MLOps-Prinzipien und simulationsgestützte Workflows, die die Teilautomatisierung von Training, Validierung und Bereitstellung von KI-Modellen ermöglichen.

#### 2.3.3.1 MLOps

Ein Ansatz zur Entwicklung und zum Betrieb ML-basierter Lösungen ist das Konzept des MLOps. Dieses ML-Engineering-Konzept verfolgt das Ziel, manuelle ML-Prozesse zu automatisieren und zu operationalisieren [KKH23, S. 31866].

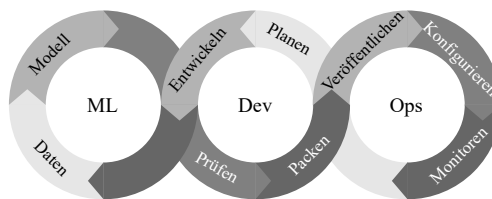


Abb. 2.21 Erweiterung von DevOps zu MLOps [Sym+22, S. 454]

MLOps basiert auf den Konzepten des Development Operations (DevOps) (siehe Abbildung 2.21), das darauf abzielt, technische Hindernisse in der Softwareentwicklung zu überwinden, indem es die Automatisierung der Überführung der Software in die Produktion durch die Strategie der kontinuierlichen Integration, kontinuierlichen Auslieferung und kontinuierlichen Bereitstellung unterstützt (engl. Continuous Integration, Continuous Delivery (CI/CD)) [KKH23, S. 31867]. Bei der kontinuierlichen Integration wird das Ziel verfolgt, neuen Code, der von Entwicklungsteams geschrieben wurde, in kurzen Intervallen in bestehende Software zu integrieren. Der Code wird dabei ständig getestet und jedes Mal geringfügig auf Basis der gefundenen Fehler und Schwächen verbessert. Dies führt zu einer Verkürzung des Softwareentwicklungszyklus. Kontinuierliche Bereitstellung bezieht sich darauf, dass ständig eine neue Version der Software für Tests, Bewertung und dann Produktion zur Verfügung gestellt wird. Mit diesem Vorgehen erreichen die Softwareversionen, die aus der kontinuierlichen Integration mit Verbesserungen und neuen Funktionen resultieren, die Endbenutzer schneller [Sym+22, S. 454]. MLOps erweitert dieses Konzept um die Automatisierung der ML-Prozesse.

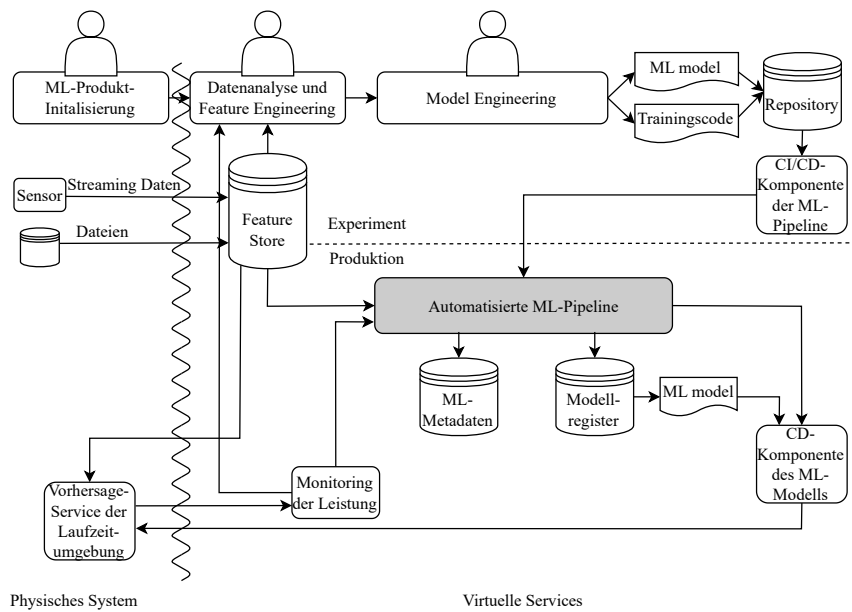


Abb. 2.22 MLOps-Architektur und -Workflow (i. A. a. [Goo23], [KKH23, S. 31873])

Abbildung 2.22 zeigt einen typischen MLOps-Workflow. Er ist grundsätzlich in die Phasen des Experimentierens (oben) und der Produktion (unten) unterteilt.

Im Bereich des Experimentierens werden basierend auf der ML-Produktidee durch Data Scientists und Softwareentwickler Datenanalysen und Feature Engineering für den Anwendungsfall durchgeführt, um dann im Model Engineering ein passendes ML-Modell und den dazugehörigen Trainingscode entwickeln zu können. Diese beiden Artefakte werden in einem Repository abgelegt.

Sind alle MLOps-Komponenten durch die Entwickler passend aufgesetzt, läuft der Prozess ab diesem Punkt automatisiert: Die CI/CD-Komponente der ML-Pipeline bringt diese inklusive des Trainingscodes und Modell-Artefakts automatisiert in die Produktion im operativ agierenden System. Durch die Pipeline werden ML-Modelle automatisiert auf aktuellen Daten trainiert und die Trainingsparameter angepasst, sodass das beste ML-Modell gefunden wird. Dieses wird dann durch eine CD-Komponente automatisiert in die Laufzeitumgebung integriert, in der es der umgebenden Software als Service zur Verfügung steht. Die Leistung des ML-Modells wird kontinuierlich geprüft, und sobald Grenzwerte überschritten werden, wird das Modell entweder automatisiert neu durch die ML-Pipeline trainiert oder es findet ein neuer manueller Experimentierprozess statt.

Nach diesem Prinzip werden durch die definierten Abläufe im MLOps-Konzept die Betriebs- und Entwicklungsprozesse von KI-Lösungen teilweise automatisiert.

### 2.3.3.2 Simulationsgestützte Entwicklungsprozesse

Konzepte für automatisierte Workflows zwischen virtueller und realer Welt existieren auch für die Kombination aus ML und Simulation, wie es für RL-basierte Systeme häufig der Fall ist.

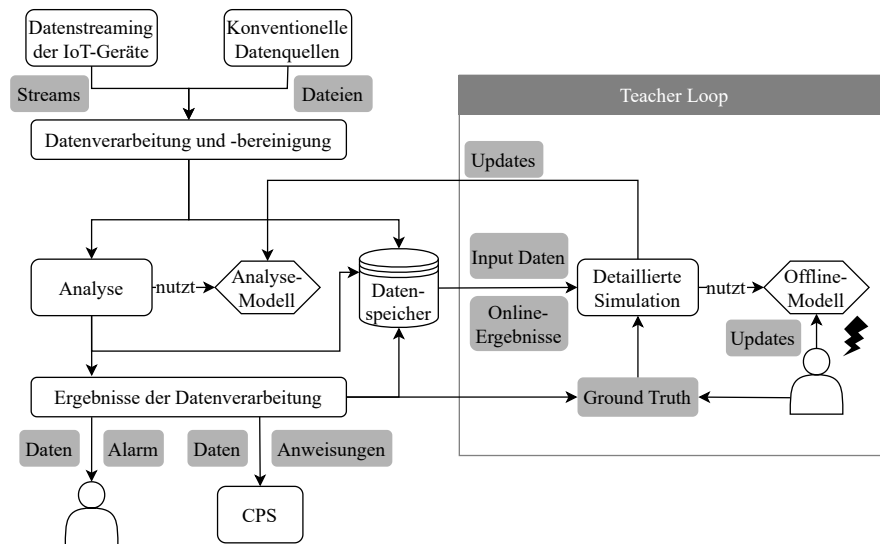


Abb. 2.23 Typischer Workflow aus Simulation und ML [Ant+19, S. 15]

Ein typischer Workflow von ML in Kombination mit Simulation wird in [Ant+19] vorgestellt und ist in Abbildung 2.23 aufgezeigt.

Daten werden von einer Vielzahl von physischen Sensoren oder anderen Quellen (z. B. Datenbanken) zum Zustand der physischen Realität aufgenommen, vorverarbeitet und bereinigt. Die Daten werden zur Dokumentation und zur Verbesserung der Analyse- und Simulationsmodelle gespeichert.

Der zweite Schritt besteht aus einer eingehenden Analyse der Daten. Das Analysemodell kodiert dabei die Analysemethoden und kann von einem einfachen Regelwerk bis hin zu einer komplexen Simulation reichen.

Im dritten Schritt werden die Analyseergebnisse verarbeitet und an menschliche Benutzer oder CPS kommuniziert.

Im Teacher Loop wird das Analysemodell, das im Produktivsystem der Lösung genutzt wird, trainiert. Je nachdem wie das Analysemodell aufgebaut ist, kann der Teacher Loop beispielsweise dazu genutzt werden, RL-Agenten in einer Simulationsumgebung zu trainieren, die dann im Analysemodell in der Produktivumgebung eingesetzt werden. Ist das Analysemodell selbst ein

Simulationsmodell, kann der Teacher Loop ebenfalls dazu genutzt werden, Parameter des Simulationsmodells auf neuen Daten zu trainieren, um die realen Gegebenheiten besser abzubilden.

In jedem Fall ist in dem Teacher Loop ein Mensch involviert, um beim Experimentieren und bei der Synchronisation des Modells mit den Daten der Ground Truth aus der realen Welt zu unterstützen.

## 2.4 Modellierung von Materialflusssystemen

Die virtuellen Lern- und Testumgebungen für AMR im Digitalen Kontinuum, die durch das in dieser Arbeit entwickelte Verfahren generiert werden, sind abstrahierte Simulationsmodelle der realen Welt. Aus den Eigenschaften von Modellen ergeben sich daher Aspekte, die bei der Generierung dieser Umgebungen beachtet werden müssen.

Der Begriff *Modell* wird in den unterschiedlichsten Zusammenhängen verwendet, um die Abstraktion einer komplexen Realität in eine leicht verständliche Darstellung zu beschreiben [AL12, S. 408]. Nach Stachowiak sind Modelle verkürzte Abbildungen eines Originals, die ausgewählte Eigenschaften eines Systems einbeziehen, um diese untersuchbar zu machen [Sta73]. Es lassen sich drei fundamentale Merkmale definieren, die jedes Modell besitzt:

- *Abbildungsmerkmal*: Ein Modell ist stets ein Abbild eines Originals, das entweder real oder künstlich sein kann. Originale können selbst ebenfalls Modelle sein.
- *Verkürzungsmerkmal*: Nicht alle Eigenschaften eines Originals werden im Modell abgebildet, sondern nur solche, die für den Ersteller oder den Nutzer des Modells im Sinne des vorgesehenen Zwecks relevant erscheinen.
- *Pragmatisches Merkmal*: Modelle werden für bestimmte Nutzer, seien es Menschen oder Maschinen, und unter spezifischen Rahmenbedingungen erstellt, um über einen definierten Zeitraum hinweg bestimmte Funktionen zu erfüllen und einem expliziten Zweck zu dienen.

Modellierung, der Prozess der Modellerstellung, bedeutet somit im Grunde, dass ein Original auf bestimmte Aspekte reduziert wird, die einem bestimmten Zweck dienen. Dieser Zweck wird von einer bestimmten Person zu einem bestimmten Zeitpunkt festgelegt.

### 2.4.1 Model-based Systems Engineering

Für die modellbasierte Entwicklung und den Betrieb von Systemen, wie sie die Entwicklung autonomer Systeme auf der Basis von Simulationsmodellen ist, wird im Model-based Systems Engineering (MBSE) ein systematischer Ansatz vorgeschlagen. MBSE bildet die Grundlage für das Verwalten und Abbilden der Systemspezifikationen und des Designs, bei dem formale, digitale

Modelle eines komplexen Systems im Mittelpunkt stehen. Dieser Ansatz steht damit im Gegensatz zu dokumentenbasierten, konventionellen Ansätzen.

Im MBSE werden die allgemeinen Merkmale von Modellen um Anforderungen an ihre Funktionalität und Nutzbarkeit erweitert [Alt12, S. 410–412]:

- *Adäquatheit*: Modelle müssen die relevanten Eigenschaften des Originals hinreichend genau widerspiegeln.
- *Konsistenz*: Modelle dürfen keine inneren Widersprüche aufweisen.
- *Zweckorientierung*: Modelle sind auf konkrete Zielstellungen auszurichten und müssen entsprechend anpassbar sein.

Zudem werden im MBSE Dimensionen der Kontinuität von Modellen aufgestellt. Kontinuität beschreibt im Allgemeinen eine stufenlose Veränderung oder einen Zusammenhang, der in Zeit, Abfolge, Inhalt oder Ausmaß ununterbrochen ist [AL12, S. 411 f.].

- *Horizontale Kontinuität*: Die horizontale Kontinuität fordert die kontinuierliche Veränderung eines Modells, die durch das Fortschreiten im Lebenszyklus des Systems verursacht wird.
- *Vertikale Kontinuität*: Die vertikale Kontinuität berücksichtigt die kontinuierliche Veränderung der Modellgranularität (d. h. des Abstraktions- und Detaillierungsgrades) je nach Untersuchungsaspekt.

### 2.4.2 SysML

Im MBSE wird häufig die Modellierungssprache Systems Modeling Language™ (SysML®, nachfolgend SysML) genutzt. SysML ist eine standardisierte grafische Modellierungssprache, die von der Object Management Group® Standards Development Organization (OMG® SDO) speziell für die Beschreibung, die Analyse und das Design komplexer Systeme entwickelt wurde [OMG23]. SysML baut auf der Unified Modeling Language (UML®, nachfolgend UML) [OMG17] auf, einer graphischen Modellierungssprache für Softwaresysteme, und erweitert sie um spezifische Konstrukte für die Systemtechnik. Im Rahmen dieser Arbeit wird SysML zur Darstellung der Elemente der Simulationsmodelle in Unterunterabschnitt 5.2.1.1 genutzt, die durch den in dieser Arbeit entwickelten Algorithmus generiert werden. Um sicherzustellen, dass die definierten Modellelemente verständlich dargestellt sind, werden im Folgenden kurz die relevanten Grundkonzepte von SysML erläutert.

In der SysML erfolgt eine Trennung zwischen dem Modell selbst und seiner Sicht. Sämtliche Elemente eines Modells sind in einer zentralen Datenbank gespeichert und werden je nach spezifischem Modellierungszweck in verschiedenen Sichten aggregiert. Die verschiedenen Sichten werden in SysML mittels verschiedener Diagrammtypen dargestellt, die die Schnittstelle zwischen dem Modell und den Anwendern bilden [Alt12, S. 39]. Nicht in

jedem Diagramm sind daher alle Elemente des Modells aufgeführt und nicht alle Elemente des Modells sind Teil eines Diagramms.

SysML bietet neun verschiedene Diagrammtypen, die der Abbildung der Struktur, des Verhaltens und der Anforderungen eines Systems dienen (siehe Abbildung 2.24).

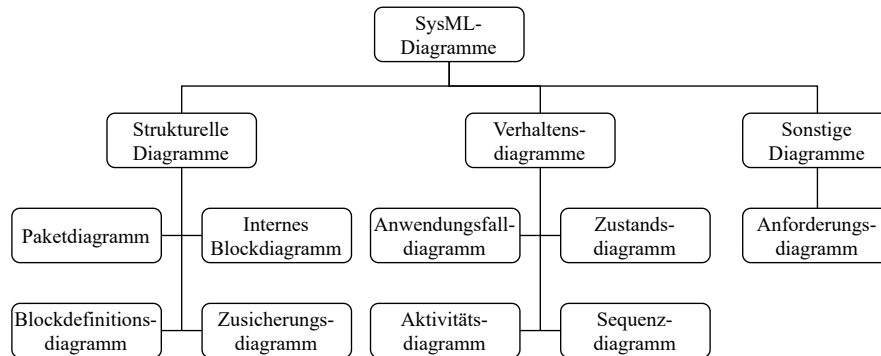


Abb. 2.24 SysML-Diagrammtypen nach [Alt12, S. 40]

Für die in dieser Arbeit genutzte Modellierung ist vor allem das Blockdefinitionsdiagramm (engl. Block Definition Diagram (BDD)) relevant. Dieses Diagramm ermöglicht die Definition von Blöcken, die äquivalent zu Klassen in UML sind und als grundlegende modulare Bausteine dienen, um die Struktur eines Systems in der SysML zu beschreiben. Abbildung 2.25 zeigt eine detaillierte Darstellung eines Blocks in SysML. Im Kopf eines Blocks sind der Stereotyp, potenziell zusätzliche Eigenschaften und der eindeutige Name des Blocks definiert. Der Körper des Blocks kann in der detaillierten Ansicht verschiedene Abschnitte enthalten, in denen Attribute des Blocks wie Bedingungen, Operationen, Referenzen, Werte, Teile und generelle Eigenschaften definiert werden.

In einem BDD werden Blöcke, die ein System beschreiben, miteinander in Beziehung gesetzt. Abbildung 2.26 zeigt die möglichen Arten von Beziehungen, die sich an der objektorientierten Notation von UML orientieren.

Eine Referenz mit Diamanten definiert eine Teil-Beziehung zwischen Blöcken. Auf der Seite mit dem Diamanten ist der Block, der das Ganze definiert, auf der anderen Seite das Teil des Ganzen. Ein schwarzer Diamant bedeutet, dass das Teil nur zu diesem Ganzen gehört, also exklusiver Teil ist. Bei einem leeren Diamanten kann das Teil auch in anderen Blöcken vorhanden sein. An der Teil-Beziehung können Kardinalitäten angegeben sein, die definieren, wie oft ein Element in einem anderen vorhanden ist.

Eine Generalisierungsbeziehung wird genutzt, um die Hierarchie und die Abstraktionsebenen in einem Modell darzustellen. Sie definiert eine Vererbungsbeziehung, bei der ein Element (das spezifischere Element) die Merkmale und das Verhalten eines anderen Elements (das allgemeinere

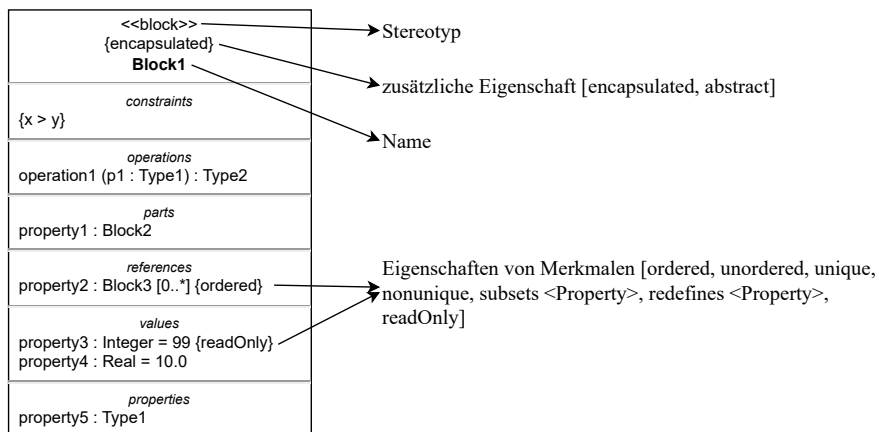


Abb. 2.25 SysML-Block

Element) erbt. In SysML wird die Generalisierungsbeziehung durch einen Pfeil dargestellt, der vom spezifischeren Element zum allgemeineren Element zeigt [Alt12, S. 43].

Eine einfache Linie zwischen zwei Blöcken bedeutet, dass die beiden Blöcke in einer Verbindung zueinander stehen. Die Art der Verbindung kann durch die Beschriftung der Linie genauer spezifiziert werden.

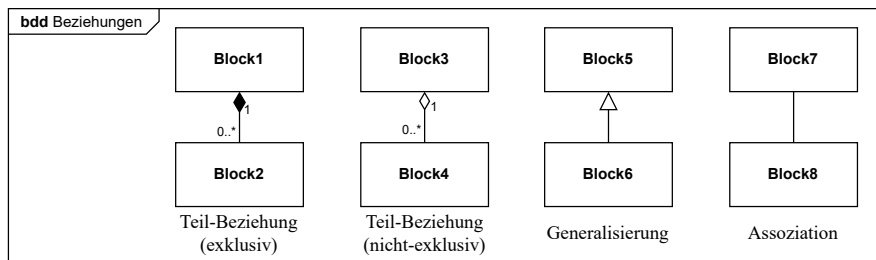


Abb. 2.26 Beziehungen zwischen Blöcken (i. A. a. [Sch20, S. 57])

## 2.5 Procedural Content Generation

Die automatisierte Generierung von Simulationsumgebungen, die als Lern- und Testumgebungen für AMR im Digitalen Kontinuum genutzt werden können, fällt unter das Forschungs- und Entwicklungsfeld der PCG. Dieses Feld beschäftigt sich mit der automatisierten Generierung von digitalen Inhalten und hat seinen Ursprung im Bereich der Computerspieleentwicklung. In diesem Abschnitt werden die Grundlagen der PCG vorgestellt.

### 2.5.1 Begriffsabgrenzungen

In verschiedenen Kontexten existieren verschiedene Definitionen für PCG (vgl. [FE17; Hen+13; RP04; Sme+11; STN16; Tog+11]). Alle Definitionen haben gemeinsam, dass sie PCG als den Prozess zur automatisierten Generierung von Inhalten definieren, der auf Algorithmen oder Mustern basiert und keine oder nur minimale Benutzereingaben zur Leitung der Generierung erfordert. Im Folgenden wird diese Definition von PCG zugrunde gelegt.

**Definition:** *Procedural Content Generation*

Procedural Content Generation ist die automatisierte Generierung von digitalen Inhalten, die auf Algorithmen oder Mustern basieren, und die keine oder nur minimale Benutzereingaben zur Steuerung der Algorithmen erfordert.

PCG zeigt damit Parallelen zu DR (vgl. Unterunterabschnitt 2.3.2.2). Die Ansätze stammen zwar aus unterschiedlichen Bereichen (RL und Spieleentwicklung), entwickeln jedoch beide Methoden zur Erzeugung von Daten oder Inhalten mithilfe von Algorithmen mit dem Ziel, Variabilität und Unvorhersehbarkeit in den erzeugten Inhalten zu schaffen ([RT20], [STN16]). Im weiteren Verlauf dieses Unterabschnitts wird der Begriff PCG verwendet, um sowohl PCG als auch DR zu beschreiben.

Die Definition der PCG zeigt außerdem Ähnlichkeiten zur Definition der generativen KI, die sich damit befasst, Inhalte zu generieren, die statistisch konsistent mit ihren Trainingsdaten sind (siehe Abschnitt 2.3). Methoden der generativen KI, wie Generative Adversarial Network (GAN), Variational Autoencoder (VAE) und rekurrente Neuronale Netze wie Long Short-Term Memory (LSTM), eröffnen neue Potenziale für die automatische Generierung von Inhalten, auch im Kontext der PCG. Sie bieten datenbasierte Ansätze zur Erzeugung komplexer Strukturen, etwa von Level-Layouts in Computerspielen oder Landschaften, und können damit klassische PCG-Techniken ergänzen. [Liu+21, S. 21]

Trotzdem ersetzt generative KI PCG nicht vollständig, sondern erweitert das Methodenrepertoire. Ein wesentlicher Unterschied ist, dass die für Computerspiele und die für Systemsimulationen generierten Inhalte nicht nur optisch konsistent, sondern auch funktional sinnvoll und interaktiv nutzbar sein müssen. Diese Aspekte können bislang nicht allein durch generative KI erreicht werden, sondern benötigen oft eine explizite Modellierung oder regelbasierte Ergänzung. [Liu+21, S. 21]

Hinzu kommt, dass die Anwendung generativer KI eine ausreichende Menge qualitativ hochwertiger Trainingsdaten voraussetzt, was im Bereich spezifischer Systemumgebungen wie der Intralogistik oft nicht gegeben ist. Für viele Anwendungsfelder stehen keine großen offenen Datensätze zur

Verfügung, was die Anwendbarkeit generativer KI-Modelle für die Generierung von Lern- und Testumgebungen für AMR im Digitalen Kontinuum einschränkt [MJV25].

### 2.5.2 Verfahren der PCG

Verfahren der PCG basieren auf der Kombination von Methoden zur Wissensrepräsentation und Methoden zur automatisierten Generierung von Inhalten [RT20, S. 3]. Eine Übersicht typischer Ansätze ist in Abbildung 2.27 dargestellt.

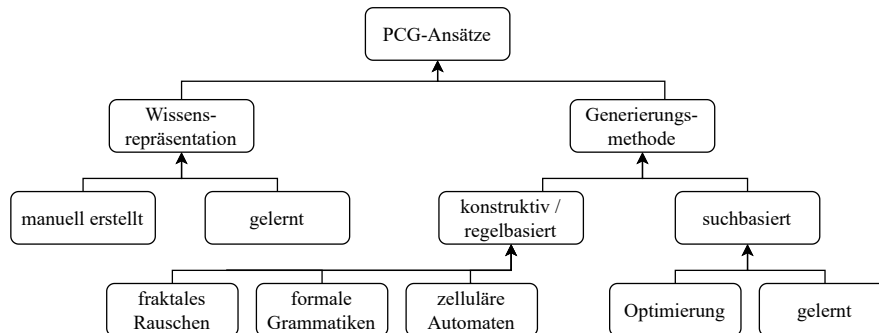


Abb. 2.27 Ansätze der Procedural Content Generation (i. A. a. [RT20, S. 3])

#### 2.5.2.1 Wissensrepräsentation in der PCG

Die Wissensrepräsentation im Kontext der PCG beschreibt die Form, in der Wissen über gültige Inhalte hinterlegt ist. Dabei wird unterschieden zwischen manuell erstellten und gelernten Repräsentationen.

*Manuell erstellte* Repräsentationen basieren auf expliziten Strukturvorgaben und Regelwerken, zum Beispiel durch modulare Bausteinbibliotheken oder graphbasierte Raumbeziehungen. Inhalte entstehen somit aus fest kodiertem, menschlich beschriebenem Domänenwissen.

Demgegenüber nutzen gelernte Repräsentationen Verfahren der KI, um aus Trainingsdaten implizite Strukturen abzuleiten. Beispiele hierfür sind GANs [Tor+19] und Diffusionsmodelle [Yu+23]. Diese Modelle bilden räumliche Muster, semantische Zusammenhänge und deren Variationsmöglichkeiten statistisch ab und können darauf basierend neue Inhalte erzeugen.

#### 2.5.2.2 Generierungsverfahren in der PCG

Gemäß [RT20, S. 3] lassen sich Verfahren zur PCG in konstruktive und suchbasierte Methoden einteilen.

### Konstruktive Generierung

Konstruktive Methoden erzeugen Inhalte direkt und ohne iterative Suche. Algorithmen in dieser Kategorie erzeugen das Generierungsergebnis in fester Zeit, wodurch diese Ansätze für Echtzeitanwendungen geeignet sind.

Ein Beispiel ist *fraktales Rauschen*, das genutzt wird, um natürliche, kontinuierliche Strukturen wie Höhenfelder oder Texturen zu erzeugen. Eines der bekanntesten Verfahren dieser Kategorie ist das *Perlin-Rauschen* [Per85]. Dabei werden stochastische Gradientenfelder über mehrere Frequenzen kombiniert, sodass zufällige und gleichzeitig glatte und kohärente Strukturen entstehen. Fraktales Rauschen ist damit die algorithmische Grundlage vieler Landschafts- und Terrainsimulationen.

Ein weiterer Bereich der konstruktiven Generierung sind *formale Grammatiken*. Diese definieren formale Generierungsregeln, die beispielsweise aus Bausteinen Schritt für Schritt konkrete Strukturen erzeugen. Ein Beispiel für Verfahren dieser Kategorie sind *L-Systeme* [Pru86], die Regeln rekursiv anwenden, sodass komplexe hierarchische Muster entstehen, wie Pflanzen oder verzweigte Höhlensysteme. Formale Grammatiken eignen sich für Inhalte mit hierarchischen Strukturen, da sie eine explizite Beschreibung von Anordnung, Wachstum und Verzweigung ermöglichen.

Weitere konstruktive Verfahren basieren auf *zellulären Automaten* [Neu51]. Bei diesen Verfahren wird ein Raum in diskrete Zellen unterteilt, deren Zustand sich auf Basis lokaler Regeln iterativ ändert. Dies ermöglicht die Generierung von Strukturen wie Höhlen, Trennwände oder Vegetationsmuster.

### Suchbasierte Generierung

Suchbasierte Generierungsmethoden durchsuchen einen Lösungsraum und bewerten die erzeugte Inhalte anhand vorgegebener Kriterien. Beispiele dafür sind evolutionäre Algorithmen [BM10], constraint-basierte Ansätze [KS17] und RL [Kha+20]. Diese Verfahren optimieren oder explorieren Eigenschaften der generierten Inhalte und eignen sich somit besonders für Anwendungsfälle, in denen spezifische Lösungen gesucht werden, die bestimmte Anforderungen erfüllen.

Die Wahl eines geeigneten PCG-Ansatzes hängt vom gewünschten Verhältnis zwischen Variabilität und Kontrolle über das Generierungsergebnis ab. In dieser Arbeit dient die beschriebene Taxonomie als Grundlage zur Auswahl von Verfahren zur automatisierten Erzeugung von Lern- und Testumgebungen für AMR in der Intralogistik (vgl. Unterabschnitt 5.1.2).

## Kapitel 3

# Das Digitale Kontinuum und aktuelle Unterbrechungen

**Zusammenfassung** In diesem Kapitel wird das Digitale Kontinuum als Leitbild entwickelt, das die Konzepte der Industrie 4.0 und des Industrial Metaverse weiterentwickelt, um industrielle Effizienz und Resilienz durch KI-basierte Systeme auf höchster Autonomiestufe zu ermöglichen. Das Digitale Kontinuum wird anhand seiner drei zentralen Konzepte erläutert: dem Entwicklungszyklus als kontinuierlichem, KI-basiertem Selbstoptimierungsprozess, der drei Dimensionen der Kontinuität sowie der unterstützenden Schlüsseltechnologien. Für jede Phase des Entwicklungszyklus wird aufgezeigt, wie aktuelle Entwicklungen im Bereich der KI, insbesondere in KI-Agenten integrierte große Sprach- und multimodale Modelle, diesen Prozess unterstützen und automatisieren können. Abschließend werden die bestehenden Unterbrechungen identifiziert, die einer vollständigen Umsetzung des Digitalen Kontinuums derzeit entgegenstehen, und daraus der notwendige Erkenntnisgewinn abgeleitet, der Gegenstand dieser Arbeit ist.

### Der Begriff *Digitales Kontinuum* in der Literatur

Der Begriff Digitales Kontinuum wird in der Literatur in unterschiedlichen Kontexten genutzt, immer mit der Idee fließender Übergänge und integrierter Systeme.

- Arbeiten aus der organisationalen Transformation beschreiben das *Digitale Kontinuum* als fortlaufenden digitalen Wandel von Unternehmen und Methoden. Es geht dabei um kontinuierliche Anpassung von Prozessen, Kultur und Technologie und steht somit außerhalb des Kontextes dieser Arbeit. (z. B. [Per22; Sim25])
- Im Bereich des High-Performance-Computings wird der Begriff des *Digitalen Kontinuums* für die Bezeichnung einer kontinuierlichen Recheninfrastruktur von Edge- über Fog- bis zu Cloud-Ressourcen genutzt [Ant+19]. Damit beschreibt der Begriff in diesem Kontext nur einen Teil des Leitbilds des *Digitalen Kontinuums*, das in dieser Arbeit entwickelt wird.
- Mit dem Fokus auf die durchgängige, datenorientierte Vernetzung industrieller Systeme wurde das Digital 4.0 Continuum Reference Framework entwickelt [RE423]. Es hat das Ziel, Daten über den gesamten Lebenszyklus eines Produktes oder Systems konsistent verfügbar und verwertbar zu machen, um sie als eigenständiges Produkt zu vertreiben. Auch dieses Konzept beschreibt mit der durchgängigen Erfassung von Daten nur einen Teil des Leitbilds des *Digitalen Kontinuums*, das in dieser Arbeit entwickelt wird.
- In dem für diese Arbeit relevanten Kontext der autonomen Systeme in der Logistik wurde der Begriff *Digitales Kontinuum* konzeptionell erstmals durch Prof. Michael ten Hompel geprägt (Zukunftskongress 2019) und seither in einem Whitepaper beschrieben [Hom+25] und in Veröffentlichungen als Vision zugrunde gelegt (z. B. [Pri23]).

## 3.1 Vorwort zum Digitalen Kontinuum

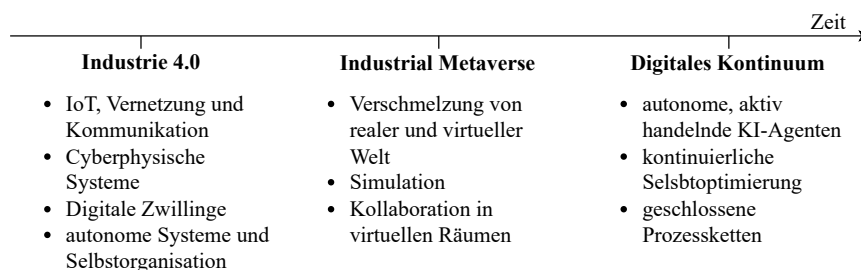
Das Konzept des Digitalen Kontinuums im Kontext autonomer logistischer Systeme wurde in den letzten Jahren mehrfach umrissen, jedoch bislang nicht systematisch ausgearbeitet. Die vorliegende Arbeit schließt diese Lücke mit der ersten konzeptuell kohärenten und empirisch untersuchten Beschreibung eines Leitbilds für die durchgängige Integration KI-basierter Systeme für die Entwicklung und den Betrieb autonomer Systeme in der Logistik. Aufbauend auf früheren Arbeiten, an deren Entwicklung die Autorin beteiligt war (vgl. [Hom+25]), wird in dieser Arbeit ein eigenständiger Bezugsrahmen ausgear-

beitet, der aktiv handelnde KI-Agenten, physische logistische Systeme und digitale Entwicklungsprozesse verbindet.

Im Mittelpunkt dieser Ausarbeitung steht die Annahme, dass digitale Systeme nicht nur reale Prozesse abbilden, sondern durch aktiv handelnde KI-Agenten selbst zu lernenden, antizipierenden und gestaltenden Akteuren werden. Das Digitale Kontinuum markiert somit einen Paradigmenwechsel. In der Erweiterung der rein abbildenden Perspektive Digitaler Zwillinge entsteht ein Leitbild, in dem Wahrnehmung, Entscheidungsfindung und physische Umsetzung eng miteinander verknüpft sind. Damit wird das Ziel verfolgt, eine neue Entwicklungsstufe industrieller Effizienz in der Entwicklung und in dem Betrieb logistischer Systeme zu erreichen.

## 3.2 Motivation und Hintergrund

Autonome Systeme werden in der Forschung und industriellen Praxis als Schlüsseltechnologie zur Steigerung von Effizienz, Flexibilität und Robustheit angesehen. Durch ihre Fähigkeit, Entscheidungen dezentral zu treffen und eigenständig auszuführen, besitzen sie das Potenzial, Logistikprozesse skalierbarer und adaptiver zu gestalten. Entsprechend sind Konzepte hochautonomer Systeme bereits in existierenden digitalen Leitbildern wie der Industrie 4.0 und dem Industrial Metaverse verankert. Ihre vollständige Umsetzung in der industriellen Realität ist jedoch bislang nicht erreicht (vgl. z. B. [Fot+21; Wah17]). Das in dieser Arbeit entwickelte Idealbild des Digitalen Kontinuums greift die Grundgedanken dieser Leitbilder auf, entwickelt sie konsequent weiter und bildet so den Rahmen für die nächste Entwicklungsstufe industrieller Autonomie. Abbildung 3.1 zeigt die Entwicklung digitaler Leitbilder mit ihren jeweiligen Grundideen.



**Abb. 3.1** Entwicklung digitaler Leitbilder von der Industrie 4.0 zum Digitalen Kontinuum

Die Industrie 4.0, die erstmals im Vorfeld der Hannover Messe 2011 vorgestellt wurde [KW22, S. 1], verfolgt die Vision industrieller Systeme, in denen intelligente Objekte durch Echtzeitkommunikation vernetzt sind, sowohl untereinander als auch mit ihren digitalen Abbildern. Diese Objekte sollen Prozesse eigenständig automatisieren und dabei in adaptiven Systemen

mit anderen Objekten und mit Menschen kooperieren. Zentrale technologische Konzepte sind dabei Internet of Things (IoT) und CPS (Unterunterabschnitt 2.2.5.1) und Digitale Zwillinge (Unterunterabschnitt 2.2.5.2).

Aufbauend auf den Zielen und Technologien der Industrie 4.0 sowie den Konzepten des Metaverse (siehe [Mys22]) entstand das Konzept des Industrial Metaverse. Dieses erweitert die reale industrielle Welt um einen digitalen Raum, in dem Menschen, Maschinen, Daten und Prozesse in virtuellen dreidimensionalen Umgebungen interagieren können. Durch die Verschmelzung realer und digitaler Elemente entsteht in dieser Vision eine neue Qualität der Simulation, Kollaboration und Steuerung, die die Konzepte von CPS und Digitalen Zwillingen erweitert [Laß22].

Trotz dieser Fortschritte bleibt die vollständige Umsetzung hochautonomer Systeme in der Industrie bislang aus. Vollständige Autonomie umfasst die eigenständige Wahrnehmung, Analyse, Planung, Entscheidungsfindung, Ausführung und Kommunikation eines Systems (Unterunterabschnitt 2.2.1.1). Eine Analyse von Fottner et al. zu Automatisierungsstufen intralogistischer Technologien und Prozesse zum Beispiel zeigt, dass die höchste Autonomiestufe 5 gemäß dem ALFUS-Modell bisher in keinem der bewerteten Aufgabenbereiche erreicht wird (Unterunterabschnitt 2.2.1.2) [Fot+21]. Insbesondere komplexe Aufgaben wie die eigenständige Planung weisen aktuell nur einen geringen Autonomiegrad auf. Operativ einfachere Aufgaben, wie die Ausführung eines Transportauftrags, können hingegen bereits Autonomiestufe 4 erreichen.

KI wird an dieser Stelle als eine entscheidende Technologie angesehen, Autonomiegrade auch in komplexeren Aufgaben und Situationen zu erhöhen [Wah17]. Um die vollen Potenziale KI-basierter Methoden für die Realisierung hochautonomer Systeme zu erreichen, fehlt es derzeit an Konzepten zur vollständigen Integration dieser Methoden sowohl in die Betriebs- als auch in die Entwicklungsprozesse autonomer Systeme.

An dieser Stelle setzt die Vision des Digitalen Kontinuums an. Das Digitale Kontinuum ist ein Leitbild, das das Ziel verfolgt, die industrielle Effizienz und Resilienz in hochautonomen Systemen zu steigern, indem es KI-Agenten als aktiv handelnde Entitäten vollständig integriert – sowohl in die Steuerung produktiver autonomer Systeme in der operativen Logistik als auch in die Steuerung ihrer Entwicklungs- und Adaptionprozesse.

Das Digitale Kontinuum beschreibt somit eine tiefgreifende Transformation, in der KI nicht mehr als isolierte Technologie, sondern als omnipräsente, integrierte und selbstständig handelnde Entität betrachtet wird und so kontinuierliche, geschlossene Prozessketten in der physischen Welt, in der digitalen Welt und in den Übergängen zwischen diesen Welten schafft.

Im Folgenden werden die grundlegenden Konzepte des Digitalen Kontinuums vorgestellt. In den darauf folgenden Abschnitten werden der Stand

der Technik erarbeitet und existierende Unterbrechungen im Digitalen Kontinuum identifiziert.

### 3.3 Kernkonzepte des Digitalen Kontinuums

In diesem Abschnitt werden die drei Kernkonzepte des Leitbilds *Digitales Kontinuum* vorgestellt, wie sie in Abbildung 3.2 dargestellt sind.

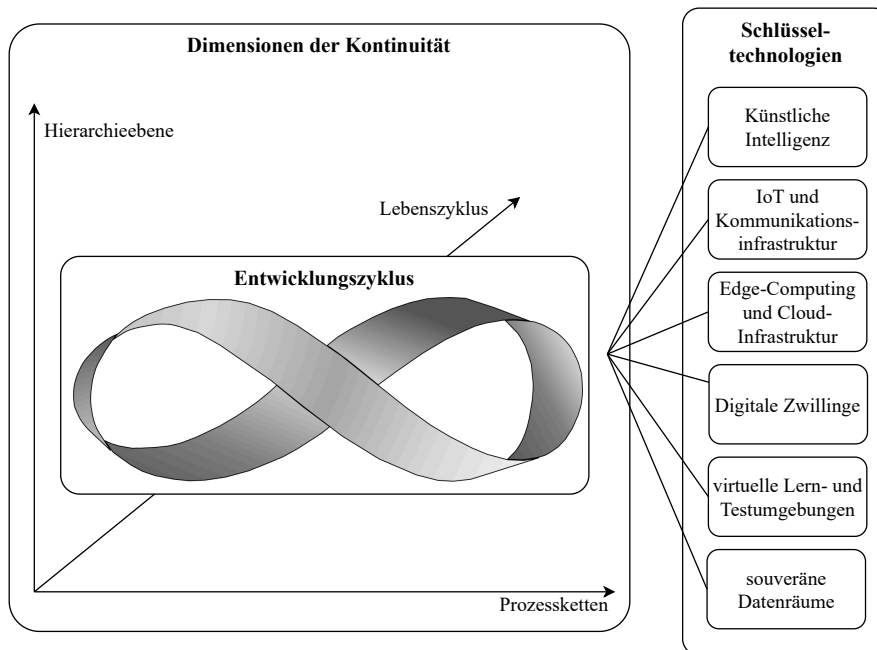


Abb. 3.2 Kernkonzepte des Digitalen Kontinuums

Im Mittelpunkt steht dabei der KI-basierte agentische *Entwicklungszyklus*, der den kontinuierlichen Selbstoptimierungsprozess autonomer Systeme in der Logistik im Zusammenspiel zwischen physischer und digitaler Realität beschreibt. Er orientiert sich an bestehenden Prinzipien für die Entwicklung und den Betrieb von KI-Systemen (siehe Unterabschnitt 2.3.3) und erweitert sie um die unterbrechungsfreie Integration ihrer Teilschritte. Dabei werden aktiv handelnde KI-Agenten in zwei Kontexten eingesetzt:

- Zum einen steuern sie autonome logistische Systeme im produktiven Betrieb in der physischen Realität. Diese KI-Agenten werden im Folgenden Steuerungsagenten des logistischen Systems genannt.
- Zum anderen lenken und führen sie den kontinuierlichen (Weiter-)Entwicklungsprozess der Steuerungsagenten des logistischen Systems aus, im Zusammenspiel zwischen physischer und digitaler Realität. Diese

KI-Agenten werden im Folgenden Steuerungsagenten des Entwicklungszyklus genannt.

Der Entwicklungszyklus bildet die Grundlage für drei *Dimensionen der Kontinuität*, die das Digitale Kontinuum auszeichnen: die horizontale Kontinuität, die die durchgängige Unterstützung über den gesamten Lebenszyklus eines Systems hinweg beschreibt, die vertikale Kontinuität, die den Übergang und die Konsistenz zwischen verschiedenen Hierarchieebenen innerhalb von logistischen Systemen ermöglicht, sowie die kausale Kontinuität, die die Durchgängigkeit von Prozessen abbildet.

Ein weiterer zentraler Bestandteil des Digitalen Kontinuums sind die *Schlüsseltechnologien*, die auf den Konzepten der Industrie 4.0 und des Industrial Metaverse aufbauen und im Digitalen Kontinuum im Sinne einer technologischen Konvergenz durch KI-Agenten zu einem integrierten Gesamtsystem verbunden werden.

Die folgenden Unterabschnitte beschreiben die Kernkonzepte des Digitalen Kontinuums näher und erläutern ihre Rolle innerhalb dieses Leitbilds.

### 3.3.1 KI-basierter Entwicklungszyklus

Im Zentrum des Digitalen Kontinuums steht der KI-basierte Entwicklungszyklus, der den Prozess integrierter, aktiv handelnder KI-basierter Technologien zwischen einem produktiven autonomen System in der Logistik und seiner digitalen Entwicklungsumgebung beschreibt. Durch die enge Integration aktiv handelnder KI-Technologien entsteht im Entwicklungszyklus ein fortlaufender Selbstoptimierungsprozess, der die Effizienz in Entwicklung und Betrieb logistischer Systeme steigert.

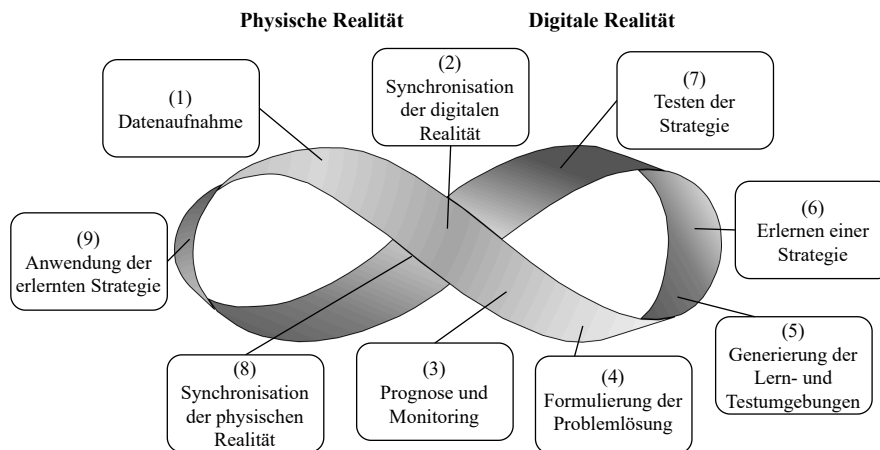


Abb. 3.3 Phasen im Entwicklungszyklus des Digitalen Kontinuums

Der Entwicklungszyklus lässt sich in neun Phasen einteilen (siehe Abbildung 3.3). In diesem Unterabschnitt werden die Abläufe innerhalb der Phasen dieses idealisierten, vollständig autonomen Zyklus beschrieben, der auf einen durchgängigen, nahtlosen Entwicklungs- und Betriebsprozess autonomer Systeme abzielt. Dabei werden exemplarisch aktuelle Entwicklungen im Bereich der KI benannt, die die einzelnen Phasen bereits heute unterstützen oder eine zukünftige Automatisierungspotenzial bieten.

Da ein produktives, operativ handelndes autonomes System im Kontext des Digitalen Kontinuums ein Roboter in einem physischen logistischen System oder ein steuerndes Softwaremodul auf einem Server sein kann, steht der Begriff *physische Realität* im Folgenden für die Umgebung, in der das autonome System logistische Prozesse im Produktivsystem steuert, und der Begriff *digitale Realität* für die digitale Entwicklungsumgebung des betrachteten autonomen Systems.

### 3.3.1.1 (1) Datenaufnahme

Die erste Phase des Entwicklungszyklus findet in der physischen Realität statt und verfolgt das Ziel, den Zustand des logistischen Systems in der physischen Realität zu erfassen. Dieser setzt sich aus dem internen Zustand des autonomen Systems selbst sowie aus Informationen über dessen Umgebung zusammen. Zur Umgebung zählen sowohl die unmittelbare physische Umwelt als auch benachbarte technische Systeme und Menschen, die relevante Informationen bereitstellen können.

Zur Wahrnehmung seiner selbst und der Umwelt nutzt das autonome System Sensoren (vgl. Abbildung 2.14). Sie erfassen strukturelle und dynamische Eigenschaften seiner Umgebung und sammeln zusätzlich Informationen über Interaktionen, zum Beispiel über Kommunikationsschnittstellen mit anderen autonomen Einheiten oder mit Menschen.

Neben den Zuständen können durch die Interaktionen auch Anforderungen und Zielvorgaben übermittelt werden, die das zukünftige Verhalten des betrachteten autonomen Systems beeinflussen. Diese können kurzfristige Anweisungen oder strategische, langfristige Ziele umfassen.

Durch den Einsatz von Edge-Computing und eingebetteter KI ist es dem autonomen System möglich, erste Vorverarbeitungsschritte bereits lokal durchzuführen. Beispielsweise können Bild- oder Sprachdaten analysiert, Muster in Maschinendaten erkannt oder irrelevante Informationen gefiltert werden, bevor die Daten an die nächste Verarbeitungsphase weitergeleitet werden (siehe z. B. [CR19; Hea24]).

Autonome Softwaresysteme greifen anstelle physischer Sensorik auf virtuelle Sensoren, Kommunikationsschnittstellen und Cloud-Ressourcen zu, um ihre Systemumgebung zu erfassen, mit anderen Instanzen zu interagieren und Informationen aufzubereiten.

Insgesamt stellt die Phase der Datenaufnahme die Grundlage für alle folgenden Prozesse im Entwicklungszyklus dar: Sie liefert Roh- und vorverarbeitete Sensordaten, Umgebungsinformationen sowie Zielvorgaben, die im weiteren Verlauf des Entwicklungszyklus analysiert und interpretiert werden.

### 3.3.1.2 (2) Synchronisation der digitalen Realität

Die zweite Phase des Entwicklungszyklus bildet den Übergang zwischen physischer und digitaler Realität. In dieser Phase nutzen KI-Agenten des Entwicklungszyklus die in Phase (1) erfassten Informationen, um das digitale Abbild, den Digitalen Zwilling, des autonomen Systems aus der physischen Realität zu synchronisieren. Über echtzeitfähige Kommunikationsnetzwerke und verteilte Recheninfrastrukturen werden die aktuellen Daten und Informationen aus der physischen Realität an den Digitalen Zwilling übermittelt, der daraufhin aktualisiert wird.

Durch den Einsatz von KI-Verfahren lässt sich dabei ein neues Maß an Genauigkeit bei der Abbildung der realen Welt im digitalen Modell erreichen. Dadurch wird der sogenannte Real-to-Sim-Gap reduziert und die Übertragbarkeit digitaler Entscheidungsmechanismen auf reale Systeme verbessert [Rue+20, S. 551]. Bei physischen autonomen Systemen bedeutet das beispielsweise, dass reale physikalische Effekte in die Abbildung der Robotersteuerung einbezogen werden, die durch Umweltbedingungen wie Temperatur, Reibung oder Abnutzung der Hardware in der physischen Realität beeinflusst werden [ZQW20, S. 739]. Durch Sensoren gemessene Daten aus der realen Umgebung ermöglichen es, diese Effekte mithilfe von KI zu modellieren und in die digital ermittelte Steuerungslogik zu integrieren.

Neben den physikalischen Eigenschaften aktualisieren KI-Agenten auch das digitale Struktur- und Verhaltensmodell auf Basis der aktuellen Sensordaten. Dieses Modell enthält sowohl strukturelle Eigenschaften der Umgebung als auch dynamische Informationen über das Verhalten anderer Akteure, wie andere autonome Einheiten oder Menschen. Für die strukturelle Aktualisierung können Methoden zur Sensorfusion eingesetzt werden, beispielsweise zur Erstellung von 3D-Umgebungsmodellen aus Kamera- und Lidar-Daten (siehe z. B. [Fab03], [Abd+21]). Das Verhalten anderer Entitäten lässt sich ebenso durch KI-basierte Verfahren wie neuronale Netze abbilden (siehe z. B. [Zhu+23], [Mir+23]), auch wenn insbesondere die Modellierung menschlicher Verhaltensweisen unter anderem aufgrund ihrer großen Variabilität und kulturellen Unterschiede weiterhin eine große Herausforderung darstellt [Zha+24].

Ebenfalls in dieser Phase erfolgt die Interpretation von Befehlen und Zielvorgaben durch KI-Agenten, die das System beispielsweise über natürliche Sprache in Phase (1) erhalten hat. KI-basierte Sprachverarbeitung ermöglicht

es, diese Informationen zu analysieren und strukturiert in Zielzustände zu überführen.

Darüber hinaus unterstützen KI-Verfahren aus dem Bereich der Cybersicherheit die Absicherung der Datenübertragung zwischen physischer und digitaler Realität. Sie helfen, potenzielle Angriffe frühzeitig zu erkennen und abzuwehren (siehe z. B. [SFN21]).

Das Ergebnis dieser Phase ist ein aktualisierter, konsistenter Digitaler Zwilling des autonomen Systems und seiner Umgebung in der physischen Realität, ergänzt um aktuelle Anforderungen und Zielvorgaben. Diese digitale Repräsentation bildet die Grundlage für die folgenden Phasen im Entwicklungszyklus.

### 3.3.1.3 (3) Prognose und Monitoring

In der dritten Phase des Entwicklungszyklus des Digitalen Kontinuums werden zukünftige Zustände des produktiven autonomen Systems sowie seiner Umgebung durch KI-Agenten des Entwicklungszyklus prognostiziert und mit den angestrebten Zielvorgaben und Anforderungen abgeglichen. Die Datenbasis für diese Prognosen setzt sich aus dem Digitalen Zwilling des Systems, einschließlich der historischen Zustandsdaten, sowie zusätzlichen Daten aus öffentlichen Quellen oder angrenzenden Systemen innerhalb souveräner Datenräume zusammen.

Souveräne Datenräume ermöglichen den sicheren, kontrollierten Austausch relevanter Informationen zwischen verschiedenen Systemen [FHM05]. Sie können dabei nicht nur aktuelle Zustände, sondern auch bereits berechnete Prognosen bereitstellen. So ermöglichen sie eine wechselseitige Abstimmung und Synchronisation zwischen logistischen Systemen und schaffen die Grundlage für koordinierte Entscheidungen.

KI-Verfahren unterstützen in dieser Phase beispielsweise die Vorhersagen über künftige Entwicklungen, etwa durch Zustandsprognosen der Umgebung oder durch Bedarfsprognosen zur Ermittlung bevorstehender Systemlasten (siehe z. B. [MDP22; NT25]).

Auf Grundlage dieser Vorhersagen entscheiden KI-Agenten des Entwicklungszyklus, ob ein Lernprozess angestoßen werden sollte, um die Steuerungsmodelle der KI-Agenten des logistischen Systems weiterzuentwickeln, und ob dazu eine Lernaufgabe neu formuliert werden muss. Die Auslösung eines Lernprozesses innerhalb eines Zyklus des Digitalen Kontinuums kann in verschiedenen Szenarien erforderlich sein (i. A. a. [Goo23]):

- *situativ*: Ein direkter Impuls in Phase (1), beispielsweise durch einen menschlichen Befehl oder ein erkanntes Ereignis, löst unmittelbar einen Lernprozess für eine bestehende Aufgabe aus.

- *zeitgesteuert*: Lernprozesse werden in regelmäßigen Intervallen angestoßen, beispielsweise wenn periodisch neue Trainingsdaten aus angrenzenden Systemen verfügbar sind.
- *bei Leistungsabfall*: Wenn ein Steuerungsmodell im produktiven Betrieb einen definierten Leistungswert unterschreitet, wird es neu trainiert. Auslöser können sich ändernde Rahmenbedingungen sein, wie ein bisher unbekannter Artikel im Kommissionierprozess.
- *bei veränderten Datenverteilungen*: Erhebliche Änderungen in den Eingangsdaten, etwa durch unerwartete Ereignisse in der Lieferkette, können darauf hinweisen, dass das bestehende Steuerungsmodell nicht mehr geeignet ist. In solchen Fällen kann neben dem Training auch eine Neukonzeption des Lösungsansatzes notwendig sein.
- *bei neuen Zielvorgaben*: Änderungen in den Zielvorgaben des Systems erfordern eine Anpassung der Bewertungsfunktionen. In der Folge müssen bestehende Steuerungsmodelle nachtrainiert oder neue Ansätze der Problemlösung konzipiert werden.
- *bei neuen Aufgaben*: Neue oder bisher nicht modellierte Aufgaben, etwa unbekannte Anforderungen oder unvorhergesehene Fehler, erfordern die Definition und Umsetzung neuer Ansätze der Problemlösung.

Das Ergebnis dieser Phase ist eine Prognose über die zukünftigen Zustände des physischen autonomen Systems und seiner Umgebung sowie die Entscheidung, ob ein Lernprozess eingeleitet werden muss und ob dazu die Notwendigkeit besteht, neue Ansätze der Problemlösung zu formulieren oder bestehende Modelle nachtrainiert werden können.

#### 3.3.1.4 (4) Formulierung der Problemlösung

In den Fällen, in denen neue Zielvorgaben, Aufgaben oder veränderte Datenverteilungen die bestehenden Steuerungsmodelle eines produktiven autonomen Systems nicht mehr ausreichend gültig oder leistungsfähig machen, muss ein neuer Ansatz zur Problemlösung formuliert werden. In dieser vierten Phase des Entwicklungszyklus definieren KI-Agenten daher, welches Entscheidungsproblem gelöst werden soll und mit welchem Mechanismus das geschehen soll. Dabei kann es sich um algorithmische Optimierungsverfahren, regelbasierte Ansätze oder um das Erlernen neuer Verhaltensstrategien mithilfe von Methoden des ML handeln.

Aktuelle Entwicklungen im Bereich großer KI-Modelle eröffnen die Möglichkeit, diesen Schritt zu automatisieren. Neue generative Modelle verfügen über Fähigkeiten zum Reasoning und damit zum Ableiten neuer Problemlösungen (siehe Abschnitt 2.3). Erste Arbeiten, wie die von Xian et al., zeigen, dass große Sprachmodelle in der Lage sind, aus einer abstrakten Aufgabendefinition ausführbaren Programmcode für robotische Lernprozesse zu erzeugen [Xia+23]. Zugleich besteht, wie Romera-Paredes et al. in [Rom+24]

diskutieren, die begründete Hoffnung, dass KI-gestützte Verfahren perspektivisch nicht nur bekannte Lösungsansätze auswählen, sondern eigenständig neue algorithmische Strategien entwickeln können, die bisher menschlicher Forschung vorbehalten waren.

Das Ergebnis der vierten Phase des Entwicklungszyklus sind formal spezifizierte Problembeschreibungen und Lösungsansätze, die entweder direkt ausführbar sind oder in ein Trainingsverfahren überführt werden können. Wenn für die Problemlösung ein lernbasierter Ansatz gewählt wird, entsteht die Anforderung, geeignete Lern- und Testumgebungen bereitzustellen, in denen Steuerungsagenten des logistischen Systems basierend auf Erfahrungen lernen und die erlernte Strategie validieren kann.

Die vierte Phase bildet damit die konzeptionelle Grundlage für den praktischen Lernprozess und leitet zur nächsten Phase über, in der die automatisierte Generierung realistischer und variantenreicher Lern- und Testumgebungen erfolgt. Die Ausgestaltung dieser fünften Phase stellt einen zentralen Schwerpunkt dieser Arbeit dar.

#### **3.3.1.5 (5) Generierung der Lern- und Testumgebungen**

In der fünften Phase des Entwicklungszyklus stoßen Steuerungsagenten des Entwicklungszyklus die automatisierte Generierung virtueller Lern- und Testumgebungen in der digitalen Realität für den zuvor definierten Lernprozess an.

Das Ziel dieser Phase ist es, eine möglichst große Vielfalt an Umgebungsvarianten zu erzeugen, um die Generalisierungsfähigkeit der KI-Modelle der Steuerungsagenten des logistischen Systems zu verbessern. Durch den Variantenreichtum steigt die Wahrscheinlichkeit, dass die erlernten Strategien auch in der physischen Realität zuverlässig funktionieren. Zur Erreichung dieses Ziels kommen beispielsweise Verfahren der PCG zum Einsatz (siehe Abschnitt 2.5), die Inhalte algorithmisch erzeugen. Die Informationsbasis zur Generierung der Modelle stammt zudem aus dem aktualisierten Digitalen Zwilling des autonomen Systems und seiner Umgebung.

Auch in dieser Phase können KI-Technologien den Entwicklungszyklus des Digitalen Kontinuums unterstützen. Je nach konkreter Lernaufgabe können neben KI-basierten Verfahren der PCG zusätzlich Reasoning-Fähigkeiten großer Sprachmodelle genutzt werden, um automatisiert Programmcode zur Umgebungsmodellierung zu generieren [Xia+23]. Zudem werden Text-to-Image-Modelle untersucht, die auf Basis textueller Beschreibungen realitätsnahe und vielfältige Umgebungen visuell darstellen und potenziell in simulationsbasierte Lernprozesse überführt werden können [Yu+23]. Eine Diskussion der Generierungsansätze für den spezifischen Anwendungsfall von AMR in der Intralogistik, der der Fokus dieser Arbeit ist, findet sich in Kapitel 4.

Das Ergebnis der fünften Phase des Entwicklungszyklus ist eine große Bandbreite an unterschiedlichen virtuellen Lern- und Testumgebungen. Sie enthalten alle relevanten Strukturen, dynamischen Prozesse und Bedingungen, die notwendig sind, um in der nächsten Phase eine neue Strategie zur Steuerung der KI-Agenten des logistischen Systems erfolgreich zu erlernen.

#### **3.3.1.6 (6) Erlernen einer Strategie**

In der sechsten Phase des Entwicklungszyklus des Digitalen Kontinuums erfolgt der eigentliche Lernprozess zur Entwicklung einer neuen Steuerungsstrategie für die Steuerungsagenten des logistischen Systems (siehe Unterunterabschnitt 2.3.2.1). Dafür wird der in Phase vier generierte Programmcode, der die zu lösende Lernaufgabe beschreibt, in den in Phase fünf modellierten Lernumgebungen ausgeführt. Das Ziel dieser Phase ist es, ein KI-Modell zu trainieren, das in der Lage ist, situationsabhängige und robuste Entscheidungen zu treffen und damit das gewünschte Systemverhalten in der physischen Realität zu erzeugen.

Das Ergebnis dieser Phase ist ein trainiertes KI-Modell, das in der folgenden Phase auf seine Leistungsfähigkeit in bisher ungesehenen Testumgebungen überprüft wird.

#### **3.3.1.7 (7) Testen der Strategie**

In der siebten Phase des Entwicklungszyklus wird das trainierte KI-Modell in virtuellen Testszenarien evaluiert. Dabei kommen die in Phase fünf generierten Testumgebungen zum Einsatz, um die erlernte Steuerungsstrategie unter bisher unbekanntem, aber realistischen Bedingungen anzuwenden. Eine breite Variation der Parameter in den Testumgebungen ist entscheidend, um die Robustheit der Strategie zu überprüfen und sicherzustellen, dass sie auch in seltenen oder kritischen Situationen wünschenswertes Systemverhalten erzeugt.

Die Phasen vier bis sieben des Entwicklungszyklus können iterativ durchlaufen werden, sollte die Leistung der erlernten Steuerungsstrategie in den Testszenarien nicht erreicht werden. In diesem Fall formulieren KI-Agenten des Entwicklungsprozesses in Phase vier einen alternativen Lösungsansatz und stoßen einen neuen Lernprozess an.

Das Ergebnis dieser Phase ist ein validiertes KI-Modell, das die Steuerungsstrategie der Agenten des logistischen Systems in der physischen Realität bestimmt und zuvor unter vielfältigen Bedingungen erfolgreich getestet wurde.

#### **3.3.1.8 (8) Synchronisation der physischen Realität**

Die achte Phase des Entwicklungszyklus des Digitalen Kontinuums umfasst den Übergang von der digitalen in die physische Realität. In dieser Phase

werden die in der digitalen Realität entwickelten Artefakte, insbesondere das trainierte KI-Modell zur Steuerung des produktiven logistischen Systems, für den Einsatz in der realen Umgebung aufbereitet und übertragen.

Die Integration erfolgt automatisiert. Dabei werden die Datenpakete, die die Steuerungsstrategie enthalten, verpackt und über Kommunikationsnetzwerke und geeignete Recheninfrastrukturen an das autonome System in der physischen Realität übermittelt.

Das Ergebnis dieser Phase ist ein aktualisiertes, handlungsfähiges autonomes System in der physischen Realität, das mit einer neu trainierten Steuerungsstrategie ausgestattet und bereit für den Einsatz in der realen Umgebung ist.

#### **3.3.1.9 (9) Anwendung der erlernten Strategie**

In der neunten Phase des Entwicklungszyklus setzen die Steuerungsagenten des logistischen Systems die in der digitalen Realität erlernte Steuerungsstrategie in der physischen Realität um. Dies kann entweder in einem eingebetteten physischen System oder als Softwareagent innerhalb einer Cloud-Infrastruktur geschehen.

Die Agenten nutzen dabei ihre Sensoren, um ihre Umgebung wahrzunehmen, verarbeiten die gesammelten Informationen über Edge- oder Cloud-Recheneinheiten, wählen anhand der erlernten Strategie geeignete Handlungen aus und führen diese mithilfe von Aktoren aus. Über Kommunikationssysteme sind sie mit anderen autonomen Agenten sowie mit Menschen im Gesamtsystem verbunden (siehe Unterabschnitt 2.2.4).

Große Sprachmodelle ermöglichen eine Kommunikation in natürlicher Sprache. Dadurch können autonome Agenten nicht nur mit menschlichen Akteuren interagieren, sondern auch Wissen, das zuvor in der digitalen Realität erlernt wurde, an andere Agenten weitergeben und somit zur Systemoptimierung beitragen [RP24].

Durch ihre kontinuierliche Interaktion mit der Umwelt und den weiteren Systemteilnehmenden erfassen die autonomen Systeme laufend neue Informationen zum aktuellen Systemzustand. Diese Daten bilden den Ausgangspunkt für die erste Phase des Entwicklungszyklus und leiten einen neuen Durchgang des Entwicklungszyklus ein.

Der Entwicklungszyklus des Digitalen Kontinuums ermöglicht es auf diese Art und Weise, vollständig automatisiert neue KI-basierte Steuerungsstrategien für autonome Systeme in der Logistik zu entwickeln. Der Zyklus wird dabei selbst durch aktiv handelnde KI-Technologien angetrieben. Dabei werden insbesondere große Sprach- und Multimodalmodelle eingesetzt: Eingebettet in KI-Agenten werden diese Modelle zu aktiv gestaltenden Akteuren und Treibern in den Entwicklungs- und Betriebsprozessen autonomer

Systeme. Es ergibt sich ein beschleunigter, selbstoptimierender Regelkreis für hochautonome Systeme in der Logistik, vom Roboter bis zur Lieferkette.

### 3.3.2 Dimensionen der Kontinuität

Der Entwicklungszyklus des Digitalen Kontinuums ist durch nahtlose Übergänge geprägt, die sich in verschiedenen Dimensionen ausprägen. Diese Kontinuitäten lassen sich in drei Kategorien einteilen: horizontal, vertikal und kausal. Sie bilden eines der Kernelemente des Digitalen Kontinuums und werden im folgenden Abschnitt erläutert.

#### 3.3.2.1 Horizontale Kontinuität

In Anlehnung an die Kontinuitätsdimensionen des MBSE (siehe Abschnitt 2.4) beschreibt die horizontale Kontinuität im Digitalen Kontinuum die fortlaufende Anpassung und Weiterentwicklung entlang des gesamten Lebenszyklus eines autonomen Systems in der Logistik.

Dies bedeutet, dass der Entwicklungszyklus des Digitalen Kontinuums bereits in frühen Entwicklungsphasen des autonomen Systems beginnt und nahtlos in den aktiven Betrieb übergeht. Entwicklungs- und Betriebsphasen des Systems sind dabei nicht mehr klar voneinander getrennt, sondern gehen in einem kontinuierlichen Optimierungsprozess ineinander über. Die digitale Realität entwickelt sich dabei stets vorauslaufend weiter und liefert die Impulse für die nächste Stufe in der physischen Realität.

#### 3.3.2.2 Vertikale Kontinuität

Ebenfalls in Anlehnung an die Kontinuitätsdimensionen des MBSE beschreibt die vertikale Kontinuität im Digitalen Kontinuum die Fähigkeit, kontinuierlich zwischen verschiedenen Hierarchieebenen autonomer Systeme in der Logistik zu wechseln. Dies umfasst insbesondere die Möglichkeit, den Abstraktionsgrad und den Detaillierungsgrad eines betrachteten Systems flexibel und konsistent anzupassen. Für logistische Systeme bedeutet dies, dass sämtliche Ebenen vom einzelnen technischen Element bis hin zur gesamten Lieferkette skalenübergreifend abgebildet werden können (siehe Abbildung 2.3).

#### 3.3.2.3 Kausale Kontinuität

Die kausale Kontinuität beschreibt im Digitalen Kontinuum die Eigenschaft, dass Prozessketten kontinuierlich, also vollständig geschlossen, sind. Der Begriff *kausal* verweist darauf, dass KI-Agenten durch ihre Fähigkeit, logische Verknüpfungen zu ziehen, in der Lage sind, kausale Verbindungen zwischen einzelnen Prozessschritten herzustellen. Dadurch können sie Ursache-Wirkungs-Beziehungen erkennen und gezielt nutzen, um Abläufe zu optimieren. Dies gilt sowohl für die Abläufe in der physischen Realität logistischer

Systeme als auch für die Prozesse im Entwicklungszyklus des Digitalen Kontinuums.

In der physischen Realität bedeutet dies, dass logistische Prozessketten geschlossen werden, indem autonome Systeme mithilfe der Steuerungsagenten des logistischen Systems die Fähigkeit erlangen, sich selbst zu optimieren und zu steuern und beispielsweise Fehler eigenständig zu beheben und mit unbekanntem Situationen umzugehen. Dadurch können Unterbrechungen in Materialflussprozessen reduziert werden.

Im Entwicklungszyklus des Digitalen Kontinuums betrifft die kausale Kontinuität insbesondere die Prozessketten, die die Entwicklung und den Betrieb autonomer Systeme verbinden. Diese Prozesse, die zwischen physischer und virtueller Welt stattfinden, werden durch den Einsatz autonomer KI-Agenten koordiniert und nahtlos miteinander verknüpft.

Die kontinuierlichen Übergänge in den Dimensionen des Digitalen Kontinuums stellen eines seiner zentralen Merkmale dar. Diese Kontinuitäten werden durch die koordinierte Integration zentraler Schlüsseltechnologien realisiert, die durchgängige Daten-, Entscheidungs- und Wirkprozesse unterstützen.

### 3.3.3 Schlüsseltechnologien

Die Schlüsseltechnologien des Digitalen Kontinuums bilden das Fundament, auf dem die Vision nahtlos integrierter, hochautonomer Systeme umgesetzt werden kann. Sie ermöglichen es, physische und digitale Welten zu verbinden, Daten effizient und skalenübergreifend zu erfassen und zu verarbeiten, Entscheidungen auf allen Hierarchieebenen logistischer Systeme zu unterstützen und in Handlungen umzusetzen.

Durch die nahtlose Integration der Schlüsseltechnologien entsteht dabei eine technologische Konvergenz (siehe [Wir18]). Im Digitalen Kontinuum bedeutet technologische Konvergenz, dass ehemals getrennte Technologien zu einem integrierten Ganzen verschmelzen, insbesondere durch die Integration von KI-Agenten. Daten und Prozesse werden durchgängig verbunden, von der physischen Realität über Digitale Zwillinge bis hin zu intelligenten Entscheidungsmechanismen. Konvergenz beschreibt damit ein Angleichungs- und Zusammenwachsensphänomen, das durch wechselseitige Beeinflussung und gemeinsame Rahmenbedingungen entsteht. Die Technologien arbeiten nicht mehr isoliert, sondern bilden ein vernetztes Ökosystem.

#### 3.3.3.1 Künstliche Intelligenz als Treiber

KI ist die zentrale Antriebskraft des Digitalen Kontinuums. Sie befähigt autonome Systeme, komplexe Zusammenhänge zu analysieren, Prognosen zu erstellen, Strategien zu optimieren und selbstständig situationsabhängige Entscheidungen zu treffen. Eingebettet in KI-Agenten übernimmt sie

sowohl Steuerungsaufgaben in autonomen Systemen in der physischen Realität als auch die Steuerung des Entwicklungszyklus zur kontinuierlichen Verbesserung und Weiterentwicklung und integriert die folgenden Schlüsseltechnologien.

#### **3.3.3.2 IoT und Kommunikationsinfrastruktur**

IoT-Technologien und Kommunikationsnetze bilden die sensorische und kommunikative Basis des Digitalen Kontinuums. IoT-Geräte erfassen kontinuierlich Daten aus der physischen Realität, während 5G-, 6G- und andere hochperformante Netzwerke den Echtzeit-Datenaustausch innerhalb und zwischen Systemen und ihren digitalen Entwicklungsumgebungen ermöglichen. Diese Infrastruktur stellt sicher, dass Informationen an die relevanten Entscheidungsmechanismen im Digitalen Kontinuums gelangen.

#### **3.3.3.3 Edge- und Cloud-Computing**

Die Kombination von Edge- und Cloud-Computing ermöglicht eine skalierbare und flexible Datenverarbeitung im Digitalen Kontinuum. Edge-Computing bringt Rechenleistung nahe an die Datenquelle, um Latenzen zu minimieren und kurzfristige Reaktionen zu ermöglichen. Cloud-Computing stellt zentrale Ressourcen für rechenintensive Analysen, Modelltrainings und die Orchestrierung komplexer Prozesse bereit. Zusammen unterstützen sie sowohl zeitkritische lokale Entscheidungen als auch umfassende strategische Analysen.

#### **3.3.3.4 Digitale Zwillinge**

Digitale Zwillinge sind die virtuellen Abbilder physischer Systeme, die in Echtzeit synchronisiert werden (siehe Unterunterabschnitt 2.2.5.2). Im Digitalen Kontinuum müssen diese digitalen Abbilder skalenübergreifend gestaltet sein, sodass sie sich dynamisch abstrahieren oder detaillieren lassen, um verschiedene Hierarchieebenen logistischer Systeme im Sinne der vertikalen Kontinuität abzubilden.

#### **3.3.3.5 Virtuelle Lern- und Testumgebungen**

Virtuelle Umgebungen dienen der sicheren und kosteneffizienten Entwicklung, Erprobung und Optimierung neuer Steuerungsstrategien. Sie ermöglichen umfangreiche Tests unter variierenden Bedingungen, bevor diese Strategien in die physische Realität übertragen werden. Im Digitalen Kontinuum sind sie eng mit den Digitalen Zwillingen verbunden und nutzen Methoden wie PCG, um eine hohe Generalisierungsfähigkeit der gelernten Modelle zu gewährleisten.

### 3.3.3.6 Souveräne Datenräume

Souveräne Datenräume regeln, welches System auf welche Daten in welchem Detailgrad zugreifen darf. Sie stellen die Einhaltung der Datenhoheit und des Datenschutzes sicher, während sie gleichzeitig einen sicheren Informationsaustausch zwischen Organisationen ermöglichen. In der digitalen Realität sind diese Datenräume mit skalenübergreifenden Digitalen Zwillingen verknüpft, sodass lokale Steuerungsaufgaben und globale Strategien durchgängig miteinander verbunden werden können. In Kombination mit Smart Contracts ermöglichen sie es, auch unternehmensübergreifende Prozesse ununterbrochen durch autonome KI-Agenten zu steuern.

Der KI-basierte Entwicklungszyklus, die Dimensionen der Kontinuität und die zugehörigen Schlüsseltechnologien bilden die zentralen Bestandteile des Idealbilds des Digitalen Kontinuums. In diesem Leitbild ermöglichen aktiv handelnde KI-Agenten die Realisierung hochautonomer, selbstoptimierender logistischer Systeme und damit die Erschließung neuer Effizienzen in der Logistik. Um das Digitale Kontinuum in der Praxis zu verwirklichen, gilt es, bestehende Unterbrechungen entlang dieses Entwicklungs- und Betriebsprozesses zu identifizieren und sukzessive zu schließen. Die systematische Identifizierung dieser Unterbrechungen steht im Fokus des folgenden Abschnitts.

## 3.4 Existierende Unterbrechungen im Digitalen Kontinuum

Aktuelle Praktiken im Bereich der Entwicklung und des Betriebs industrieller autonomer Systeme zeigen eine Reihe an Unterbrechungen in Bezug auf die im Digitalen Kontinuum angestrebten Prozesse. Die Unterbrechungen lassen sich entlang der Dimensionen der Kontinuität im Digitalen Kontinuum einordnen: der horizontalen, der vertikalen und der kausalen Kontinuität.

### 3.4.1 Unterbrechungen in der horizontalen Kontinuität

Die Unterbrechungen der horizontalen Kontinuität des Digitalen Kontinuums ergeben sich im Verlauf des Lebenszyklus eines autonomen logistischen Systems. Die Inbetriebnahme der Systeme ist hier in aktuellen Vorgehensweisen als die wesentliche Unterbrechung zu nennen.

- *Inbetriebnahme*: Die Inbetriebnahme autonomer Systeme ist aktuell ein ressourcenintensiver Prozess, der das Digitale Kontinuum zwischen Entwicklung und Betrieb unterbricht. Für AMR erfolgt während der Inbetriebnahme beispielsweise die Einrichtung der Steuerungssoftware, die unter anderem die manuelle Anpassung von Robotermodellen und Parametern erfordert [Col+14]. Hinzu kommen typischerweise die

manuelle Erstellung von Karten der Einsatzumgebung, das Beibringen von Routen und Übergabepunkten, die Simulation und Validierung von Einsatzszenarien sowie die iterative Optimierung des Layouts und der Steuerungslogik [Sol20]. Für digital agierende Steuerungsagenten des logistischen Systems erfolgen während der Inbetriebnahme die manuelle Anbindung und Konfiguration von Live-Datenbanken. Insgesamt ist der Übergang von der digitalen Entwicklung in den realen Betrieb häufig mit zeitintensiven, nicht automatisierten Schritten verbunden.

### 3.4.2 Unterbrechungen in der vertikalen Kontinuität

Die Unterbrechungen der vertikalen Kontinuität zeigen die Brüche zwischen den verschiedenen Hierarchieebenen logistischer Systeme im Digitalen Kontinuum auf. Sie entstehen, wenn Informationen, Modelle oder physische Objekte nicht nahtlos von einer Systemebene zur nächsten übertragen werden können. Diese Unterbrechungen betreffen insbesondere folgende Aspekte:

- *Datenverfügbarkeit:* Zwischen den verschiedenen Hierarchieebenen logistischer Systeme bestehen häufig Unterbrechungen im Informationsfluss, die auf heterogene Datenformate, unterschiedliche Datenverfügbarkeit und Bedenken hinsichtlich der Datensicherheit zurückzuführen sind. Unterschiedliche Systeme entlang der Hierarchieebenen logistischer Systeme (siehe Abbildung 2.3) verwenden proprietäre Schnittstellen oder inkompatible Datenstrukturen, was die direkte Weitergabe und Verarbeitung von Informationen erschwert. Hinzu kommt, dass sicherheits- oder datenschutzrelevante Informationen oft nur eingeschränkt zugänglich sind [Hei+25]. Obwohl der Einsatz von standardisierten, föderierten Datenräumen hier einen Lösungsansatz bietet, steht eine umfangreiche Realisierung dieser Technologie noch aus.
- *Vertragliche Regelungen:* Zwischen verschiedenen Hierarchieebenen, insbesondere zwischen unterschiedlichen Organisationen innerhalb einer Lieferkette, bestehen oft vertragliche Vorgaben, die die Material-, Informations- und Finanzflüsse zwischen den Akteuren eines logistischen Systems regeln. Diese müssen heute meist manuell überprüft und umgesetzt werden, was zu Verzögerungen und Administrationsaufwand führt. Der Einsatz von Smart Contracts auf Basis verteilter Ledger-Technologien kann diese Prozesse automatisieren, doch sind diese Technologien heute wenig realisiert.

### 3.4.3 Unterbrechungen in der kausalen Kontinuität

Die Unterbrechungen in der kausalen Kontinuität im Digitalen Kontinuum zur aktuellen Zeit können in Unterbrechungen in physischen und in digitalen Prozessen sowie in Prozessen zwischen diesen Welten unterteilt werden.

### 3.4.3.1 Unterbrechungen in physischen Prozessen

Physische Prozesse in logistischen Systemen können durch verschiedene Faktoren unterbrochen werden, die auf Einschränkungen aktuell industriell eingesetzter Automatisierungs- und KI-Technologien zurückzuführen sind.

- *Komplexität der Umgebung*: Autonome Systeme stoßen aktuell in hochdynamischen, heterogenen Einsatzumgebungen, wie sie in der Logistik beispielsweise in Umschlagszentren zu finden sind, noch an Grenzen ihrer adaptiven Leistungsfähigkeit. Während Menschen flexibel auf unvorhergesehene Ereignisse reagieren und Störungen situativ beheben können, fehlen aktuellen autonomen Systemen häufig noch robuste Mechanismen zur Umwelt- und Situationsanpassung [AKD25]. Ein Beispiel hierfür ist die Notwendigkeit der Auflösung von Störfällen durch menschliche Eingriffe.
- *Sensomotorische, emotionale und soziale Intelligenz*: Trotz der Fortschritte im Bereich großer Sprach- und Multimodalmodelle sind KI-basierte autonome Systeme heute noch in ihrer sensomotorischen, emotionalen und sozialen Intelligenz eingeschränkt [Wah17, S. 411]. Das begrenzt ihren Einsatz in Prozessen, die ein feinfühliges, situationsabhängiges Handeln oder eine ausgeprägte Mensch–Maschine–Interaktion erfordern.
- *Alignment-Problem*: Unzureichend definierte Zielvorgaben und Nebenbedingungen oder seltene Randfälle für KI-Systeme können zu Handlungen führen, die nicht mit den für das KI-System beabsichtigten Zielen übereinstimmen. Das sogenannte Alignment-Problem ist eine Quelle für Fehlentscheidungen und kann bei Eintreten unerwarteter Systemzustände in der Realität zu Prozessunterbrechungen führen.

### 3.4.3.2 Unterbrechungen in digitalen Prozessen

Unterbrechungen in den digitalen Prozessen des Digitalen Kontinuums entstehen zum aktuellen Stand der Technik zum einen durch Unterbrechungen in aktuellen Entwicklungsprozessen und zum anderen durch die digitale Repräsentation der physischen Welt.

- *Formulierung der Problemlösung*: Unterbrechungen im Entwicklungsprozess treten aktuell vor allem in der Konzeptionierungs- und Experimentierphase einer KI-Lösung für ein autonomes System auf (vgl. Unterabschnitt 2.3.3). Sie zeigen sich bei der Planung der ML-Anwendung, der Formulierung von Anforderungen und deren Zerlegung in lösbarere Teilprobleme sowie bei der Auswahl geeigneter Methoden und Verfahren. Auch im Model Engineering, von den ersten Experimenten bis hin zur Entwicklung eines passenden ML-Ansatzes einschließlich Trainingscode für die spezifische Anwendung, lassen sich Brüche beobachten. Weitere Unterbrechungen entstehen zudem beim manuellen Aufbau automatisierter Pipelines.

- *Generierung von Lern- und Testumgebungen*: Die Erzeugung realitätsnaher Lern- und Testumgebungen für autonome Systeme ist heute häufig mit hohem manuellem Aufwand verbunden. Zudem limitieren unzureichende Abbildungsgenauigkeit, eingeschränkte Systemgrenzen und fehlende Berücksichtigung dynamischer Umgebungsaspekte die Nutzbarkeit digitaler Lern- und Testszenarien. Eine ausführliche Betrachtung dieser Unterbrechung erfolgt in Kapitel 4.

### 3.4.3.3 Unterbrechungen in Prozessen zwischen der digitalen und der physischen Realität

- *Sim-to-Real-Gap und Generalisierungsfähigkeit*: Die Simulation von Robotern wie AMR ermöglicht eine signifikante Reduktion von Entwicklungszeiten und -kosten, da Anpassungen an Hardware und Verhalten zunächst digital entworfen, getestet und optimiert werden können, bevor sie in der physischen Realität umgesetzt werden. Dieses Potenzial kann jedoch nur ausgeschöpft werden, wenn die in der Simulation gewonnenen Erkenntnisse zuverlässig auf das reale, physische System übertragbar sind. In der Praxis zeigt sich häufig ein Leistungseinbruch bei der Übertragung, beispielsweise in Form einer geringeren Erfolgsrate. Zudem können Steuerungsparameter, die in der Simulation funktionieren, in der Realität nicht oder nur eingeschränkt anwendbar sein und unerwartete Effekte hervorrufen [Mur+22, S. 1].
- *Digitale Modelle aktuell halten*: Digitale Zwillinge und die darauf basierenden Simulationsmodelle sind entscheidend für die Entwicklung und Anpassung autonomer Systeme an dynamische Einsatzbedingungen. Derzeit werden diese digitalen Modelle in vielen Fällen einmalig im Rahmen der Planungs- und Entwicklungsphase erstellt. Änderungen am physischen System erfordern aufwändige, meist manuelle Anpassungen im digitalen Abbild. Dies führt dazu, dass die Synchronizität zwischen physischer und digitaler Welt oft nicht gewährleistet ist.
- *Abbildung menschlichen Verhaltens und Intentionen*: Die digitale Modellierung menschlicher Verhaltensweisen und Absichten stellt weiterhin eine hochkomplexe Aufgabe dar. Unterschiede in kulturellen, sozialen und demografischen Kontexten erschweren die Entwicklung generalisierbarer Modelle und erzeugen Unsicherheiten bei der Integration in Mensch-Maschine-Umgebungen [Zha+24]. Forschungen zur generativen Modellierung menschlicher Bewegung, wie MotionDiffuse [Zha+24] oder die Methode von Mir et al. [Mir+23], zeigen zwar Fortschritte bei der Animation realistischer Bewegungsabläufe, bilden jedoch keine logistischen Bewegungsmuster ab und sind daher nur eingeschränkt auf intralogistische Szenarien übertragbar.

### 3.5 Fazit zum Digitalen Kontinuum und seinen Unterbrechungen

Das Digitale Kontinuum beschreibt einen durchgängigen, KI-basierten Entwicklungs- und Betriebsrahmen für autonome Systeme in der Logistik, in dem aktiv handelnde KI-Agenten die physische und die digitale Realität eng miteinander koppeln. Das Ziel dabei ist die Erschließung neuer Potenziale industrieller Effizienz durch unterbrechungsfreie, beschleunigte Entwicklungs- und Betriebsprozesse. Das entwickelte Leitbild vereint den kontinuierlichen Entwicklungszyklus, die Dimensionen der Kontinuität und die Schlüsseltechnologien zu einer ganzheitlichen Vision für die Entwicklung und den Betrieb hochautonomer logistischer Systeme.

Im Rahmen der vorliegenden Arbeit steht somit nicht nur die direkte Echtzeitregelung autonomer Systeme durch KI-Methoden wie neuronale Netze im Vordergrund. Die Ausarbeitung zielt vielmehr auf der Entwicklung ganzheitlicher Steuerungsverfahren ab, die die operative Kontrolle logistischer Systeme übernehmen. KI wird damit nicht primär als online-regelnde Instanz, sondern als strategisch generierendes System verstanden, das in der Lage ist, Steuerlogiken zu entwerfen, zu testen und in dynamischen Umgebungen anzuwenden.

KI-Agenten im Sinne des Digitalen Kontinuums agieren daher nicht nur reaktiv innerhalb vorgegebener Handlungsräume, sondern entwickeln im Zuge simulationsgestützter Prozesse eigenständig neue Regelwerke und Verhaltensstrategien und bieten somit eine Form der algorithmischen Selbstentwicklung, die in hochdynamischen logistischen Kontexten als zukunftsweisend gilt. In diesem Leitbild übernehmen KI-Agenten in der Logistik zunehmend Aufgaben, die bisher dem Menschen vorbehalten waren. Durch das Ermöglichen einer steigenden Autonomie und damit Unabhängigkeit vom Menschen spielt das Digitale Kontinuum eine wesentliche Rolle in der Zukunftsvision posthumaner Logistik [HKR25]. In dieser findet eine Neuausrichtung logistischer Systeme an einer Realität weitgehend ohne menschliche Beteiligung statt, um Entwicklungen wie den bestehenden Arbeitskräftemangel und demografischen Wandel durch technische Fortschritte wie denen im Bereich der KI und der humanoiden Robotik entgegenzuwirken.

In dieser Arbeit liegt der Fokus auf einem ersten Schritt zur Umsetzung des Digitalen Kontinuums. Die Analyse aktueller Entwicklungs- und Betriebsprozesse KI-basierter autonomer Systeme zeigt vor diesem Hintergrund noch substanzielle Unterbrechungen in horizontaler, vertikaler und kausaler Kontinuität, die den Übergang zu vollständig autonomen, selbstopimierenden Systemen einschränken. Besonders relevant ist dabei der heute überwiegend manuelle Aufbau von Lern- und Testumgebungen für das Lernen und Testen autonomer Steuerungsstrategien. Durch eine Automatisierung dieses Prozesses lassen sich zum einen direkte Modellierungsaufwände re-

duzieren. Zum anderen werden dadurch ebenfalls die Unterbrechungen des Alignment-Problems und des Sim-to-Real-Gaps im Digitalen Kontinuum reduziert.

Ein prozeduraler Generierungsansatz auf Basis formaler, semantisch beschriebener Modelle erzeugt reproduzierbar große Mengen realistischer Lern- und Testumgebungen, erhöht die Abdeckung relevanter Randfälle und reduziert manuellen Modellierungsaufwand. Dadurch lässt sich das Alignment-Problem entschärfen, weil Ziel- und Nebenbedingungen in unter unterschiedlichsten Bedingungen getestet und Gegenbeispiele gezielt erzeugt werden können. Zugleich wird der Sim-to-Real-Gap verkleinert, da die systematische Variation struktur- und sensorrelevanter Parameter die Generalisierung erlernter Strategien fördert und die Übertragung in die physische Realität verbessert.

Die automatisierte Generierung von Lern- und Testumgebungen wird damit zu einem Schlüsselbaustein des Digitalen Kontinuums. Sie stärkt die kausale Kontinuität der Entwicklungs- und Betriebsprozessketten und verkürzt die Schleifen des Entwicklungszyklus. Das folgende Kapitel greift diese Schlussfolgerung auf, beschreibt die Anforderungen an ein Verfahren zur automatisierten Generierung von Lern- und Testumgebungen am Anwendungsfall von AMR in Materialflusssystemen und leitet für diesen Anwendungsbereich die Forschungslücke ab.

## Kapitel 4

# Anforderungen an ein Verfahren zur Generierung von Lern- und Testumgebungen und Forschungslücke

**Zusammenfassung** In diesem Kapitel werden die Anforderungen an ein Verfahren zur Generierung von Lern- und Testumgebungen für AMR im Digitalen Kontinuum hergeleitet und auf Basis des aktuellen Stands der Technik die bestehende Forschungslücke identifiziert. Die Anforderungen werden aus drei zentralen Kontextbereichen abgeleitet: dem Digitalen Kontinuum, den Aufgaben von AMR in Materialflusssystemen und den Anforderungen des Policy Learning in der Robotik. Darauf aufbauend wird der Stand der Technik in den Bereichen Generierung von Struktur- und von Simulationsmodellen von Materialflusssystemen sowie Generierung von Lernumgebungen für Roboter analysiert. Das abschließende Fazit zeigt, dass bislang kein Verfahren existiert, das eine Vielzahl randomisierter, semantisch kohärenter und realistischer Struktur- und Verhaltensmodelle erzeugen kann, die als Lern- und Testumgebungen für die KI-basierte Entwicklung und den Betrieb von AMR in der Intralogistik geeignet sind.

## 4.1 Ableitung der Anforderungen

Motiviert aus dem Digitalen Kontinuum als idealisiertem Leitbild zur Entwicklung und zum Betrieb hochautonomer Systeme in der Logistik, in dem KI-Agenten als aktiv handelnde Entitäten integriert sind, ergibt sich die Notwendigkeit von Verfahren zur automatisierten Generierung von Lern- und Testumgebungen für KI-basierte Steuerungsstrategien.

Diese Lern- und Testumgebungen sowie ihr Generierungsverfahren müssen dabei eine Reihe von Anforderungen erfüllen. Für den in dieser Arbeit betrachteten Anwendungsfall des Digitalen Kontinuums, AMR in Materialflusssystemen, ergeben sich die Anforderungen aus der Anwendung des Generierungsverfahrens im Digitalen Kontinuum, aus den Aufgabenbereichen und Steuerungskonzepten von AMR in Materialflusssystemen (siehe Unterabschnitt 2.2.4 bis Unterabschnitt 2.2.2) und aus dem Erlernen von Steuerungsstrategien durch Roboter (siehe Unterabschnitt 2.3.2). Sie werden im Folgenden abgeleitet (siehe [Mur23] für eine erste Identifikation der Anforderungsbereiche).

### 4.1.1 Generelle Anforderungen aus dem Digitalen Kontinuum

Das Digitale Kontinuum stellt generelle Anforderungen an ein Verfahren zur Generierung variantenreicher Lern- und Testumgebungen für KI-basierte Steuerungsstrategien. Das Verfahren wird innerhalb eines Durchlaufs des Entwicklungszyklus des Digitalen Kontinuums (siehe Unterabschnitt 3.3.1) automatisiert durch KI-Agenten initiiert und ausgeführt. Aus den Anforderungen des Entwicklungszyklus und aus den Dimensionen der Kontinuität, die in einem vollständig geschlossenen Digitalen Kontinuum angestrebt werden, lassen sich folglich grundlegende Anforderungen an ein solches Generierungsverfahren ableiten.

- *Automatisierung*: Zur Schließung der Prozessketten in der digitalen Realität soll das Generierungsverfahren automatisiert ablaufen und ohne manuelle Eingriffe auskommen.
- *Parallelisierung*: Um die Lern- und Testprozesse effizient zu gestalten und zu beschleunigen, sollen sie in verschiedenen digitalen Umgebungen parallel durchgeführt werden können. Dies ist einer der Vorteile digitaler Entwicklungsumgebungen, und er soll im Entwicklungszyklus des Digitalen Kontinuums genutzt werden.
- *Variabilität angepasst an Lebenszyklus*: Das Verfahren zur Generierung der Lern- und Testumgebungen soll im Digitalen Kontinuum sowohl in der Entwicklungs- als auch in der Betriebsphase autonomer Systeme einsetzbar sein. In der Entwicklungsphase ist eine hohe Variationsbreite aller Umgebungsparameter erforderlich, um eine robuste Generalisierung der

Steuerungsstrategien und damit Übertragbarkeit auf das zukünftige reale System zu ermöglichen und eine Vielzahl potenzieller Szenarien zu erproben. In der Betriebsphase hingegen soll die Variation nur solche Parameter betreffen, die auch in der festgelegten realen Einsatzumgebung veränderbar sind. Im weiteren Verlauf der Arbeit wird der Fokus auf die Entwicklungsphase von AMR in Materialflusssystemen gelegt, da hier die größte Variation an Lern- und Testumgebungen ermöglicht wird. Eine Einschränkung der randomisierten Parameter in der Betriebsphase ist anschließend möglich.

- *Durchgängige Datenintegration*: Die Informationen zum aktuellen Zustand eines autonomen Systems und seiner Umgebung sollen aus Digitalen Zwillingen der Umgebung und des AMR stammen. Um die Integration unterschiedlicher Datenquellen und Hierarchieebenen logistischer Systeme zu ermöglichen, ist die Etablierung eines einheitlichen semantischen Datenmodells erforderlich, auf dem die Lern- und Testumgebungen basieren.

Neben diesen generellen Anforderungen aus der Integration im Digitalen Kontinuum ergeben sich aus dem spezifischen Anwendungsfall von AMR in Materialflusssystemen weitere konkrete Anforderungen an die Lern- und Testumgebungen, die im Folgenden abgeleitet werden.

#### **4.1.2 Anforderungen aus dem Einsatz von AMR in Materialflusssystemen**

AMR übernehmen in Materialflusssystemen ein breites Aufgabenspektrum, von einfachen Transportprozessen bis hin zu komplexen Handhabungs- und Kollaborationsprozessen. Wie in Unterabschnitt 2.2.2 beschrieben, zählen hierzu unter anderem der Transport sowie die Ein- und Auslagerung von Ladeeinheiten, die Handhabung ganzer Ladeeinheiten oder einzelner Artikel, die Unterstützung bei kollaborativen Kommissionierprozessen sowie Serviceaufgaben wie Inventuren. Diese Aufgaben gilt es in den Lern- und Testumgebungen abzubilden, um die Erfüllung der Aufgaben erlernbar zu machen.

Die Erfüllung der Aufgaben durch AMR erfolgt im Digitalen Kontinuum durch autonome Entscheidungsfindung auf Basis der von den eingebauten Sensoren erfassten Informationen und durch die Umsetzung der Entscheidungen in konkrete Handlungen mithilfe spezifischer Hardwarekomponenten (siehe Unterabschnitt 2.2.4). Je nach Steuerungsarchitektur und Autonomiegrad können AMR neben reinen robotischen Steuerungsaufgaben, wie der Ausführung vorgegebener Greifprozesse, auch Steuerungsentscheidungen auf Ebene des Materialflusssystems übernehmen. Die für den effizienten

Betrieb relevanten Steuerungsentscheidungen sind in Unterabschnitt 2.2.3 dargestellt.

Aus diesen Grundlagen ergeben sich spezifische Anforderungen an Lern- und Testumgebungen für AMR in Materialflusssystemen:

- *Abbildung der Struktur des gesamten Materialflusssystems:* Da AMR Aufgaben im gesamten Materialflusssystem übernehmen können, soll die Systemgrenze der Lern- und Testumgebungen alle in Unterabschnitt 2.1.2.1 dargestellten Funktionsbereiche einer Materialflusssysteme umfassen. Dabei sollen möglichst realistische und semantisch kohärente Strukturen von Materialflusssystemen abgebildet werden, um die Übertragbarkeit auf die Realität zu erhöhen.
- *Abbildung physikalisch realistischer, interaktiv nutzbarer Elemente des Materialflusssystems:* Die physikalischen Eigenschaften der Elemente müssen so modelliert werden, dass AMR sowohl Wegeführung als auch Handhabungsaufgaben in der Lern- und Testumgebung erlernen können. Dazu sollen die Elemente physikalisch realistisch und interaktiv nutzbar abgebildet werden. Zudem müssen operative logistische Elemente wie Ladeeinheiten und Flurförderzeuge mit in die Generierung eingebunden werden. Dadurch wird auch die Vorbereitung auf unvorhersehbare Ereignisse wie blockierte Wege oder Engstellen durch Objekte, die in Gängen stehen, möglich.
- *Modellierung von Materialflussbeziehungen:* Viele der Steuerungsentscheidungen auf Ebene des Materialflusssystems, die für den Betrieb von AMR in Materialflusssystemen zu treffen sind, basieren auf den Materialflussbeziehungen zwischen Quellen und Senken in dem System. Diese Materialflussbeziehungen müssen dynamisch im Zeitverlauf in den Modellen abgebildet werden, um geeignete, ganzheitliche Steuerungsstrategien basierend auf ihnen erlernen zu können.
- *Abbildung der technischen Komponenten für autonome Entscheidungsfindung:* Die Funktionsprinzipien der Sensoren, Rechenhardware, Aktoren und Kommunikationsinfrastruktur der AMR sind möglichst realitätsnah zu modellieren, um eine Übertragbarkeit der in der Lernumgebung entwickelten Steuerungsstrategien auf reale Systeme zu ermöglichen.
- *Abbildung variabler Steuerungsarchitekturen:* Das Verfahren muss sowohl dezentrale Steuerungsentscheidungen einzelner AMR als auch zentrale Steuerungsansätze unterstützen, um die unterschiedlichen Steuerungsarchitekturen, die in realen Materialflusssystemen eingesetzt werden, abzubilden.
- *Abbildung von Menschen im Materialflusssystem:* Für kollaborative Aufgaben ist die realitätsnahe Modellierung von Menschen und ihrem Verhalten im Materialflusssystem von Bedeutung. Die Generierung menschlichen Verhaltens stellt jedoch eine eigenständige wissenschaftliche Herausforderung dar und wird in einem eigenen Forschungsfeld adressiert

(vgl. z. B. [BEP24]). Da der Fokus dieser Arbeit auf den logistischen Aspekten des Einsatzes von AMR in Materialflusssystemen liegt, wird die detaillierte Abbildung menschlicher Verhaltensweisen im weiteren Verlauf ausgeklammert. Zukünftig können jedoch Erkenntnisse aus diesem Forschungsbereich in die hier entwickelten Lern- und Testumgebungen integriert werden, um kollaborative Szenarien abzubilden.

### 4.1.3 Anforderungen aus dem Policy Learning in der Robotik

Auch das Policy Learning aus dem Bereich der Robotik stellt Anforderungen an die zu generierenden Lern- und Testumgebungen. Policy Learning, insbesondere unter Einsatz von RL (siehe Unterunterabschnitt 2.3.2.1), hat sich in den letzten Jahren als zentrale Methode für das Training von Steuerungsstrategien autonomer Agenten, wie sie in AMR eingebettet sind, in digitalen Umgebungen etabliert. Zur Steigerung der Robustheit, der Generalisierungsfähigkeit und der Übertragbarkeit auf die Realität von erlernten Steuerungsstrategien hat sich DR als erfolgreich erwiesen. Die generierten Lern- und Testumgebungen für AMR müssen daher die für RL und DR relevanten Aspekte abdecken.

Es ergeben sich daraus folgende Anforderungen an Lern- und Testumgebungen:

- *Abbildung von End-to-End-Steuerung*: Die Lern- und Testumgebungen müssen ermöglichen, die AMR über End-to-End-Algorithmen zu steuern und diese Steuerung in den Umgebungen zu lernen und zu testen. Unter End-to-End-Algorithmen versteht man Verfahren, die Sensordaten ohne Zwischenschritte direkt in Steuerungsaktionen überführen. Diese Aspekte bilden eine Grundvoraussetzung für die Einbindung von Lernverfahren in die Lern- und Testumgebungen. So wird es auch möglich, durch den Einsatz von RL-Algorithmen eine Steuerungsstrategie (engl. Policy) zu finden, die die Steuerung der Agenten der AMR in den Lern- und Testumgebungen vorgibt. Voraussetzung dafür ist, dass die Elemente und operativen Objekte in einem Materialflusssystem sowie die Robotikkomponenten physikalisch korrekt abgebildet werden.
- *Randomisierung der Struktur des Materialflusssystem*: Bei der Generierung der Struktur der Lern- und Testumgebungen ist von Bedeutung, eine Vielzahl von verschiedenen geometrischen Anordnungen und Variationen der Objekte in den Simulationsmodellen zu erzeugen, um die Robustheit der erlernten Steuerungsstrategien gegenüber unbekanntem Umgebungen zu erhöhen.
- *Variation der Materialflussbeziehungen*: Da die Steuerungsentscheidungen auf den Materialflüssen in einem Materialflusssystem basieren und unter verschiedenen Lastszenarien variieren können (z. B. durch saisonale Schwankungen, Tageszeiten oder Störungen), muss das Verfahren varia-

bel konfigurierbare Materialflussbeziehungen in Form von Transportaufträgen für AMR abbilden können, um flexibel angepasste Steuerungsstrategien erlernen zu können.

- *Randomisierung der physikalischen Parameter*: Um den Sim-to-Real-Gap zu schließen und in der Simulation erzeugtes Verhalten auch in der Realität erzeugen zu können, hat sich die Randomisierung der physikalischen Parameter in den Simulationsumgebungen als erfolgreicher Ansatz etabliert (siehe Unterunterabschnitt 2.3.2.2). Da es zu diesem Bereich in der Robotik einige Forschungsarbeiten gibt (siehe z. B. [Mur+22]) und der Fokus dieser Arbeit auf den logistischen Aspekten von AMR in Materialflusssystemen liegen soll, wird die Randomisierung der physikalischen Parameter im Folgenden nicht weiter verfolgt.

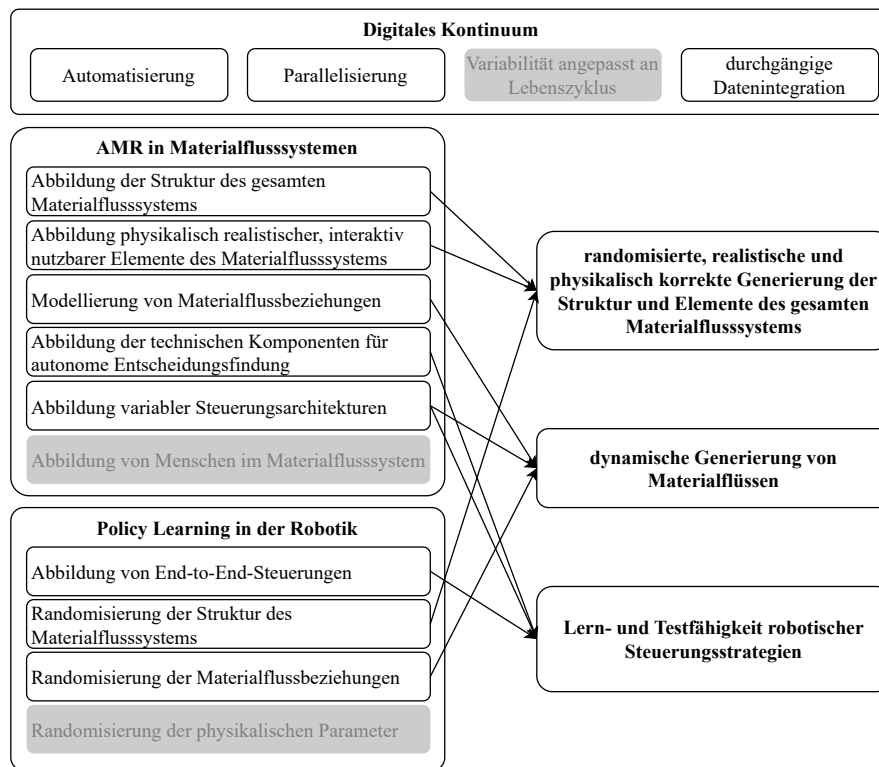
#### 4.1.4 Zusammenfassung der Anforderungen

Die Analyse der Anforderungen aus der Integration in den Entwicklungszyklus des Digitalen Kontinuums, aus dem Einsatz von AMR in Materialflusssystemen und aus dem Policy Learning in der Robotik zeigt, dass ein Verfahren zur Generierung von Lern- und Testumgebungen für AMR ein Spektrum an technischen und funktionalen Eigenschaften erfüllen muss (siehe Abbildung 4.1).

Aus dem Digitalen Kontinuum ergeben sich Anforderungen, die übergeordnet an das Generierungsverfahren gestellt werden: ein hoher *Automatisierungsgrad*, *Parallelisierung* und eine *durchgängige Datenintegration* auf Basis einheitlicher Datenstandards bei der Generierung. Die Anpassbarkeit entsprechend dem Lebenszyklus von AMR wird an dieser Stelle ausgeklammert. Der Fokus liegt auf der Entwicklungsphase, in der noch kein spezifisches logistisches System für den Einsatz der entwickelten AMR festgelegt ist.

Die Anforderungen an die Lern- und Testumgebungen selbst, die aus dem Bereich der AMR in Materialflusssystemen und dem Policy Learning in der Robotik entstehen, lassen sich in drei Kategorien einordnen:

- *Randomisierte, realistische und physikalisch korrekte Generierung der Struktur und der Elemente des gesamten Materialflusssystems*: Die Struktur der generierten Modelle von Materialflusssystemen soll möglichst variantenreich sein und dabei der Struktur realer Materialflusssysteme entsprechen. Zudem sollen physikalische Eigenschaften der Objekte, wie das optische Erscheinungsbild oder Gewichte, in den Modellen möglichst physikalisch korrekt sein, um von Sensorinformationen direkt auf die Robotersteuerung schließen zu können. Neben den technischen Systemen in den Materialflusssystemen sollen auch operative Elemente wie Ladeeinheiten abgebildet werden und interaktiv nutzbar sein.
- *Dynamische Generierung von Materialflüssen*: Die Generierung von Materialflüssen soll dynamisch zur Simulationslaufzeit erfolgen können.



**Abb. 4.1** Zusammenfassung der Anforderungen an die Lern- und Testumgebungen

Dabei sollen die Materialflüsse in Form von Transportaufträgen für AMR variabel konfiguriert werden können und ohne Einschränkung der Steuerungsarchitektur des Materialflusssystems und der AMR in der Simulationsumgebung bereitstehen.

- *Lern- und Testfähigkeit robotischer Steuerungsstrategien:* Es soll möglich sein, in den Lern- und Testumgebungen alle Elemente für realistische End-to-End-Steuerungsstrategien von Robotern abzubilden, die in den Umgebungen erlernt und getestet werden können. Dazu zählen Aufnahme der Zustände durch Sensorinput, Verarbeitung des Sensorinputs zu Steuerungsaktionen und Ausführung dieser Aktionen in den Umgebungen. Die Funktionsweisen der Komponenten für die robotische Steuerung (Sensoren, eingebettete Recheneinheiten, Aktoren und Kommunikationsinfrastruktur) sollen dabei realistisch abgebildet werden können. Die Steuerungsarchitektur der AMR im Materialflusssystem soll zwischen zentral und dezentral flexibel zu gestalten sein.

Darüber hinaus sind bei der Entwicklung des Generierungsverfahrens grundsätzlich die Prinzipien des MBSE zu berücksichtigen (siehe Abschnitt 2.4),

um die *Relevanz der abgebildeten Eigenschaften* für die Zielsetzung und die *Zweckorientierung* der generierten Modelle sicherzustellen.

## 4.2 Relevante Vorarbeiten und ihre Grenzen

Abgeleitet aus den im vorangegangenen Abschnitt identifizierten Anforderungen ergeben sich drei Bereiche, in denen relevante Vorarbeiten für ein Verfahren zur automatisierten Generierung von Lern- und Testumgebungen von AMR in Materialflusssystemen existieren [Mur23, S. 8 f.]:

- Die Generierung von Strukturmodellen von Materialflusssystemen beschäftigt sich mit der Bestimmung der physischen Anordnung und Darstellung der Elemente in Materialflusssystemen und liefert die statische Basis, indem Layouts und Topologien der Systemelemente abgebildet werden.
- Die Generierung von Simulationsmodellen bildet die dynamischen Prozess- und Materialflussbeziehungen ab, die für das Erlernen von Steuerungsstrategien erforderlich sind.
- Die Generierung von Lernumgebungen für Roboter beschäftigt sich mit der Generierung von Struktur und physikalischem Verhalten in RL-tauglichen, variierbaren Szenarien, die sich insbesondere auf universelle robotische Steuerungsaufgaben konzentriert.

Im Folgenden werden die relevanten Vorarbeiten aus diesen Bereichen vorgestellt und ihre Grenzen für die Anwendung im Digitalen Kontinuum herausgearbeitet.

### 4.2.1 Generierung von Strukturmodellen von Materialflusssystemen

Vorarbeiten zur Generierung von Strukturmodellen von Materialflusssystemen stammen überwiegend aus dem Bereich der Planung und Layoutierung logistischer und produktionstechnischer Systeme. In diesem Unterabschnitt werden zunächst existierende Ansätze präsentiert und anschließend ihre Grenzen für die Anwendung als Lern- und Testumgebungen für AMR in Materialflusssystemen herausgearbeitet.

#### 4.2.1.1 Existierende Ansätze

Ein grundlegendes und intensiv untersuchtes Optimierungsproblem im Bereich der Strukturgenerierung von Materialflusssystemen ist das Facility Layout Problem (FLP). Das FLP beschäftigt sich mit der optimalen Anordnung von Elementen im physischen Raum in industriellen Produktionssystemen. Es gilt als eine der wichtigsten Entscheidungen in der Produktionsplanung, da es wesentlichen Einfluss auf Kosten, Effizienz und Produktivität hat.

Dementsprechend wurde es in der Forschung bereits vielfältig betrachtet und auch auf Materialflusssysteme übertragen.

Im Ablauf des FLP ist die Generierung von Layoutalternativen der Teilschritt, der für die Generierung der Struktur der Lern- und Testumgebungen für AMR in Materialflusssystemen relevant ist (siehe Abbildung 4.2). Je nach Planungsphase unterscheidet man im FLP zwischen *Block-Layout-Phase* (Anordnung von Funktionsbereichen) und *Detail-Layout-Phase* (Anordnung von Technologien und Elementen innerhalb der Funktionsbereiche).

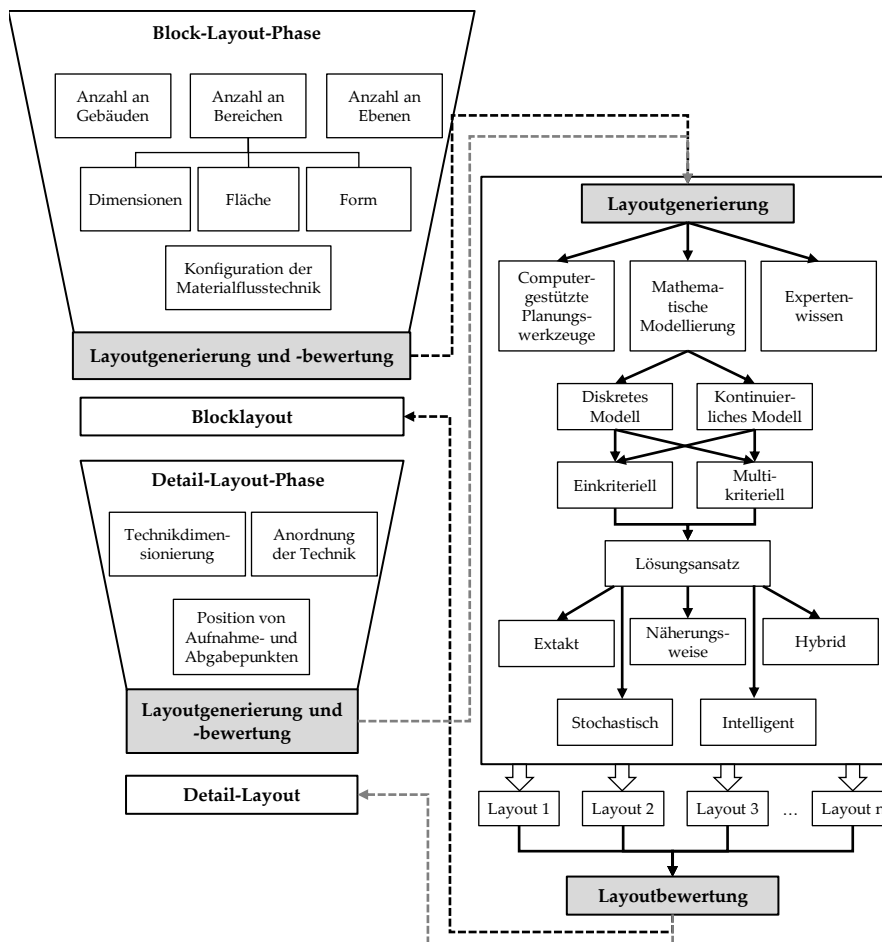


Abb. 4.2 Übersicht über Teilschritte des FLP [PMD21, S. 3778]

Das Ziel des FLP ist die Identifikation eines optimalen Layouts für den jeweiligen Anwendungsfall, häufig auf Basis der Materialflussbeziehungen zwischen den Subsystemen eines Materialflusssystemes. Entsprechend sind viele Lösungsansätze dieses Optimierungsproblems darauf ausgerichtet, in

einem einzelnen Suchlauf ein möglichst effizientes Layout zu finden. Für die in dieser Arbeit betrachtete Generierung vielfältiger, randomisierter Strukturvarianten für Lern- und Testumgebungen sind hingegen Lösungsansätze des FLP relevant, die eine große Vielzahl von Alternativen erzeugen, um sie im Suchprozess zu berücksichtigen. Die folgende Betrachtung der relevanten Vorarbeiten konzentriert sich daher auf diese Ansätze. Ausführliche Übersichten über die Lösungsansätze für das FLP können in [Da+17] und [PMD21] gefunden werden.

Die relevanten Vorarbeiten aus dem Bereich der Strukturgenerierung von Materialflusssystemen sind im Folgenden in die Bereiche der Generierung von alternativen Anordnungen der Funktionsbereiche und der Elemente innerhalb der Funktionsbereiche unterteilt.

### **Anordnung von Funktionsbereichen im Materialflusssystem**

Die automatisierte Erzeugung von Funktionsbereichen in Materialflusssystemen zielt darauf ab, die übergeordnete räumliche Struktur der Systeme festzulegen. Der Großteil der existierenden Ansätze aus diesem Bereich kann der PCG-Kategorie *manuell erstellte Wissensrepräsentation* und *formale Grammatiken* in Abbildung 2.27 eingeordnet werden.

Eine relevante Gruppe bilden in diesem Kontext graphbasierte Layoutdarstellungen. Dabei wird die räumliche Struktur des Layouts in einer formalen Darstellungsform beschrieben. Beispiele hierfür sind Slicing Bäume, Sequence Pairs,  $B^*$ -Bäume und O-Bäume. Diese Methoden beschreiben, wie Funktionsbereiche relativ zueinander angeordnet sind, zum Beispiel rechts von, links von, oberhalb oder unterhalb. Varianten werden basierend auf diesen Darstellungsformen über Verfahren zur Veränderung der Strukturen gebildet, etwa durch Tauschen von Elementen, Umhängen von Teilbäumen oder Drehen von Teilen. Ursprünglich im Bereich der Chipflächenplanung entwickelt [Cha+00; Guo+14; MK98], wurden diese Ansätze später auf Materialflusssysteme übertragen [Mos17; PMD21].

Darüber hinaus werden regelbasierte und constraint-orientierte Verfahren eingesetzt, bei denen gültige Layouts durch formale Regeln und Nebenbedingungen erzeugt werden. Beispiele sind Ansätze wie Constraintprogrammierung Tavares et al. [TRN02] und verschiedene Arbeiten zu Antwortmengeprogrammierung für die Layoutplanung [Kai+23; Sch20; The+24].

Parametergesteuerte Konstruktionsverfahren sind eine weitere relevante Gruppe zur Generierung von Layoutvarianten. Dabei wird das Layout anhand definierter Zielvorgaben, Randbedingungen und Entscheidungsvariablen wie Flächenanteilen, Abständen und Mindestgrößen aufgebaut [SP21]. Ein Beispiel aus der Gebäudekonstruktion und der Einteilung des Gebäudes in Räume, vergleichbar mit Funktionsbereichen in Materialflusssystemen, wird in [FE20] vorgeschlagen.

Aktuelle Entwicklungen umfassen außerdem lernende Generierungsansätze, etwa Verfahren des RL in Kombination mit graphbasierten Ansätze, die Layoutstrukturen in der Simulationen lernen [YDY24].

Insgesamt zeigt sich, dass bestehende Ansätze zur Generierung alternativer Anordnungen von Funktionsbereichen für Planungsaufgaben entwickelt wurden. Trotz ihrer grundsätzlichen Fähigkeit, eine Bandbreite von möglichen Strukturen zu erzeugen, sind sie bisher nicht für die Generierung randomisierter Layouts eingesetzt. Dennoch bieten sie teils relevante Ansatzpunkte für die randomisierte Generierung der Anordnung von Funktionsbereichen.

### **Anordnung von Elementen in Funktionsbereichen**

Die Generierung der Inhalte von Funktionsbereichen umfasst die Bestimmung der geometrischen Anordnung von Elementen wie Regalen, Verkehrswegen und weiteren physischen Objekten. Auch in diesem Bereich erfolgt die Generierung meist nach *formalen Grammatiken*.

Ein verbreiteter Ansatz ist die parametergesteuerte Konstruktion von Regalen und Gangsystemen in Lagerbereichen des Materialflusssystems. Hierbei werden feste Regeln und geometrische Parameter wie Gangabstände, Winkel, Sicherheitsabstände oder Rastergrößen zur Erzeugung von Layoutvarianten genutzt. Ein Beispiel ist ein konstruktives Verfahren zur Gestaltung von Gängen in Kommissionierlagern, das unter anderem Gangwinkel und Abmessungen systematisch variiert [ÖGM14].

Auch für die Anordnung von Elementen in Funktionsbereichen kommen constraint-orientierte Verfahren wie Antwortmengenprogrammierung zum Einsatz. Dabei werden beispielsweise Anordnungen von Regalen nur dann erzeugt, wenn Anforderungen wie Erreichbarkeit oder Verkehrswege erfüllt sind. Solche Verfahren werden unter anderem in der automatischen Generierung von Layouts von Lagerbereichen eingesetzt [Kai+21].

Darüber hinaus existieren inkrementelle Konstruktionsverfahren, die Layouts schrittweise aufbauen und bei Bedarf anpassen. Ein Beispiel dafür ist die Arbeit von Shahroudnejad et al., in der auf Basis einer constrained Beam Search fortlaufend Layoutelemente wie Regalblöcke platziert und ungültige Konfigurationen aussortiert werden, um innerhalb vorgegebener Flächen möglichst effiziente Layouts zu erzeugen [Sha+24].

Auch erste RL-basierte Ansätze zur prozeduralen Generierung von Arbeitsstationen und zur Lösung universeller Layoutierungsprobleme wurden untersucht [Lóp+20; UB21].

Insgesamt ist den existierenden Ansätzen gemein, dass sie nur einzelne Funktionsbereiche betrachten und oft in der Praxis unübliche oder stark abstrahierte Layouts erzeugen, die operative, interaktive Elemente nicht betrachten und daher für robotische Anwendungen nicht geeignet sind.

#### 4.2.1.2 Einschränkungen für AMR-Lern- und Testumgebungen

Die bestehenden Verfahren zur automatisierten Generierung von Layouts in Materialflusssystemen bieten eine gute Basis für die Generierung der Struktur von Lern- und Testumgebungen für AMR, zeigen jedoch Grenzen.

Klassische Strukturgenerierungsansätze sind in erster Linie darauf ausgerichtet, in Optimierungsverfahren eingesetzt zu werden und ein einzelnes möglichst effizientes und regelkonformes Layout zu planen. Für das Training und die Validierung erlernter Steuerungslogiken von AMR wird jedoch nicht eine optimale Lösung benötigt, sondern eine große Vielfalt an unterschiedlich komplexen, realitätsnahen und semantisch kohärenten Szenarien.

Hinzu kommt, dass bestehende Ansätze zur Layoutgenerierung typischerweise geometrische Strukturen nur bis zu einem Detailgrad abbilden, der für die Planung relevant ist, und keine physikalisch korrekten Eigenschaften oder operative, interaktive Elemente wie Ladeeinheiten und Flurförderzeuge in den Lagersystemen und Gängen integrieren. Für AMR sind diese Elemente jedoch für realistische Sensorwahrnehmungen, Kollisions- und Kontaktmechaniken und unvorhersehbare Ereignisse wie blockierte Wege oder Engstellen entscheidend.

Zudem werden die Ansätze zur Generierung der Funktionsbereiche und ihrer Inhalte in der Regel nicht miteinander kombiniert. Der Fokus der Generierung von Elementen in Funktionsbereichen liegt auf Lagersystemen, sodass es an ganzheitlichen Ansätzen zur detaillierten Abbildung gesamter Materialflusssysteme fehlt.

Insgesamt fehlt damit ein ganzheitlicher Strukturgenerierungsansatz, der physikalische Realitätstreue, semantische Anforderungen und hohe Variationsfähigkeit vereint.

#### 4.2.2 Generierung von Simulationsmodellen für Materialflusssysteme

Weitere relevante Vorarbeiten für die Generierung von Lern- und Testumgebungen im Digitalen Kontinuum, die für den Anwendungsfall von AMR in Materialflusssystemen geeignet sind, liegen im Forschungsfeld der Generierung von Simulationsmodellen für Materialflusssysteme. Dieses Forschungsfeld hat zum Ziel, das Verhalten von Elementen in Materialflusssystemen, ihre Interaktionen auf Materialflussebene und die zeitlichen Abläufe automatisch in einem virtuellen Modell darzustellen. Für die Entwicklung und das Testen von AMR-Funktionalitäten sind solche Modelle von Bedeutung, da sie Prozessbedingungen, wie z. B. Materialflüsse, Auslastungen oder Engpässe, abbilden können.

Im Folgenden werden existierende Ansätze zur Generierung von Simulationsmodellen für Materialflusssysteme vorgestellt und ihre Grenzen für den

Einsatz als Lern- und Testumgebungen für AMR im Digitalen Kontinuum aufgezeigt.

#### 4.2.2.1 Existierende Ansätze

Wenzel [Wen+19] und Bergmann [BS10, S. 546] unterscheiden die gängigen Ansätze der automatisierten Generierung von Simulationsmodellen für Materialflusssysteme (engl. Automated Simulation Model Generation (ASMG)) in:

- *parametrische Ansätze*: Diese Verfahren verwenden vordefinierte Bausteine oder Modellbibliotheken, die anhand von Parametern konfiguriert und kombiniert werden. Sie verfolgen das Ziel, die Modellierung zu beschleunigen und wiederverwendbare Simulationskomponenten zu schaffen. Ein Beispiel dafür ist der Ansatz von Kallat et al., die eine automatische Variation bestehender Simulationsmodelle für die Lagerplanung entwickeln. Ihr Ansatz generiert automatisch Varianten aus einer Komponentendatenbank am Beispiel eines Blocklagers [Kal+21].
- *strukturbasierte Ansätze*: Diese Methoden generieren Simulationsmodelle direkt aus strukturellen Datenquellen wie Computer-aided-Design-(CAD-)Modellen, Anlagenlayouts oder Produktionsdaten. Steinbacher et al. (2024) zum Beispiel zeigen, wie aus Daten von Logistikdienstleistern automatisch Materialflusssimulationen abgeleitet werden können [Ste+24].
- *hybride wissensbasierte Ansätze*: Hierbei werden parametrische und strukturbasierte Verfahren kombiniert und mit Domänenwissen oder KI-Techniken erweitert. Die Ansätze basieren auf dem Prinzip, Modellstruktur und -verhalten durch semantische Modelle, Ontologien oder regelbasierte Wissenssysteme zu steuern [SGM25; Wen+19].

Bei den Ansätzen in diesen Kategorien geht es hauptsächlich um die Generierung von Simulationsmodellen existierender Systeme oder um die Planung von Systemen. Die Modelle sind dabei zweidimensional, physikalisch vereinfacht und nicht für die Abbildung robotischer Steuerungsstrategien geeignet. Eine weiterführende Übersicht über gängige ASMG-Ansätze für Materialflusssysteme bieten Reinhardt et al. [RWP19].

#### 4.2.2.2 Einschränkungen für AMR-Lern- und Testumgebungen

Bestehende Ansätze der automatisierten Generierung von Simulationsmodellen im Bereich Materialflusssysteme konzentrieren sich auf die Abbildung realer oder geplanter Materialflusssysteme und deren planerische Optimierung. Für Lern- und Testumgebungen für AMR ergeben sich daraus verschiedene Einschränkungen.

Die meisten Verfahren erzeugen Modelle, die ein bestimmtes System oder eine konkrete Anlagenausprägung nachbilden. Damit fehlt die Fähigkeit,

randomisierte und zufällige Prozessvariationen und Anordnungen im System zu erzeugen, wie sie für das Training robuster Steuerungsstrategien erforderlich sind.

Auch die Integration roboterrelevanter Entscheidungsmechanismen und -komponenten ist bislang nicht abgebildet. Klassische Materialflusssimulationen bilden Prozesse auf einer abstrakten Ebene ab, beispielsweise durch Transportaufträge oder Ressourcenbeziehungen. Aspekte wie Lokalisierung, Hindernisvermeidung oder Interaktion durch Handhabung sind nicht Teil der Modelllogik. Auch die Komponenten der Roboter werden in diesen Umgebungen nicht detailgenau abgebildet. Damit fehlen die für das Lernen einer robotischen Steuerungsstrategie notwendigen Elemente.

Diese Lücken verdeutlichen die Notwendigkeit, Simulationsmodelle zu generieren, die sowohl strukturelle als auch verhaltensrelevante Variabilität bieten und direkt für das Lernen von Steuerungsstrategien in der Robotik einsetzbar sind.

### 4.2.3 Generierung von Lernumgebungen für Roboter

Vorarbeiten im Bereich der Generierung von Lernumgebungen für Roboter stammen primär aus der Robotik- und KI-Forschung. Diese Arbeiten beschäftigen sich damit, Umgebungen zu schaffen, in denen autonome Agenten Steuerungsstrategien erlernen, testen und validieren können. Dabei liegt der Fokus entweder auf der Entwicklung von Benchmark-Frameworks, die universelle Aufgaben und Umgebungen für Roboter bereitstellen, oder auf generativen Verfahren, die eine automatische und variantenreiche Erstellung von Szenen ermöglichen. Ein weiterer Schwerpunkt sind Methoden zur DR, die Variabilität insbesondere in die visuelle Erscheinung der Lernumgebungen einführen, um die Generalisierungsfähigkeit der trainierten Modelle zu erhöhen (siehe Unterunterabschnitt 2.3.2.2).

In diesem Unterabschnitt werden existierende Ansätze und ihre Grenzen in Bezug auf die Eignung als Lern- und Testumgebungen für AMR in Materialflusssystemen aufgezeigt.

#### 4.2.3.1 Existierende Ansätze

Einige der Ansätze zur Generierung robotischer Lernumgebungen setzen auf die Erzeugung visueller Varianz. Mit GenAug schlagen Chen et al. beispielsweise ein Verfahren zur Generierung robotischer Trainingsumgebungen vor, das auf Bild-Text-Generierungsmodellen basiert. Der Ansatz generiert vielfältige Szenen eines Tisches, in denen Roboter Objekte erkennen und greifen müssen, und variiert dazu Texturen, Hintergründe, zusätzliche Elemente und Objektkategorien [Che+23]. Der Fokus liegt auf der visuellen Vielfalt und nicht auf der Abbildung komplexer logistisch relevanter Umgebungen.

Auch Initiativen wie GRADE [BXA24] legen den Fokus auf die Erzeugung visueller Varianz in der Darstellung der Szenen.

Zhou et al. entwickeln eine modulare Simulations- und Testumgebung als industrielles Benchmark-Framework für die KI-gestützte Robotermanipulation auf Basis von Nvidia Omniverse Isaac Sim. Das Framework beschreibt acht universelle Manipulationsaufgaben, die erweitert oder parametrisiert werden können [Zho+23]. Für die Generierung von Lernumgebungen für Manipulationsaufgaben stellt die Arbeit somit ein Rahmenwerk dar, beschreibt jedoch selbst keine automatisierte oder dynamische Generierung neuer Lern- und Testumgebungen.

Im Bereich des autonomen Fahrens existiert mit SceneControl ein Generierungsansatz für Lernumgebungen, der verschiedene, realistische Verkehrsszenarien erzeugt [Lu+24]. Dabei werden die initiale Platzierung und Eigenschaften der Akteure im Modell über ein Diffusionsmodell festgelegt (z. B. Größe, Geschwindigkeit), bevor deren Verhalten im Verlauf der Simulation durch eine Policy bestimmt wird. Dieses Prinzip kann auch für Szenen von Materialflusssystemen sinnvoll sein.

Mit konkretem Bezug zur Generierung von Modellen von Materialflusssystemen existiert der Ansatz SceneBlox von Nvidia [NVI23]. Es ist ein Ansatz zur prozeduralen Generierung randomisierter Materialflusssysteme in physikalisch akkuraten Simulationsumgebungen, basierend auf dem Wave-Function-Collapse-Algorithmus. Obwohl SceneBlox eine breite Variation an Layouts generiert und damit den Anwendungsfall im Digitalen Kontinuum adressiert, folgen die erzeugten Strukturen in den Lern- und Testumgebungen nicht den in der Logistik etablierten Gestaltungsprinzipien und sind semantisch nicht kohärent. Dadurch ist ihre Realitätsnähe für intralogistische Anwendungen nur eingeschränkt gegeben. Zudem werden dynamische Materialflussbeziehungen, die während des Systembetriebs entstehen, nicht berücksichtigt.

#### 4.2.3.2 Einschränkungen für AMR-Lern- und Testumgebungen

Die bestehenden Arbeiten zur Generierung von Lernumgebungen für Roboter liefern relevante Ansätze für visuelle Varianz und universelle Robotikaufgaben. Sie adressieren jedoch nur teilweise die spezifischen Anforderungen von AMR in Materialflusssystemen.

- Generative Ansätze wie GenAug [Che+23] fokussieren auf visuelle Vielfalt, ohne logistische Abläufe oder physische Strukturen von Materialflusssystemen einzubeziehen.
- Benchmark-Frameworks wie [Zho+23] bieten modular konfigurierbare Simulationsplattformen, setzen den Fokus jedoch auf universelle Robotikaufgaben und beziehen sich nicht konkret auf Materialflusssysteme.

- SceneBlox [NVI23] bietet zwar prozedurale Szenengenerierung von Materialflusssystemen, orientiert sich jedoch nicht an realistischen Prinzipien der Gestaltung von Materialflusssystemen und sind damit nur eingeschränkt übertragbar.

Insgesamt fehlt im Bereich der Lernumgebungen für Roboter bislang ein Verfahren, das die Generierung variantenreicher, realitätsnaher Struktur- und Verhaltensmodelle für Materialflusssysteme ermöglicht und dabei die spezifischen Anforderungen von AMR in Materialflusssystemen erfüllt. Bestehende Ansätze liefern Robotik-relevante Bausteine, decken die Anforderung einer konsistenten, auf logistischen Prinzipien basierenden Umgebung jedoch nicht ab.

### 4.3 Forschungslücke

Die Analyse bestehender Ansätze in den Bereichen der automatisierten Strukturmodellierung, der ASMG und der Generierung von robotischen Lernumgebungen zeigt vielversprechende Ansätze, jedoch auch ein strukturelles Defizit, das zentrale Anforderungen des Digitalen Kontinuums bislang nicht abdeckt.

Verfahren zur Layoutgenerierung werden in der Regel im Rahmen von Optimierungsverfahren eingesetzt und zielen dadurch auf eingeschränkte Varianz in statischen Planungsproblemen ab. Außerdem bleiben die erzeugten Modelle häufig auf einer hohen Ebene geometrischer Abstraktionen und beziehen interaktive operative Elemente wie Ladeeinheiten nicht mit ein. Es fehlt ein systematischer Ansatz zur automatisierten Generierung variantenreicher, semantisch kohärenter Umgebungen, wie sie für lernfähige agentische AMR-Systeme erforderlich sind.

In der ASMG dominieren Reproduktions- und Parametrisierungsansätze: Bestehende Systeme werden abgebildet, modular zusammengesetzt oder variiert, jedoch kaum von Grund auf randomisiert erzeugt. Zudem eignen sich die gängigen Detaillierungsgrade der Modelle nicht für die Abbildung robotischer End-to-End-Steuerungsalgorithmen. Die Fähigkeit zur Szenariengenerierung im Sinne von Lern- und Testumgebungen für autonom agierende AMR bleibt damit unerschlossen.

Ansätze aus dem Bereich der Generierung von Lern- und Testumgebungen für Roboter unterstützen relevante Teilaspekte, insbesondere visuelle und physikalische Realitätsnähe, fokussieren jedoch vorrangig auf Manipulationsaufgaben und beziehen intralogistische Prozesszusammenhänge nicht ein.

Vor diesem Hintergrund ergibt sich eine zentrale Forschungslücke, die nicht nur methodischer, sondern struktureller Natur ist:

Es existiert bislang kein Verfahren zur automatisierten Generierung realitätsnaher, logistikgerechter und variantenreicher Lern- und Testumgebungen, das zugleich

- die heterogene Struktur intralogistischer Systeme abbildet,
- dynamische Lastszenarien simulativ integriert,
- realistische Entscheidungsmechanismen von KI-Agenten in AMR einbindet und
- den Prinzipien der Anwendung in einem lernenden, selbstregulierenden Digitalen Kontinuum entspricht.

Diese Lücke ist mehr als ein fehlendes Werkzeug. Sie ist die Schwelle, die überschritten werden muss, um simulationsbasierte kontinuierliche Intelligenz und Autonomie in der Logistik operationalisierbar zu machen. Die vorliegende Arbeit setzt genau an diesem Punkt an und formuliert einen Ansatz zur Generierung von Lern- und Testumgebungen als funktionale Grundlage des Digitalen Kontinuums. Damit wird nicht nur eine methodische Lücke geschlossen, sondern der nächste notwendige Entwicklungsschritt hin zu einer KI-integrierten Logistikarchitektur eingeleitet.



## Kapitel 5

# Entwicklung eines Verfahrens zur automatisierten Generierung von Lern- und Testumgebungen für AMR in Materialflusssystemen

**Zusammenfassung** Basierend auf den in Kapitel 4 dargestellten Anforderungen wird in diesem Kapitel ein Algorithmus für die automatisierte Generierung von Lern- und Testumgebungen für AMR in Materialflusssystemen entwickelt. Der generelle Ablauf des Algorithmus ist in zwei Phasen unterteilt:

- die Generierung der statischen Struktur der Lern- und Testumgebungen vor Start eines Simulationslaufs und
- die Generierung des Verhaltens der Lern- und Testumgebungen während der Simulation.

Insbesondere in der ersten Phase steht die Randomisierung der Umgebungen im Fokus, um das durch die digitalen Agenten der AMR erlernte und getestete Verhalten robust gegenüber einer Vielzahl verschiedener Strukturvarianten zu machen und so die Generalisierungsfähigkeit auch in realen Systemen zu erhöhen.

## 5.1 Grundlegende Generierungsprinzipien

In diesem Abschnitt werden die grundlegenden Prinzipien vorgestellt, die die Basis des im weiteren Verlauf entwickelten Generierungsansatzes bilden. Damit wird der Rahmen geschaffen, in dem die Anforderungen aus der Forschungslücke systematisch in ein Verfahren überführt werden können. Zunächst wird der übergeordnete Ablauf des Generierungsverfahrens beschrieben, um die Logik des Ansatzes nachvollziehbar darzustellen. Anschließend erfolgt die Auswahl und Einordnung geeigneter Verfahren aus dem Bereich der PCG (siehe Abschnitt 2.5), die die identifizierten Teilaufgaben unterstützen.

### 5.1.1 Übergeordneter Ablauf des Generierungsverfahrens

Das entwickelte Generierungsverfahren ist in die Phasen der Initialisierung der Simulationsumgebung, in der die Struktur der Lern- und Testumgebungen erzeugt wird, und der Simulationszeit, zu der das Verhalten in den Umgebungen generiert wird, eingebettet, wie in Abbildung 5.1 dargestellt.

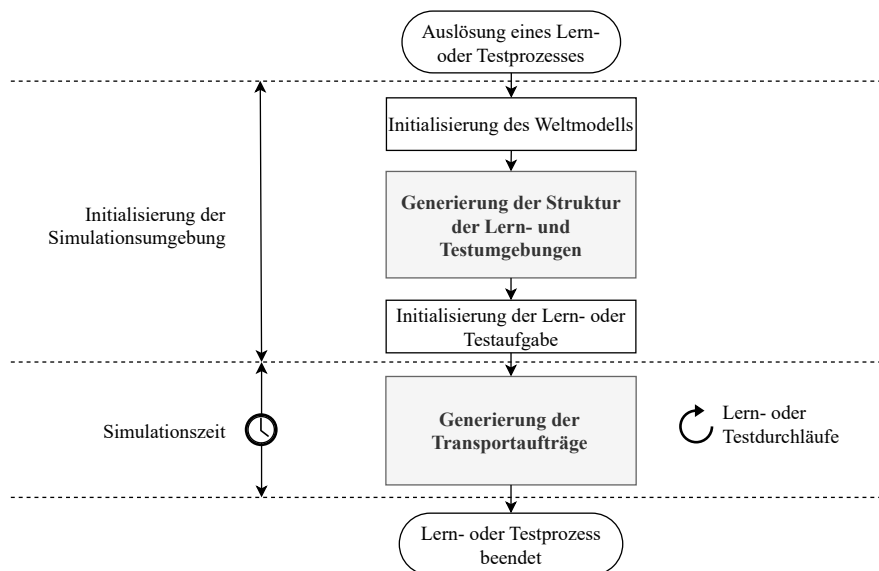


Abb. 5.1 Grundsätzlicher Ablauf des Generierungsverfahrens

Aus den in Abschnitt 4.1 identifizierten Anforderungen folgt, dass reine Materialflusssimulationsumgebungen nicht als Lern- und Testumgebungen für AMR im Digitalen Kontinuum ausreichen. Erforderlich sind vielmehr Modelle, die auf einer Physiksimulation basieren, um sowohl die physikalisch konsistente Abbildung der Elemente als auch die Modellierung der technischen Komponenten von AMR zu gewährleisten.

Der im Folgenden entwickelte Generierungsansatz ist daher eingebettet in den generellen Ablauf von Physiksimulationen, die für RL von robotischen Steuerungsstrategien genutzt werden (siehe Abbildung 5.1). Nach Auslösen eines Lern- oder Testprozesses erfolgt zunächst die Initialisierung des Weltmodells. Dabei werden Generierungsparameter (z. B. Objekttypen, Anzahl und Typen von Funktionsbereichen), physikalische Umgebungsbedingungen (z. B. Lichtquellen, Zeitspanne zwischen Aktualisierungen der Physiksimulation, Maßeinheiten) sowie ein Raster für parallele Lern- und Testumgebungen festgelegt.

Darauf folgt die randomisierte Erzeugung der Struktur für jede einzelne Umgebung.

Mit Beginn der Simulation werden schließlich Transportaufträge dynamisch erzeugt und die Lern- oder Testdurchläufe episodisch durchgeführt.

Das Generierungsverfahren gliedert sich somit in zwei Kernphasen:

1. Vor Beginn der Simulation wird die randomisierte, physische Struktur des Modells erzeugt.
2. Während der Simulation werden die Materialflüsse dynamisch generiert, sodass alle relevanten Parameter für die Steuerungsentscheidungen der AMR bereitstehen.

Damit umfasst das Verfahren sowohl die initiale Erstellung der physischen Struktur als auch die Abbildung dynamischer Materialflüsse während der Laufzeit. Für beide Phasen sind geeignete Generierungsverfahren erforderlich, die den formulierten Anforderungen entsprechen und Variabilität sowie Realitätsnähe einbinden.

## 5.1.2 Auswahl geeigneter Generierungsansätze

In diesem Unterabschnitt werden geeignete Ansätze für die beiden Phasen des Generierungsverfahrens ausgewählt. Für die Generierung der Struktur wird ein Verfahren der PCG betrachtet, das vor Simulationsstart die statische Konfiguration des Materialflusssystemes erzeugt. Die Generierung der Materialflüsse erfolgt hingegen dynamisch während der Simulationslaufzeit. Dazu wird ein serverbasierter Ansatz herangezogen, bei dem eine zentrale Steuerlogik zur Laufzeit Transportaufträge erzeugt.

### 5.1.2.1 Struktur

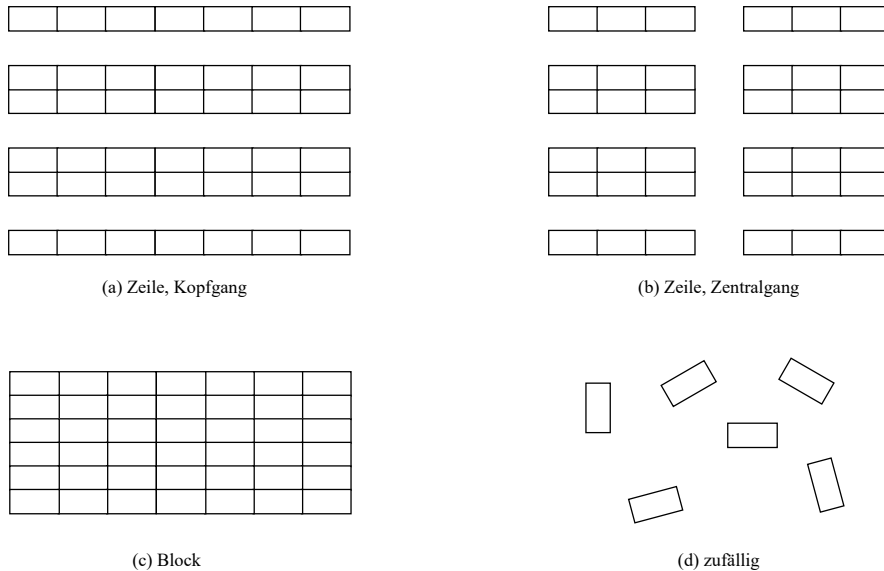
Die Generierung der Struktur einer Lern- und Testumgebung umfasst die Aufgabe, die statischen Elemente eines Materialflusssystemes und die darin enthaltenen operativen Objekte in geeigneter Weise zu positionieren. Der Generierungsalgorithmus muss dafür bestimmen,

- welche Objekte
- an welchen Positionen und

- in welcher Orientierung

platziert werden. Zu diesen Objekten zählen sowohl die installierten Materialflusstechnologien als auch operative Einheiten wie Ladeeinheiten, Ladehilfsmittel und Verpackungsmaterialien, die für eine realistische Abbildung des Systembetriebs erforderlich sind.

Ein wesentliches Merkmal von Materialflusssystemen ist ihr hierarchischer, strukturierter Aufbau. Sie sind in Funktionsbereiche organisiert, in denen technische Systeme installiert sind, in denen Elemente wie Ladeeinheiten platziert sind (siehe Abschnitt 2.1). Viele Anordnungen innerhalb der Funktionsbereiche folgen wiederkehrenden Mustern, etwa regelmäßigen Rastern oder Blockstrukturen, wie sie beispielsweise in Regallagern oder Bodenblocklagern zum Einsatz kommen [Roi22, S. 71]. Typischerweise orientieren sich die statischen Objekte an Zeilen- oder Blockmustern, während mobile Objekte im laufenden Betrieb zufällig im System verteilt sind (siehe Abbildung 5.2).



**Abb. 5.2** Anordnungsarten von Objekten in einem Funktionsbereich

Da ein großer Teil des Wissens über die Struktur von Materialflusssystemen bereits formalisiert und beschrieben ist, muss die Anordnungslogik nicht durch datenbasierte Verfahren gelernt werden. Vielmehr bietet es sich an, diese Logik direkt in der Generierungsmethode abzubilden. Der Einsatz constraint-basierter Algorithmen wie des Wave Function Collapse ist hier wenig zielführend, da die einfachen Muster von Materialflusssystemen deren Potenzial kaum ausschöpfen würden, erhebliche Einschränkungen notwendig wären und die Muster über einfache Regeln kodierbar sind.

Ein geeigneter Ansatz besteht daher in der Kombination aus einer *manuell erstellten* Wissensrepräsentation und einer *formalen Grammatik* (siehe Abbildung 2.27), wie sie häufig auch im Bereich der Layoutgenerierung für Planungsprozesse eingesetzt werden (siehe Unterabschnitt 4.2.1). Die Wissensrepräsentation erfolgt durch manuell definierte Bausteine, die die Objekte von Materialflusssystemen abbilden. Die Generierungsmethode sorgt anschließend für eine zufällige, aber dennoch realistische und semantisch kohärente Variation dieser Strukturen, indem sie Parameter wie Position, Rotation oder die Anzahl der Elemente randomisiert.

Eine zentrale Herausforderung besteht darin, die richtigen Parameter für die Randomisierung zu identifizieren: Einerseits muss genügend Variabilität erzeugt werden, um vielfältige Lern- und Testumgebungen zu schaffen, andererseits dürfen die erzeugten Strukturen nicht unrealistisch werden oder die logischen Rahmenbedingungen verletzen, die für Materialflusssysteme charakteristisch sind.

Zusammenfassend erfordert die Generierung der Struktur von Lern- und Testumgebungen für AMR im Digitalen Kontinuum einen Ansatz, der auf manuell erstellten Wissensrepräsentationen und formalen Grammatiken beruht, der die bekannten strukturellen Muster von Materialflusssystemen aufgreift und durch gezielte Randomisierung eine große Vielfalt realistischer Varianten erzeugt.

#### 5.1.2.2 Materialflüsse

Neben der strukturellen Modellierung der Lern- und Testumgebungen ist die Generierung realistischer Materialflüsse ein zentraler Bestandteil des Verfahrens. Unter Materialflüssen werden in diesem Kontext Transportaufträge verstanden, die während der Simulation erzeugt und durch AMR ausgeführt werden sollen.

Um größtmögliche Flexibilität für unterschiedliche Steuerungsarchitekturen zu gewährleisten, werden die Transportaufträge nicht direkt einzelnen Akteuren zugeordnet, sondern zunächst neutral im System bereitgestellt. Dadurch ist es möglich, dass sowohl zentrale Steuerungsinstanzen als auch dezentrale Architekturen mit verteilten Entscheidungsmechanismen der AMR auf die Aufträge zugreifen und diese entsprechend der Lernaufgabe bearbeiten können.

Die Generierung der Transportaufträge erfolgt dynamisch zur Simulationszeit. Hierfür wird ein serverbasierter Mechanismus eingesetzt, der kontinuierlich neue Aufträge erzeugt. Dabei orientiert sich das Verfahren an etablierten Ansätzen aus der Materialflusssimulation, bei denen die Verteilung der Auftragseingänge häufig durch einen Poisson-Prozess modelliert wird. Auf diese Weise entsteht stochastisch Dynamik im System.

Das Ergebnis dieser Komponente des Generierungsverfahrens ist ein Strom von Transportaufträgen, der unabhängig von der gewählten Steue-

rungsarchitektur genutzt werden kann und eine realistische Abbildung dynamischer Materialflüsse in der Simulation ermöglicht.

## 5.2 Generierung der Struktur der Lern- und Testumgebungen

Bei der Generierung der Struktur der Lern- und Testumgebungen ist von besonderer Bedeutung, eine Vielzahl von verschiedenen geometrischen Anordnungen und Variationen der Objekte in den Simulationsmodellen zu erzeugen und dabei weiterhin realistische Materialflusssysteme zu erzeugen. In den Generierungsschritten der Struktur sind daher randomisierte Elemente eingebaut, die für Variation in der Konfiguration der Lern- und Testumgebungen sorgen und dabei gängige Gestaltungsmuster in Materialflusssystemen befolgen. Die einzelnen Generierungsschritte sind modular aufgebaut und können jeweils durch andere Verfahren ergänzt, ausgetauscht und erweitert werden, um zukünftig weitere Variation, Funktionsbereiche und Technologien mit in den Algorithmus aufnehmen zu können. In dem in der vorliegenden Arbeit entwickelten Vorgehen werden alle Lern- und Testumgebungen parallel in die gleiche Welt, also das gleiche Simulationsmodell, generiert, um so eine parallele Ausführung der Lern- und Testprozesse erreichen zu können.

In den folgenden Unterabschnitten wird zunächst ein Überblick über den semantischen Aufbau der Lern- und Testumgebungen gegeben, bevor die Generierungsschritte inklusive der randomisierten Parameter im Detail vorgestellt werden.

### 5.2.1 Überblick über die Strukturgenerierung

Im Folgenden wird ein übergeordneter Einblick in den semantischen Aufbau der generierten Modelle und den generellen Ablauf der Strukturgenerierung gegeben.

#### 5.2.1.1 Elemente der Lern- und Testumgebung

Zunächst werden der semantische Aufbau und die Bestandteile des Modells erläutert, das dem Generierungsalgorithmus für digitale Lern- und Testumgebungen für AMR in Materialflusssystemen zugrunde liegt, wie er im weiteren Verlauf des Kapitels entwickelt wird. Abbildung 5.3 bietet die Übersicht über den semantischen Aufbau in der SysML (siehe Unterabschnitt 2.4.2). Er orientiert sich an dem hierarchischen Aufbau realer Materialflusssysteme (siehe Abschnitt 2.1).

Eine Lern- und Testumgebung ist eingebettet in einen globalen Kontext, der die übergeordnete Weltumgebung abbildet. Die übergeordnete Welt besitzt eine generelle Physikumgebung, durch die physikalische Eigenschaften

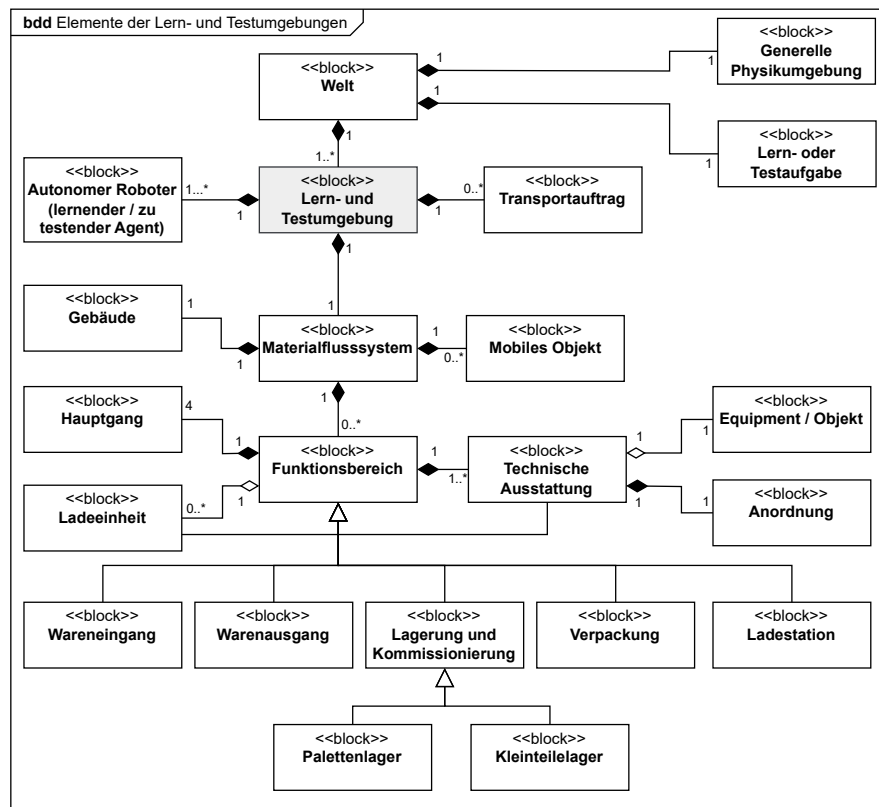


Abb. 5.3 Semantisches Modell der Elemente der Lern- und Testumgebungen im Simulationsmodell

definiert werden, die für alle Lern- und Testumgebungen gültig sind. Zu diesen physikalischen Eigenschaften gehören beispielsweise Lichtquellen, die Grundfläche im Modell, die Zeitspanne zwischen Aktualisierungen der Physiksimulation und Maßeinheiten im Modell. Außerdem ist eine Lern- oder Testaufgabe Teil der Welt, die die relevanten Parameter der Aufgabe festlegt. In einer Welt können zahlreiche Lern- und Testumgebungen existieren, jedoch besteht mindestens eine.

Die Bestandteile einer digitalen Lern- und Testumgebung orientieren sich an den Anforderungen an diese Umgebungen, die in Abschnitt 4.1 hergeleitet wurden. Sie besteht aus genau einem Materialflusssystem, das die strukturellen Elemente der intralogistischen Lern- und Testumgebung abbildet, einem oder mehreren Transportaufträgen, die im Verlauf der Simulation das Verhalten des Systems erzeugen, und mindestens einem autonomen Roboter, der in diesen Umgebungen lernen und getestet werden soll.

Das Materialflusssystem bildet die strukturelle Grundlage für die Lern- und Testumgebung und ist selbst aus verschiedenen Teilen aufgebaut. Ein

Materialflusssystem besteht mindestens aus einem Gebäude. Diese Konfiguration kann für Anwendungsfälle sinnvoll sein, in denen ein leeres Gebäude als Grundlage dient. In der Regel jedoch werden die Materialflusssysteme in den Lern- und Testumgebungen ebenfalls Funktionsbereiche und mobile Objekte besitzen. Die mobilen Objekte sind selbst keinem der Funktionsbereiche direkt zugewiesen und dienen dazu, die Umgebungen realitätsnäher zu gestalten. Sie werden als letzter Schritt der strukturellen Generierungslogik erstellt (siehe Abbildung 5.4) und beliebig im System verteilt, um realistische Hindernisse zu erzeugen, die den AMR beim Erlernen und Testen ihres Verhaltens begegnen können.

Den Hauptteil der Struktur eines Materialflusssystems machen die Funktionsbereiche aus. Jeder Funktionsbereich besteht aus Zugängen zum Hauptgang, den ihm zugeordneten Ladeeinheiten und mindestens einer technischen Ausstattung. Ein Hauptgang bildet die Schnittstelle des Funktionsbereichs zum übrigen System und ist daher für jeden Funktionsbereich erforderlich. In einem Funktionsbereich können keine bis zahlreiche Ladeeinheiten vorhanden sein. Da die Ladeeinheiten von einem Funktionsbereich im Materialflusssystem zu einem anderen Funktionsbereich transportiert werden können, sind sie nicht exklusiver Teil des Funktionsbereichs. Die technische Ausstattung des Funktionsbereichs bestimmt, in welcher Aufbauorganisation der Bereich mit welchem Equipment ausgestattet wird. Die technischen Ausstattungen werden in den folgenden Abschnitten für die einzelnen modellierten Funktionsbereiche detailliert dargestellt. Die Funktionsbereiche in einem Materialflusssystem sind Wareneingang, Warenausgang, Lagerung und Kommissionierung, Verpackung und Ladestation. Der Bereich Lagerung und Kommissionierung kann in dieser Arbeit entweder ein Palettenlager oder ein Kleinteilelager sein, da sie sich auf manuelle Materialflusssysteme zur Stückgutabwicklung fokussiert.

#### 5.2.1.2 Ablauf

Der Ablauf der Generierung der Struktur der Lern- oder Testumgebungen folgt der hierarchischen Struktur realer Materialflusssysteme (siehe Abbildung 5.4). Das Ziel der Strukturgenerierung ist, eine möglichst diverse Verteilung der Objekte im Sinne der ausgewählten Assets, ihrer Position und ihrer Rotation zu erreichen, und dabei die gängigen Gestaltungsmuster in Materialflusssystemen zu befolgen. *Assets* bezeichnet im Folgenden die Modellbausteine, die logistische Objekte in den digitalen Lern- und Testumgebungen abbilden.

Nachdem das Weltmodell inklusive seiner Physikumgebung und der Positionen der Lern- oder Testumgebungen im Raster erstellt ist sowie die Parameter der Assetbibliothek und der gewünschten Konfiguration der Materialflusssysteme durch die Vorgaben für die Lern- oder Testaufgabe feststehen, werden die Materialflusssysteme an den Koordinaten im Raster

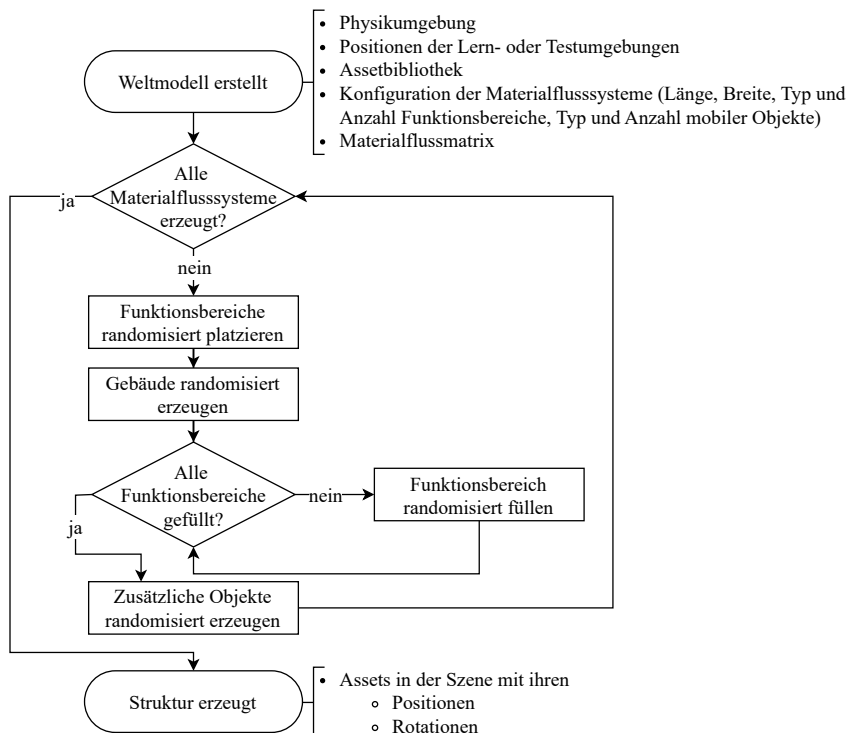


Abb. 5.4 Ablauf der Strukturgenerierung

erzeugt. Die Funktionsbereiche geben vor, an welchen Stellen technische Systeme in Materialflusssysteme installiert sind. Daher werden zunächst die Funktionsbereiche randomisiert innerhalb der vorgegebenen Maße der Materialflusssysteme platziert. Sobald die Funktionsbereiche erzeugt sind, steht fest, an welchen Stellen im Gebäude spezifische Gebäudeelemente wie Tore verbaut sind. Daher wird als Nächstes das Gebäude randomisiert erzeugt. Anschließend werden die Funktionsbereiche randomisiert mit ihren technischen Systemen gefüllt. Das Ergebnis des Generierungsalgorithmus spezifiziert die Assets mit ihren Positionen und Rotationen in jeder generierten Lern- oder Testumgebung.

### 5.2.2 Generierung der Materialflusssysteme

Im Folgenden wird die Generierung der Materialflusssysteme im Detail beschrieben.

Ein Materialflusssystem ist die logische Einheit, die die strukturelle, statische Umgebung einer Lern- und Testumgebung umfasst und die statischen Objekte im Simulationsmodell unter sich vereint. Im Kontext der Lern- und

Testumgebungen für AMR besteht es aus einem *Gebäude*, *Funktionsbereichen* und *mobilen Objekten* (siehe Abbildung 5.5).

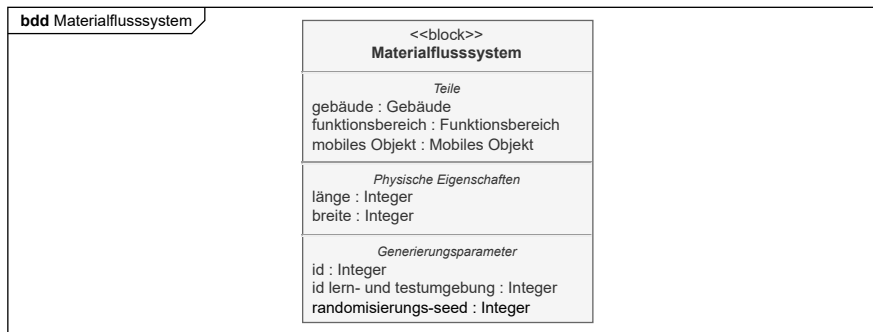


Abb. 5.5 Semantisches Modell eines Materialflusssystems

Ein Materialflusssystem hat physische Abmessungen in Form der *Länge* und *Breite*. Es hat eine eindeutige *ID* und ist einer festen *Lern- und Testumgebung* zugeordnet, deren *ID* ebenfalls im Materialflusssystem vermerkt ist. Außerdem ist jedem Materialflusssystem ein *Randomisierungsseed* zugeordnet. Der Seed legt den Startwert fest, der verwendet wird, um einen Zufallszahlengenerator zu initialisieren. Durch die Übergabe des Seeds für die Randomisierung der geometrischen Anordnung der Assets im Materialflusssystem wird zum einen sichergestellt, dass die Materialflusssysteme aller Lern- und Testumgebungen mit unterschiedlichen Zufallszahlen generiert werden und dadurch verschieden sind, zum anderen werden die Ergebnisse reproduzierbar.

Durch die vorangegangenen Schritte sind die Inputparameter für die Generierung der Struktur der Materialflusssysteme gegeben: die  $x$ - und  $y$ -Position des Mittelpunkts des Materialflusssystems in der Welt, die Länge  $l_S$  und Breite  $b_S$  des Systems, die Anzahlen der verschiedenen Typen an Funktionsbereichen  $N_{FB}^S$  im System und die Anzahlen und Typen an mobilen Objekten im System  $N_{MO}^S$ .

### 5.2.2.1 Generierung der Funktionsbereiche

Die Generierung der Materialflusssysteme startet mit der randomisierten Aufteilung des Materialflusssystems in die spezifizierten Arten und Anzahlen der Funktionsbereiche. Es wird eine rechteckige Form der Funktionsbereiche angenommen. Im Vergleich zu bisher entwickelten Algorithmen zur Anordnung von Funktionsbereichen in einem Materialflusssystem, die sich damit beschäftigen, die optimale Anordnung der Bereiche mit festen Größen zu finden (siehe Unterabschnitt 4.2.1), wird in diesem Algorithmus eine unter realistischen Randbedingungen möglichst variable Anordnung der Funktionsbereiche angestrebt. Das Verfahren orientiert sich an dem Ansatz von Freiknecht und Effelsberg zur Verteilung von Räumen in Gebäuden [FE20].

### Semantisches Modell

Zunächst wird das semantische Modell der Funktionsbereiche in einer Lern- und Testumgebung vorgestellt. Funktionsbereiche bilden den Hauptteil der inneren Struktur eines Materialflusssystemes. Insbesondere ihre technische Ausstattung bestimmt, wie sich die strukturelle Anordnung spezifischer Assets im Modell und damit die physische Erscheinung des Funktionsbereichs ausprägt.

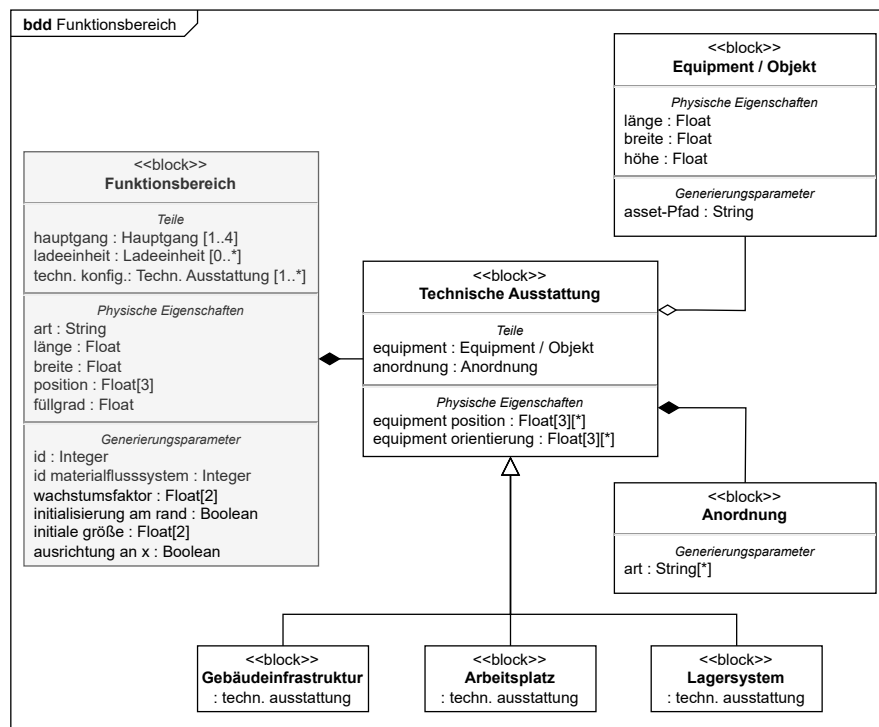


Abb. 5.6 Modellierung eines Funktionsbereichs und seiner technischen Konfiguration

Abbildung 5.6 stellt die Eigenschaften dar, mit denen die Funktionsbereiche und ihre technischen Ausstattungen im Simulationsmodell abgebildet werden. Die *Art* des Funktionsbereichs bestimmt seine technische Ausstattung und ist daher ein Attribut des Funktionsbereichs. Die Kategorien für die *Art* eines Funktionsbereichs, die in dem in dieser Arbeit entwickelten Generierungsalgorithmus berücksichtigt sind, sind Wareneingang, Warenausgang, Lagerung und Kommissionierung, Verpackung und Ladestation. Ein Funktionsbereich hat eine *Länge* und eine *Breite*, die die Ausdehnung des Bereichs in x- und y-Richtung im Simulationsmodell beschreiben, und eine *Position*, die die x-, y- und z-Koordinaten des Mittelpunktes des Funktionsbereichs in der Welt der Lern- und Testumgebungen abbildet. Der *Füllgrad* des Funktionsbe-

reichs bestimmt den Anteil der zur Verfügung stehenden Lagerplätze, der in dem Bereich mit Ladeeinheiten gefüllt ist. Jeder Funktionsbereich hat eine eindeutige *ID*, über die er identifiziert und gefunden werden kann. Zudem ist der Funktionsbereich einem eindeutigen *Materialflusssystem* zugeordnet, das ebenfalls über eine ID gekennzeichnet ist. Die letzten vier Eigenschaften eines Funktionsbereichs sind Werte, die für den Generierungsalgorithmus relevant sind. Der *Wachstumsfaktor*, der boolesche Wert *Initialisierung am Rand* und die *initiale Größe* sind relevant für den Teil des Generierungsalgorithmus, in dem ein Funktionsbereich in das Materialflusssystem generiert wird, bevor er mit Objekten ausgestattet wird. Der boolesche Wert *Ausrichtung an x* ist für die Bestimmung der Orientierung der technischen Ausstattung relevant. Eine Ausrichtung an der x-Achse bedeutet, dass in den Funktionsbereichen vorhandene Gänge im Simulationsmodell entlang der x-Achse verlaufen.

Ein Funktionsbereich kann eine Reihe an verschiedenen technischen Ausstattungen besitzen, abhängig von seiner Art und Aufgabe im Materialflusssystem. Die technische Ausstattung eines Funktionsbereichs besteht aus dem Equipment, das durch Objekte im Simulationsmodell dargestellt wird, und der Art der Anordnung dieser Objekte. Sie kombiniert diese beiden Teile und bestimmt die *Position* und die *Orientierung* aller Objekte des Equipments in x-, y- und z-Richtung im Simulationsmodell. Die technische Ausstattung kann unterteilt werden in die Kategorien spezifische Gebäudeinfrastruktur, die für einen Funktionsbereich benötigt wird, wie beispielsweise Tore im Warenein- und Warenausgang, Arbeitsplatz und Lagersystem.

### Generierungslogik

Im Folgenden wird beschrieben, wie die Funktionsbereiche randomisiert im Materialflusssystem platziert werden. Die Funktionsbereiche werden zunächst zufällig in der durch Länge und Breite des Materialflusssystems vorgegebenen Fläche verteilt (siehe Abbildung 5.7 (a)). Dabei werden die Funktionsbereiche, die einen Zugang zu einer Außenwand benötigen, direkt am Rand der Fläche erzeugt. Die Funktionsbereiche, die nicht zwingend einen Zugang zu einer Wand benötigen, werden im Inneren der Fläche generiert. Durch zwei weitere Randbedingungen bei der initialen Platzierung wird sichergestellt, dass die Funktionsbereiche eine für ein reales Materialflusssystem realistische Größe annehmen:

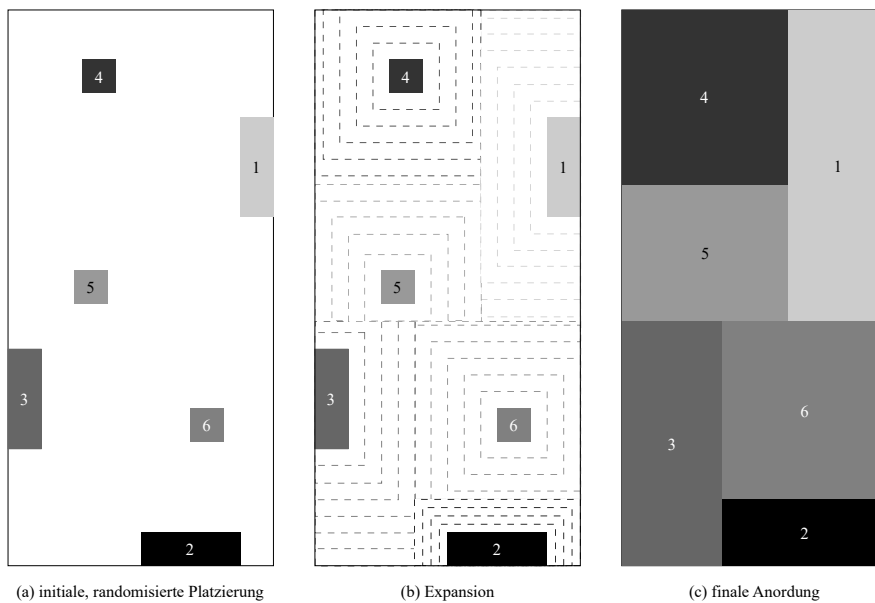
- Jeder Funktionsbereich hat eine festgelegte initiale Größe in x- und y-Richtung,  $b_{fb}^{init}$  und  $l_{fb}^{init}$ . Diese initiale Größe wird so gewählt, dass mindestens ein Element der technischen Ausstattung in dem Funktionsbereich Platz findet.
- Der minimale Abstand zwischen den initialen Positionen der Mittelpunkte der Funktionsbereiche wird durch den Faktor  $a_{fb}^{min}$  festgelegt.

**Randomisierungsfaktor 1: Platzierung der Funktionsbereiche im Materialflusssystem**

Die  $x$ - und  $y$ -Koordinaten der Mittelpunkte der Funktionsbereiche werden zufällig innerhalb der durch das Materialflusssystem vorgegebenen Fläche gewählt. Randbedingungen sind dabei, dass die initiale Länge und Breite der Funktionsbereiche, der minimale Abstand zwischen Funktionsbereichen und die Vorgabe der Lage an der Wand oder im Inneren des Materialflusssystem eingehalten werden.

$$x_{fb} \sim U(x_S^{min} + \frac{1}{2}b_{fb}^{init}, x_S^{max} - \frac{1}{2}b_{fb}^{init}) \quad \forall fb \in FB \quad (5.1)$$

$$y_{fb} \sim U(y_S^{min} + \frac{1}{2}l_{fb}^{init}, y_S^{max} - \frac{1}{2}l_{fb}^{init}) \quad \forall fb \in FB \quad (5.2)$$



**Abb. 5.7** Zufällige Platzierung und Expansion der Funktionsbereiche

Bei der Initialisierung der Funktionsbereiche werden ihnen randomisierte Wachstumsfaktoren für die  $x$ - und  $y$ -Richtung zugeordnet. Je nach Typ des Funktionsbereichs wird ein mögliches Intervall  $[\gamma_{fb}^{min}, \gamma_{fb}^{max}]$  für die Wachstumsfaktoren festgelegt. Diese Einschränkung der Wachstumsfaktoren ist ein weiterer Mechanismus im Generierungsalgorithmus, um realistische Größen der Funktionsbereiche in Materialflusssystem zu erreichen. So ist beispielsweise der Funktionsbereich mit Ladestationen in der Regel kleiner als die

Lagerbereiche für Paletten und Kleinteile und sollte daher mit kleineren Wachstumsfaktoren initialisiert werden.

Anhand der Wachstumsfaktoren werden die Funktionsbereiche anschließend iterativ expandiert (siehe Abbildung 5.7 (b)). Dabei werden die Funktionsbereiche anhand ihrer Generierungsreihenfolge durchlaufen und jeweils an allen vier Seiten um ihren Wachstumsfaktor in der jeweiligen Richtung erweitert. Sobald eine Erweiterung des Bereichs dafür sorgt, dass ein anderer Funktionsbereich überschritten wird, wird der wachsende Funktionsbereich nur noch um die Differenz bis zu dem angrenzenden Bereich erweitert. Im Anschluss an die Expansion steht die finale Anordnung der Funktionsbereiche, inklusive ihrer Dimensionen und Position (siehe Abbildung 5.7 (c)). Alle Materialflusssysteme im Simulationsmodell erhalten auf diese Art und Weise unterschiedliche Layouts.

#### **Randomisierungsfaktor 2: Wachstumsfaktoren der Funktionsbereiche**

Durch die randomisierte Verteilung der Wachstumsfaktoren der Funktionsbereiche in  $x$ - und  $y$ -Richtung werden die finale Größe, die Position und die Form der Funktionsbereiche bestimmt und randomisiert.

$$\gamma_{fb}^x \sim U_{fb}(\gamma_{fb}^{min}, \gamma_{fb}^{max}) \quad \forall fb \in FB \quad (5.3)$$

$$\gamma_{fb}^y \sim U_{fb}(\gamma_{fb}^{min}, \gamma_{fb}^{max}) \quad \forall fb \in FB \quad (5.4)$$

An diesem Punkt der Generierung steht fest, an welchen Positionen des Materialflusssystemes Funktionsbereiche mit speziellen Anforderungen an die Gebäudeinfrastruktur liegen und welche ihrer Seiten an Wände angrenzen. Im nächsten Schritt wird daher das Gebäude um die Funktionsbereiche herum generiert.

#### **5.2.2.2 Generierung des Gebäudes**

Das Gebäude eines Materialflusssystemes stellt seine äußere Hülle dar (siehe Unterunterabschnitt 2.1.2.2). Es ist ein bedeutender Faktor des Erscheinungsbildes eines Materialflusssystemes, da es durch die Gebäudeinfrastruktur wichtige Merkmale liefert, die die Wahrnehmung der Umgebung durch autonome Systeme beeinflussen. Im Folgenden werden sein semantisches Modell und seine Generierungslogik beschrieben.

#### **Semantisches Modell**

Das Gebäude ist mit einer generellen Infrastruktur ausgestattet, die für Industriebauten erforderlich ist, wie es Lagerhallen sind. Zusätzlich können Funktionsbereiche spezifische Anforderungen an die Gebäudeinfrastruktur haben.

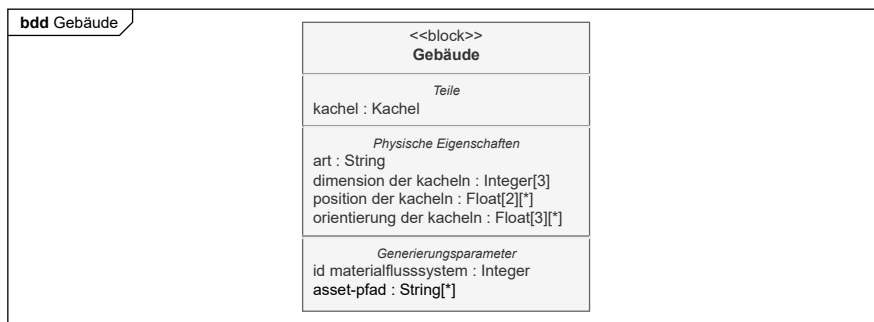


Abb. 5.8 Semantisches Modell des Gebäudes

In dem Kontext des hier entwickelten Generierungsalgorithmus wird für die Generierung des Gebäudes ein blockbasierter PCG-Ansatz eingesetzt. Aus diesem Grund besteht das Gebäude aus einer Menge an *Kacheln*, die durch den Generierungsalgorithmus zusammengesetzt werden (siehe Abbildung 5.8). Das Gebäude verwaltet die *Art* und *Dimensionen* der Kacheln, ihre *Position* und *Orientierung* im Simulationsmodell und den *Asset-Pfad*, der festlegt, auf welchen Ort in der Assetbibliothek die Elemente im Simulationsmodell verweisen.

Die Generierung des Gebäudes erfolgt durch die Platzierung von zufällig ausgewählten Gebäudekacheln aus einer Menge an qualifizierten Gebäudekacheln, die die Randbedingungen des jeweiligen Bereichs des Materialflusssystemes erfüllen. Randbedingungen für die Menge an qualifizierten Gebäudekacheln ergeben sich aus

- der Geometrie des Gebäudes (Ecke, Wand, Innenteil) und
- der Lage und den spezifischen technischen Anforderungen einzelner Funktionsbereiche (z. B. Tore im Warenein- und Warenausgang).

Die Gebäudekacheln enthalten die gesamte Gebäudeinfrastruktur. Dazu gehören Boden, Decke und bei den Wandkacheln Wände inklusive Fenster sowie Gebäudeinfrastruktur wie Rohre für die Sprinklerung, Belüftungsschächte und Beleuchtung. Durch die Aneinanderreihung der Gebäudekacheln lässt sich daher eine komplette Halle zusammenstellen, die einer realistischen Abbildung einer realen Logistikhalle entspricht. Alle Gebäudekacheln haben die gleiche, quadratische Grundfläche und sind untereinander kompatibel.

Die Form der Materialflusssysteme wird in dieser Arbeit als rechteckig angenommen. Diese Form ist die in der Realität am häufigsten gewählte Form (siehe Unterunterabschnitt 2.1.2.2) und über die Länge  $l_S$  und Breite  $b_S$  des Systems vollständig beschrieben. Die Länge und die Breite des Systems entsprechen jeweils Vielfachen der Abmessungen der Gebäudekacheln,  $b_k$  und  $l_k$ . Die Anordnung der Gebäudekacheln erfolgt im Blocksatz (siehe Abbildung 5.2(c)). Die  $x$ - und  $y$ -Koordinaten des Mittelpunktes einer Gebäudekachel ergeben sich dann wie folgt in Abhängigkeit vom Mittel-

punkt des Materialflusssystemes  $(x_S, y_S)$ , der durch das Raster der Lern- und Testumgebungen im Simulationsmodell bestimmt ist:

$$x_k = x_S - \frac{l_S}{2} + \left(i - \frac{1}{2}\right) * l_k \quad \forall i \in \left\{1, \dots, \frac{l_S}{l_K}\right\} \quad (5.5)$$

$$y_k = y_S - \frac{b_S}{2} + \left(j - \frac{1}{2}\right) * b_k \quad \forall j \in \left\{1, \dots, \frac{b_S}{b_K}\right\} \quad (5.6)$$

Die Blockpositionen der Gebäudekacheln werden iteriert, und zu jeder Position wird die Menge an qualifizierten Gebäudekacheln ermittelt, die die Anforderungen der Position im Block erfüllen. Aus der Menge an qualifizierten Gebäudekacheln wird zufällig gewählt. Auf diese Art und Weise entstehen unterschiedliche Gebäude für die Vielzahl an verschiedenen Lern- und Testumgebungen im Simulationsmodell.

### **Randomisierungsfaktor 3: Zufällige Wahl der Gebäudekachel-Assets**

Aus einer Menge an  $n$  qualifizierten Gebäudekacheln  $K = \{k_1, \dots, k_n\}$  wird gleichverteilt zufällig ausgewählt und somit die Gestaltung der Gebäudeinfrastruktur randomisiert.

$$P(k) \sim U(1, n) \quad \forall k \in K \quad (5.7)$$

Im Anschluss an die Generierung des Gebäudes stehen die äußere Hülle des Materialflusssystemes und die Aufteilung der Fläche in Funktionsbereiche. Es folgt die Ausstattung der Funktionsbereiche im Materialflusssystem mit Objekten.

#### **5.2.2.3 Ausstattung der Funktionsbereiche**

Auch bei der Generierung der Ausstattung der Funktionsbereiche in den Materialflusssystemen der Lern- und Testumgebungen wird das Ziel verfolgt, eine hohe Varianz an verschiedenen Ausstattungen zu generieren, die weiterhin realistischen intralogistischen Gestaltungsprinzipien folgen. Bevor die Generierung spezifischer Funktionsbereiche erläutert wird, werden zunächst die Aspekte beschrieben, die für alle Funktionsbereiche gelten: die Generierung der Hauptgänge und die Generierung wiederkehrender Muster in den Funktionsbereichen.

#### **Hauptgang**

Die Generierung der Elemente in den Funktionsbereichen startet mit der Erstellung der Zugänge zum Hauptgang, der den Hauptverkehrsweg zwischen den Funktionsbereichen des Materialflusssystemes bildet. Dazu wird rund um den Funktionsbereich herum im Abstand der Hälfte der Breite des Haupt-

gangs  $b_{HG}$  eine Markierung gesetzt. Die Gesamtheit der Markierungen ergibt das Hauptverkehrsnetz des Materialflusssystem (siehe Abbildung 5.9).

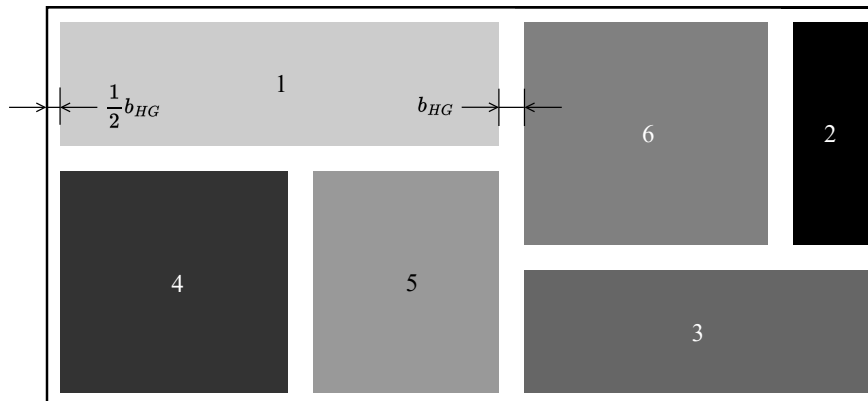


Abb. 5.9 Bildung der Hauptgänge im Materialflusssystem

Die Verkehrswege in einem Materialflusssystem werden für Transporte mit Flurförderzeugen und durch Fußgänger genutzt. In Systemen, in denen AMR eingesetzt werden, nutzen auch diese die Strecken, um Transporte durchzuführen oder zu ihrem nächsten Einsatzort zu fahren. Auf den Verkehrswegen begegnen sie anderen Verkehrsteilnehmern und müssen angemessene Strategien finden, um auf diese zu reagieren, ohne sie in Gefahr zu bringen, und gleichzeitig ihr eigenes Ziel weiterverfolgen. Die Verkehrswege sind daher ein wichtiges Element für die Lern- und Testaufgaben von AMR in Materialflusssystemen, insbesondere für Aufgaben, die mit der Navigation im System zusammenhängen. Um neben der Verkehrsführung durch das System, die sich aus der Position und Ausdehnung der Funktionsbereiche ergibt, weitere Variation in die Gestaltung der Verkehrswege zu bringen, wird bei der Generierung der Hauptgänge die Breite variiert. Durch die Veränderung der Breite der Verkehrswege werden die AMR in Lern- und Testszenarien auf unterschiedliche Ausweichsituationen treffen und auf die Realität vorbereitet, denn auch in realen Materialflusssystemen ist die Gangbreite variabel.

In der Praxis wird die Breite der Verkehrswege in einem Materialflusssystem durch eine Reihe an Normen und Richtlinien vorgegeben. Nach [DIN 18225, S. 3] setzt sich die Gangbreite aus der größten Breite der eingesetzten Transportmittel inklusive des Ladeguts und einem Randzuschlag von 0,5 m bis 0,75 m pro Seite zusammen. Bei Gegenverkehr kommt ein weiterer Zuschlag von 0,4 m hinzu. Bei geringen Verkehrsbewegungen können alle Zuschläge auf 1,10 m heruntersgesetzt werden. Laut [VDI 2199, S. 31] ist ein Sicherheitsabstand von 0,5 m auf beiden Seiten des Verkehrswegs einzuplanen. Bei Fußgängergegenverkehr wird 1,2 m Gangbreite und bei Gegenverkehr mit Flurförderzeugen 4 m Gangbreite empfohlen. Die technischen Regeln für

Arbeitsstätten der Berufsgenossenschaft Handel und Warenlogistik (BGHW) geben 0,9 m bis 2,4 m Gangbreite für Fußgängerverkehr an, je nach Anzahl der Personen, und orientieren sich bei Wegen für Fahrzeugverkehr an der DIN 18225 [ASR A1.8, S. 7 ff.]. Die Gangbreite in Materialflusssystemen ist also abhängig von der Verkehrsart, der Breite der eingesetzten Förderzeuge und der Kombination von Personen- und Fahrzeugverkehr und kann daher auch in der Realität variieren. Bei einer Breite von Flurförderzeugen zwischen 1 m und 2,3 m können realistische Gangbreiten zwischen 2 m und 6,5 m liegen. Für den Generierungsalgorithmus können die Grenzen der Gangbreiten anwendungsfallspezifisch vorgegeben werden.

#### Randomisierungsfaktor 4: Breite des Hauptgangs

Die Breite des Hauptgangs wird gleichverteilt zufällig zwischen zwei Grenzen  $[b_{HG}^{min}, b_{HG}^{max}]$  gewählt, die reale Richtlinien und Normen erfüllen. Durch die Variation der Breite werden die Randbedingungen der Verkehrssituationen verändert und die Lern- und Testaufgaben so in verschiedenen Umgebungen, die in der Realität auftreten können, durchgeführt.

$$b_{HG} \sim U(b_{HG}^{min}, b_{HG}^{max}) \quad (5.8)$$

Nach der Integration der Hauptverkehrswege in einen Funktionsbereich wird dieser mit Assets ausgestattet, die das technische Equipment im Materialflusssystem darstellen. Die Funktionsbereiche im Materialflusssystem sind je nach Art jeweils mit verschiedenen technischen Ausstattungen gefüllt (siehe Abschnitt 2.1). Die technische Ausstattung ist unterteilt in die Kategorien *Gebäudeinfrastruktur*, *Arbeitsplatz* und *Lagersystem*. Jede technische Ausstattung hat ein *Equipment*, das einem Asset im Simulationsmodell entspricht, und eine *Anordnung*, die bestimmt, in welcher Struktur die Assets des Equipments in dem Funktionsbereich verortet werden.

In bisherigen Arbeiten, die sich mit der Auswahl und Anordnung von Technologien in Funktionsbereichen beschäftigt haben, wird das Ziel verfolgt, die eine optimale Konfiguration für das Materialflusssystem zu finden. In diesen Ansätzen werden die Technologien zunächst basierend auf Kapazitäts- und Leistungsvorgaben dimensioniert und anschließend im System platziert. Durch die Dimensionierung der Technologie ergibt sich die Größe der Funktionsbereiche (vgl. z. B. [Sch20]). Weitere Ansätze beschäftigen sich damit, die Fläche baulich stark eingeschränkter Funktionsbereiche möglichst effizient auszunutzen, wobei Anordnungsmuster entstehen, die für den Anwendungsfall optimal sind, jedoch in der Realität nicht häufig vorkommen (vgl. z. B. [Kai+21]). In dem in dieser Arbeit entwickelten Generierungsalgorithmus ver-

hält es sich wie folgt: Die Größe der Funktionsbereiche steht an diesem Punkt fest und die ausgewählten Technologien werden entsprechend realistischen Anordnungsmustern in dem vorgegebenen Funktionsbereich eingesetzt.

### Wiederkehrende Muster

Wie in Unterabschnitt 5.1.2 gibt es Strukturen in Materialflusssystemen, die sich wiederholen: Anordnung in Zeilen, Anordnung im Block und zufällige Muster. Die Bestimmung der Position der Assets in diesen Mustern wird im Folgenden vorgestellt.

#### *Anordnung in Zeilen*

Die Anordnung von Assets in Zeilen kann in intralogistischen Systemen an verschiedenen Stellen gefunden werden. Klassischerweise werden Lagersysteme in Zeilen angeordnet, da so ein direkter Zugriff auf alle Ladeeinheiten in dem Lager stattfinden kann. Auch feste Arbeitsplätze sind häufig in Zeilen angeordnet.

Bei der Anordnung in Zeilen werden die Assets auf zwei Seiten eines Ganges platziert und dabei jeweils die Vorderseite des Assets zum Gang hin gedreht. Für die Generierung von Assets in Zeilen spielt daher nicht nur die Position in  $x$ - und  $y$ -Richtung eine Rolle, sondern auch die Drehung um die  $z$ -Achse.

Außerdem sind folgende Faktoren relevant, um die Positionen von Assets in Zeilen in einem vorgegebenen Bereich zu bestimmen (siehe Abbildung 5.10):

- Die Längen und Breiten der eingesetzten Assets sind nicht zwangsweise Vielfache der Längen und Breiten der zu füllenden Bereiche. Es bleibt dementsprechend Fläche übrig, die nicht groß genug ist, um ein weiteres Asset einzufügen. Um die Assets in Zeilen in die Mitte des zu füllenden Bereiches zu platzieren, wird der Platz, der in  $x$ - und  $y$ -Richtung verbleibt,  $l_v$  in  $x$ -Richtung und  $b_v$  in  $y$ -Richtung, bestimmt und auf beiden Seiten verteilt.
- Ein weiterer Faktor ist die Ausrichtung der Gänge. Verlaufen sie entlang der  $x$ -Achse, sind die Positionen der Assets anders zu bestimmen, als wenn sie entlang der  $y$ -Achse verlaufen. Dieser Punkt ist auch für die Bestimmung der Orientierung eines Assets im Modell von Bedeutung.

Verlaufen die Gänge in Richtung der  $x$ -Achse, lassen sich die Koordinaten eines Assets ausgehend vom Mittelpunkt des zu füllenden Bereichs  $B$  wie folgt berechnen:

$$x_O = x_B - \frac{l_B}{2} + \frac{l_v}{2} + \left(i - \frac{1}{2}\right) * l_A \quad \forall i \in \left\{1, \dots, \left\lfloor \frac{l_B}{l_A} \right\rfloor\right\} \quad (5.9)$$

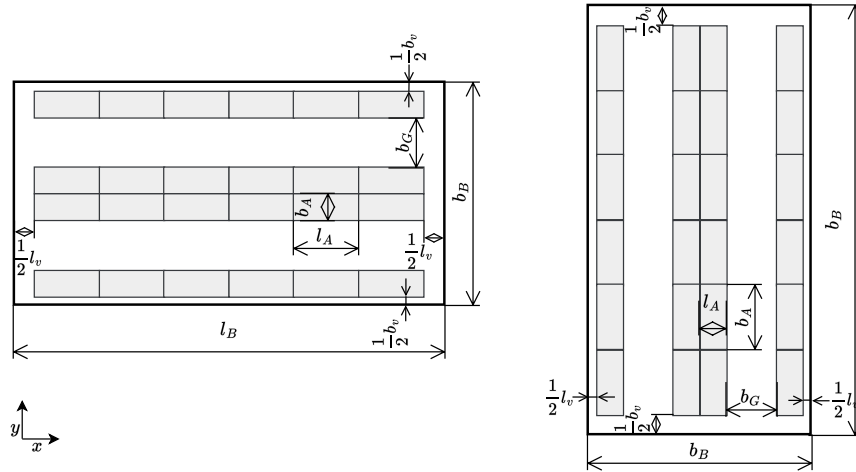


Abb. 5.10 Parameter für die Bestimmung der Position von Assets in Zeilen

$$y_O = y_B - \frac{b_B}{2} + \frac{b_v}{2} + \left(j - \frac{1}{2}\right) * b_A + (1 - \text{mod } \frac{j}{2}) * b_G \quad (5.10)$$

$$\forall j \in \left\{1, \dots, \left\lfloor \frac{b_B}{b_A + b_G/2} \right\rfloor\right\}$$

Verlaufen die Gänge in Richtung der  $y$ -Achse, lassen sich die Koordinaten eines Assets ausgehend vom Mittelpunkt des zu füllenden Bereichs  $B$  wie folgt berechnen:

$$x_O = x_B - \frac{l_B}{2} + \frac{l_v}{2} + \left(i - \frac{1}{2}\right) * l_A + (1 - \text{mod } \frac{i}{2}) * b_G \quad (5.11)$$

$$\forall i \in \left\{1, \dots, \left\lfloor \frac{l_B}{l_A + b_G/2} \right\rfloor\right\}$$

$$y_O = y_B - \frac{b_B}{2} + \frac{b_v}{2} + \left(j - \frac{1}{2}\right) * b_A \quad \forall j \in \left\{1, \dots, \left\lfloor \frac{b_B}{b_A} \right\rfloor\right\} \quad (5.12)$$

Die Orientierung der Assets hängt von der ursprünglichen Orientierung ab, die sie bei der Ablage in der Assetbibliothek haben. Für die Anordnung in Zeilen muss sie in der Richtung, die quer zu den Gängen verläuft, jedes zweite mal um 180 Grad gedreht werden.

Zeilenanordnungen können im Kopfganglayout und im Zentralganglayout konfiguriert sein. Die Generierungslogik ist für beide gleich. Beim Zentralganglayout wird der Bereich in zwei kleinere Bereiche mit Zentralgang dazwischen eingeteilt und jeweils mit Zeilen an Assets gefüllt.

### *Anordnung im Block*

Die Platzierung von Assets im Block kommt typischerweise bei gleichartigen logistischen Objekten zum Einsatz. Beispielsweise werden große Mengen des gleichen Artikels auf Paletten häufig im Block angeordnet. Die Anordnung im Block kann sowohl bei Regallagerung als auch bei der Lagerung auf dem Boden zum Einsatz kommen. Immer häufiger aufkommende Cube-Storage-Systeme sind Beispiele für die regalbasierte Blocklagerung von Kleinteilen.

Die Berechnung der Position von gleichartigen Assets im Block erfolgt wie in Gleichung 5.5 und Gleichung 5.6 für die Gebäudekacheln beschrieben. Im Falle der Anordnung von Assets im Block ist es nicht relevant, in welche Richtung die Gänge im Funktionsbereich verlaufen.

### *Zufällige Anordnung*

Bei der zufälligen Anordnung von Assets in einem Bereich  $B$  kommt die gleiche Generierungslogik zum Einsatz wie bei der zufälligen Platzierung von Funktionsbereichen im Inneren des Materialflusssystems (Gleichung 5.1, Gleichung 5.2). Die Grenzwerte für  $[x^{min}, x^{max}]$  und  $[y^{min}, y^{max}]$  sind dabei abhängig von der Länge und Breite des zu füllenden Bereiches. Zusätzlich ist bei der zufälligen Anordnung darauf zu achten, dass keine Assets, die sich bereits im System befinden, überschritten werden.

Im Folgenden werden die spezifischen Generierungslogiken für die Funktionsbereiche in den Lern- und Testumgebungen beschrieben.

## **Wareneingang und Warenausgang**

Da die Funktionsbereiche Wareneingang und Warenausgang vergleichbare Funktionen erfüllen und dadurch einen ähnlichen strukturellen Aufbau haben, werden sie hier gemeinsam betrachtet. Die semantische Modellierung des Wareneingangs und des Warenausgangs inklusive der technischen Ausstattung, die zur Durchführung der Prozesse benötigt wird, ist in Abbildung 5.11 dargestellt. Im Vergleich zu anderen Funktionsbereichen haben Warenein- und Warenausgang zwei zusätzliche Eigenschaften, da sie die Randbedingung erfüllen müssen, dass sie Zugang zu einer Wand benötigen. *Seite an Wand* vermerkt, welche Seite des Funktionsbereichs bei seiner Initialisierung an einer Wand des Gebäudes liegt. Im Anschluss an seine Initialisierung lässt der Generierungsalgorithmus die Funktionsbereiche wachsen, sodass anschließend eine weitere Seite des Funktionsbereichs an einer anderen Wand des Gebäudes liegen kann. Die Seite, an der die Ent- bzw. Verladetore eingefügt werden, wird mit dem Wert *Seite mit Toren* abgebildet.

Zur technischen Ausstattung gehören im Wareneingang die Ladetore, die sich der Gebäudeinfrastruktur zuordnen lassen, ein Zwischenlager zur Bereitstellung von Ladeeinheiten, mindestens ein Arbeitsplatz für die Vereinnahmung und den Versand sowie ein Umschlagmittel, mit dem die La-

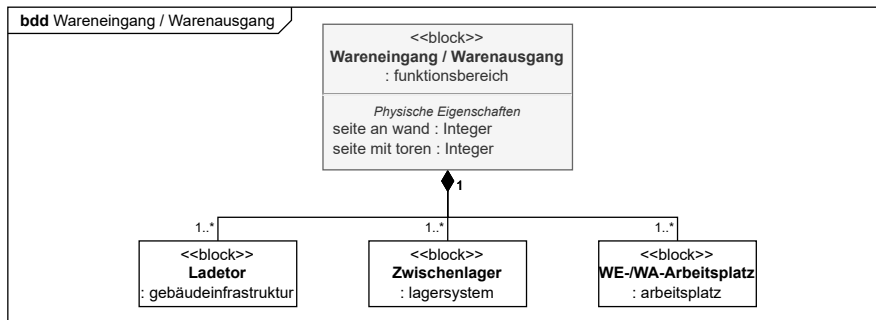


Abb. 5.11 Technische Ausstattung von Wareneingang und Warenausgang in einer Lern- und Testumgebung

deinheiten ent- und verladen werden können. Das Umschlagmittel wird über die mobilen Objekte im Materialflusssystem abgebildet, da es nicht zur statischen Infrastruktur gehört.

Die im Generierungsalgorithmus zugrunde gelegte Anordnung der Assets der technischen Ausstattung in Warenein- und Warenausgang ist in Abbildung 5.12 dargestellt.

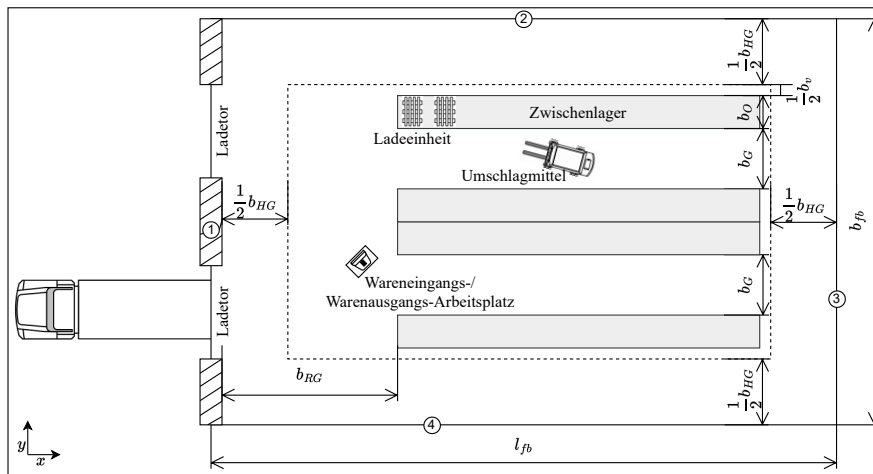


Abb. 5.12 Schematische Darstellung der Funktionsbereiche Wareneingang und Warenausgang in einer Lern- und Testumgebung

Die Fläche, die im Wareneingang und Warenausgang für das *Zwischenlager* zur Verfügung steht, wird durch den Bereich für die Hauptgänge rund um den Funktionsbereich (siehe gestrichelte Linie in Abbildung 5.12) und den Rangiergang vor den Ver- und Entladetoren eingeschränkt. Der Rangiergang dient der Einrichtung von genügend Rangiermöglichkeiten und ggf. von Überladebrücken vor den Ver- und Entladetoren, die den Höhenunterschied zwischen dem Gebäude und dem Transportfahrzeug ausgleichen (vgl. [VDI

2199, S. 22 ff.]). Auch hier kann die Gangbreite variiert werden: Typischerweise liegt die Breite des Rangiergangs  $b_{RG}$  bei 2 m bis 4 m [VDI 2199, S. 24]. Das Zwischenlager wird in der übrigen Fläche des Wareneingangs und Warenausgangs eingerichtet, die durch ihre Länge und Breite sowie ihren Mittelpunkt vollständig beschrieben ist. In dem Fall, dass die Tore an Seite 1 oder Seite 3 eingerichtet sind (siehe Nummerierung in Abbildung 5.12), steht folgende Fläche für das Zwischenlager zur Verfügung:

$$l_B^{WE/WA} = l_{FB} - b_{RG} - \frac{b_{HG}}{2} - b_{offset} \quad (5.13)$$

$$b_B^{WE/WA} = b_{FB} - b_{HG} - 2b_{offset} \quad (5.14)$$

$b_{offset}$  ist dabei ein Faktor, der die Markierung rund um den Funktionsbereich und einen Abstand bis zum Start des Equipments des Zwischenlagers einbezieht. Der Mittelpunkt der Fläche für das Lagersystem entspricht in  $y$ -Richtung dem des Funktionsbereichs, und entlang der  $x$ -Achse wird er um  $(b_{RG} - b_{HG}/2)/2$  zur Seite gegenüber der Tore verschoben.

Das Equipment und die Anordnung, die im Zwischenlager im Wareneingang und Warenausgang häufig zum Einsatz kommen, ist die Bodenzeilenlagerung, bei der die Gassen in Verlängerung der Aus- und Einfahrten der Tore verlaufen. Diese Art der technischen Ausstattung erlaubt ein schnelles Ent- und Verladen der Ladeeinheiten mit direktem Zugriff auf den Lagerplatz und minimaler Vertikalfahrt des Umschlagmittels und ist die technische Ausstattung, die in dem hier entwickelten Generierungsalgorithmus umgesetzt ist.

Ein Faktor, der bei der Generierung des Bodenzeilenlagers im Wareneingang und Warenausgang randomisiert wird, ist die Gangbreite  $b_G$ . Wie im Abschnitt *Hauptgang* beschrieben, können realistische Gangbreiten in Materialflusssystemen zwischen 2 m und 6,5 m liegen, wobei die Breite des eingesetzten Flurfördermittels der bestimmende Faktor ist. Zur Be- und Entladung im Wareneingang und Warenausgang werden häufig Unstetigfördermittel wie ein Handgabelhubwagen oder Gabelstapler eingesetzt. Um einen reibungslosen Ablauf mit einem Front- oder Schubmaststapler zu garantieren, werden Gangbreiten zwischen 2,5 m und 4 m angenommen, die typisch für Breitganglager sind.

Die Randomisierung der Gangbreiten in Zeilenlagern ist übertragbar auf alle Funktionsbereiche, in denen die Zeilenanordnung für Lagerequipment gewählt wird.

**Randomisierungsfaktor 5: Gangbreiten im Funktionsbereich bei Zeilenlagerung**

Die Breite der Gänge im Zeilenlager wird gleichverteilt zufällig zwischen zwei Grenzen  $[b_G^{\min}, b_G^{\max}]$  gewählt, die reale Richtlinien und Normen erfüllen. Durch die Variation der Breite werden die Randbedingungen der Verkehrssituationen, die in Funktionsbereichen auftreten können, verändert und die Lern- und Testaufgaben so in verschiedenen Umgebungen, die in der Realität auftreten können, durchgeführt.

$$b_G \sim U(b_G^{\min}, b_G^{\max}) \quad (5.15)$$

Neben der Lagertechnologie sind für Handhabungsaufgaben auch die operativen logistischen Objekte in den Lern- und Testumgebungen von Relevanz. Daher wird im Anschluss an die Generierung der technischen Ausstattung des Zwischenlagers dieses entsprechend dem Füllgrad des Funktionsbereichs,  $\epsilon_{WE/WA}$ , mit *Ladeeinheiten* gefüllt. Dazu wird die Lagerfläche in den Lagermitteln entsprechend den Abmessungen der Ladeeinheiten in Stellplätze eingeteilt. Alle Stellplätze  $SP_{WE/WA}$ , die im Lagersystem zur Verfügung stehen, werden iterativ durchlaufen und die Wahrscheinlichkeit für die Belegung des Platzes  $p$  wird berechnet:

$$P(p = 1) = \epsilon \quad \forall p \in SP_{fb} \quad (5.16)$$

Wenn der Platz mit einer Ladeeinheit belegt wird, wird das Asset für die Ladeeinheit zufällig aus einer Menge an für das Lagersystem qualifizierten Ladeeinheiten ausgewählt, ähnlich der Wahl der Gebäudekacheln aus einer qualifizierten Menge (Gleichung 5.7). Die Qualifikation einer Ladeeinheit wird durch das eingesetzte Equipment bestimmt (Paletten in Palettenlagern und Kleinteile in Kleinteilelagern).

In realen, manuellen Lagersystemen, wie es bei der Bodenzeilenlagerung im Wareneingang und Warenausgang der Fall ist, sind Ladeeinheiten in der Regel nicht absolut gerade ausgerichtet, sondern variieren leicht in ihrer Position und Orientierung. Dieser Faktor wird bei der Generierung der Materialflusssysteme ebenfalls berücksichtigt: Bei der Platzierung der Ladeeinheiten wird eine leichte Randomisierung der Position und Orientierung integriert, um die Randbedingungen für Handhabungsaufgaben von AMR realistischer zu gestalten.

Das Füllen der Lagersysteme mit Ladeeinheiten ist somit ein weiterer Faktor, durch den die Struktur der Materialflusssysteme in den Lern- und Testumgebungen randomisiert wird. Für die Lern- und Testaufgaben von AMR in den generierten Umgebungen ist dies relevant, da die AMR das Handling von Ladeeinheiten unter verschiedenen Randbedingungen erlernen.

**Randomisierungsfaktor 6: Füllen der Lagersysteme mit Ladeeinheiten**

- Die Stellplätze, die in einem Lagersystem mit Ladeeinheiten belegt sind, werden randomisiert aufgrund des Füllgrads des Lagersystems bestimmt (siehe Gleichung 5.16).
- Aus einer Menge an  $n$  qualifizierten Assets für Ladeeinheiten  $LE = \{le_1, \dots, le_n\}$  wird gleichverteilt zufällig ausgewählt und somit die Belegung der Lagersysteme randomisiert.

$$P(le) \sim U(1, n) \quad \forall le \in LE \quad (5.17)$$

- Die Position und Orientierung der Ladeeinheiten im Lagersystem werden leicht randomisiert verändert, um die manuelle Einlagerung realistisch abzubilden.

Der *Arbeitsplatz* im Wareneingang und Warenausgang ist häufig in Form einer mobilen Arbeitsstation mit Computer, Monitor, Scanner und Drucker gestaltet. Da die Arbeitsstation sich an einem beliebigen Ort im Funktionsbereich befinden kann, wird eine zufällige Position und Orientierung innerhalb des Funktionsbereichs gewählt (siehe Unterunterabschnitt 5.2.2.4).

Das Umschlagmittel im Wareneingang und Warenausgang ist in vielen Fällen ein Unstetigfördermittel wie ein Handgabelhubwagen oder ein Gabelstapler, die nicht exklusiv im Wareneingang oder Warenausgang eingesetzt werden. Da sie an einer beliebigen Stelle im Materialflusssystem verortet sein können, werden sie über *mobile Objekte* abgebildet (siehe Unterunterabschnitt 5.2.2.4).

**Lagerung und Kommissionierung**

Im Funktionsbereich Lagerung und Kommissionierung werden die Kernfunktionen eines Materialflusssystem erfüllt: das Lagern im Sinne der Aufbewahrung von Waren und das Kommissionieren im Sinne des Zusammenstellens von Teilmengen dieser Waren zu kundenspezifischen Aufträgen. Prozessual werden beide Aufgaben häufig getrennt voneinander betrachtet, strukturell finden sie jedoch in einem Funktionsbereich statt.

Abbildung 5.13 zeigt das semantische Modell der Elemente der technischen Ausstattung des Funktionsbereichs Lagerung und Kommissionierung, wie sie in dieser Arbeit modelliert sind.

Die technische Ausstattung des Funktionsbereichs besteht aus mindestens einem Lagersystem, ggf. Arbeitsplätzen und einem Fördersystem.

Die Art der Lagersysteme, die für die Generierung von Lern- und Testumgebungen für AMR in den Funktionsbereichen der Lagerung und Kommissionierung relevant sind, sind insbesondere Paletten- und Kleinteilelager. Eine Unterscheidung der Funktionsbereiche in diese Kategorien im generierten

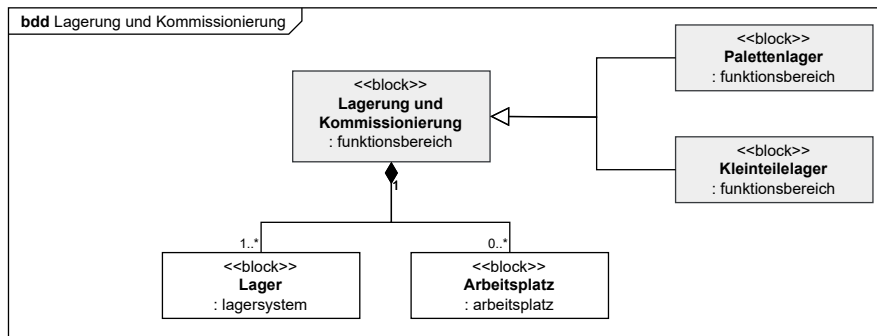


Abb. 5.13 Technische Ausstattung der Lagerung und Kommissionierung in einer Lern- und Testumgebung

Simulationsmodell ist sinnvoll, um die Vielfalt der Lern- und Testumgebungen zu erhöhen und AMR die Möglichkeit zu geben, sowohl das Handling von Kleinteilen wie Boxen als auch das Handling von Paletten zu erlernen. Bis auf die gewählten Assets für die Abbildung des Lagermittels und der eingelagerten Ladeeinheiten ergeben sich für den Generierungsalgorithmus jedoch keine Unterschiede.

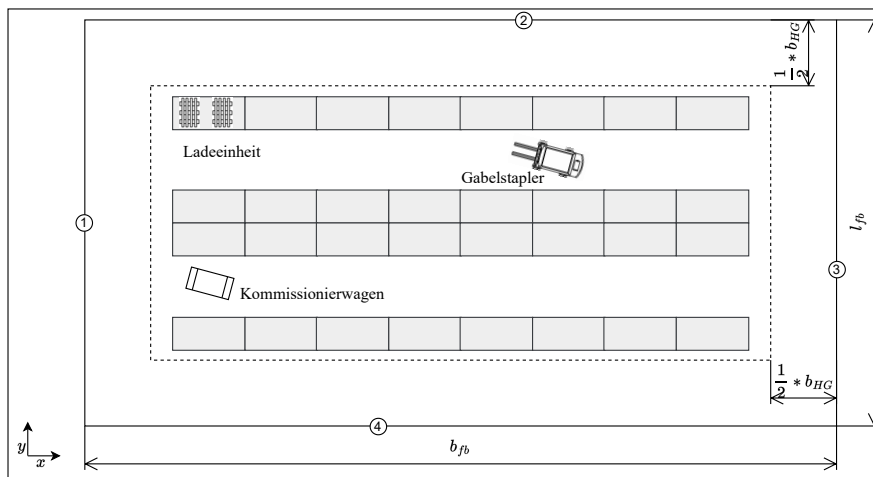
Die Arbeitsplätze in einem Lager- und Kommissionierbereich lassen sich, ähnlich wie im Wareneingang und Warenausgang, in statische und mobile Arbeitsplätze unterscheiden. In manuellen Regallagern sind mobile Arbeitsplätze üblich. In manuellen Palettenlagern erfolgt die Kommissionierung beispielsweise von einer statisch bereitgestellten Palette auf eine Palette auf einem Gabelstapler oder einem Hubwagen. In manuellen Kleinteilelagern können Kommissionierwagen als mobile Arbeitsplätze zum Einsatz kommen, auf denen Artikel eines oder mehrerer Aufträge zusammengestellt werden.

Fördermittel werden im Bereich Lagerung und Kommissionierung demnach zum einen zur Einlagerung oder Bereitstellung zur Einlagerung und zum anderen für die Kommissionierung beziehungsweise den Transport zum nächsten Funktionsbereich eingesetzt. Sie können mobil im Materialflusssystem eingesetzt werden und sind daher den mobilen Objekten zugeordnet.

Abbildung 5.14 zeigt die schematische Darstellung eines Lager- und Kommissionierbereichs, wie er in dem in dieser Arbeit entwickelten Algorithmus generiert wird. Das *Lagersystem* nimmt den Hauptteil der Fläche ein. Der gesamte Funktionsbereich wird mit einem Band umrandet, das der halben Breite des Hauptgangs entspricht, über den der Funktionsbereich mit dem Hauptverkehrsnetz des Materialflusssystems verbunden ist. Nach Abzug des Faktors  $b_{offset}$  ist die Fläche, die für das Lagersystem zur Verfügung steht, wie folgt:

$$l_B = l_{FB} - b_{HG} - 2b_{offset} \quad (5.18)$$

$$b_B = b_{FB} - b_{HG} - 2b_{offset} \quad (5.19)$$



**Abb. 5.14** Schematische Darstellung des Funktionsbereichs Lagerung und Kommissionierung in einer Lern- und Testumgebung

Der Mittelpunkt der Fläche für das Lagersystem entspricht dem des Funktionsbereichs.

Sowohl in manuellen Paletten- als auch in manuellen Kleinteilelagern kann das Lagergut in Regalen gelagert werden. Diese sind in der Realität in Zeilen angeordnet, wie in Abbildung 5.14 abgebildet. Auch im Funktionsbereich Lagerung und Kommissionierung können die Gangbreiten randomisiert werden (Randomisierungsfaktor 5). In Palettenlagern liegt die Gangbreite wie im Wareneingang und Warenausgang typischerweise zwischen 2,5 m und 4 m, da in diesen Gängen Raum für das Rangieren mit dem Gabelstapler eingeplant werden muss. In Kleinteilelagern, in denen mit Handgabelhubwagen oder Kommissionierwagen gearbeitet wird, ist die Gangbreite mit typischerweise 0,8 m bis 1,2 m geringer.

Zusätzlich kann im Funktionsbereich Lagerung und Kommissionierung die Ausrichtung der Gänge randomisiert werden. Sie können sowohl in Richtung der  $x$ -Achse als auch in Richtung der  $y$ -Achse verlaufen. Durch die Veränderung der Ausrichtung entstehen verschiedene Wegenetze im Materialflusssystem, mit denen die digitalen Abbilder der AMR umzugehen lernen. Diese Veränderung ist zum Beispiel für Aufgaben aus dem Bereich Navigation relevant.

**Randomisierungsfaktor 7: Ausrichtung der Gänge bei Zeilenlagerung**

In Funktionsbereichen, in denen die Objekte in Zeilen angeordnet sind, gibt es zwei Optionen für die Ausrichtung der Gänge: [ $g_x$  = Ausrichtung an x-Achse,  $g_y$  = Ausrichtung an y-Achse]. Die Wahl der Ausrichtung der Gänge wird gleichverteilt zufällig getroffen.

$$P(g_x) = P(g_y) = 0,5 \quad (5.20)$$

Ein zusätzlicher Randomisierungsfaktor sind die Assets, die zur Darstellung der technischen Ausstattung verwendet werden. Innerhalb der Regallagerung können die Regale im Paletten- und Kleinteilelager unterschiedliche Abmessungen, Fachhöhen, Anzahlen von Traversen bzw. Fachböden sowie verschiedene Farben aufweisen. All diese Faktoren beeinflussen sowohl die Struktur des Lagers – und damit die Wege, die von AMR befahren werden müssen, in horizontaler und vertikaler Richtung – als auch die Wahrnehmung der Umgebung. Aus den für ein spezifisches Lagersystem qualifizierten Assets wird eines zufällig ausgewählt, das anschließend wiederholt in Zeilen in das Simulationsmodell eingefügt wird. Die Qualifikation eines Assets für ein Lagersystem hängt von der Kompatibilität des Lagerguts mit dem Lagermittel ab (Paletten in Palettenregalen und Kleinteile in Fachbodenregalen).

**Randomisierungsfaktor 8: Wahl der Assets für Lagermittel**

Aus einer Menge von  $n$  qualifizierten Assets zur Darstellung des Lagermittels  $LM = \{lm_1, \dots, lm_n\}$  wird gleichverteilt zufällig ausgewählt und somit die Struktur und Darstellung des Lagersystems randomisiert.

$$P(lm) \sim U(1, n) \quad \forall lm \in LM \quad (5.21)$$

Genau wie im Zwischenlager des Wareneingangs und Warenausgangs werden die Objekte der Lagersysteme im Funktionsbereich Lagerung und Kommissionierung anschließend mit *Ladeeinheiten* gefüllt. Es werden auch hier pro Lagerplatz und entsprechend dem Lagerfüllgrad qualifizierte Ladeeinheiten zufällig aus einer Objektbibliothek ausgewählt (Randomisierungsfaktor 6).

Schließlich zählen die *Arbeitsplätze* im Bereich manueller Lager- und Kommissioniersysteme noch zur technischen Ausstattung dieser Funktionsbereiche. Sie sind häufig in Form von Kommissionierwagen oder Kommissionierstaplern gestaltet. Da diese mobilen Arbeitsplätze sich an einem beliebigen Ort im Funktionsbereich befinden können, wird für sie eine zufällige Position und Orientierung innerhalb des Funktionsbereichs gewählt (siehe Unterunterabschnitt 5.2.2.4).

## Verpackung

Die technische Ausstattung von Verpackungsbereichen besteht aus mindestens einem Packplatz (siehe Abbildung 5.15). Ein manueller Packplatz ist ausgestattet mit einem Arbeitstisch, auf und unter dem Packmittel wie Kartonen und Packhilfsmittel wie Klebeband bereitgestellt sind. Zudem sind die Arbeitsplätze häufig mit einer Anbindung an das führende IT-System ausgestattet, sodass Informationen zur Verpackung eines Auftrags abgerufen und auf einem Monitor angezeigt werden können. Rund um den Arbeitsplatz sind meist Stellflächen für Ladeeinheiten eingeplant.

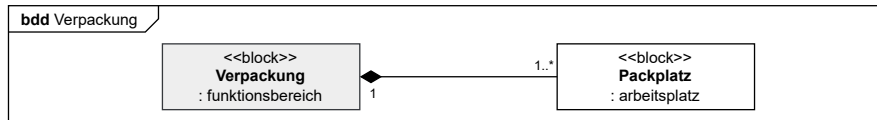


Abb. 5.15 Technische Ausstattung des Verpackungsbereichs in einer Lern- und Testumgebung

Abbildung 5.16 zeigt den schematischen Aufbau des Packbereichs, wie er durch den in dieser Arbeit entwickelten Generierungsalgorithmus gestaltet wird. Die *Arbeitsplätze* sind gruppierte Assets, die entsprechend der gewählten Anordnungsart in den Funktionsbereich eingesetzt werden. Für die Arbeitsplätze wird die Zeilenanordnung ausgewählt. Die Fläche, die für die Arbeitsplätze zur Verfügung steht, ist wie im Bereich Lagerung und Kommissionierung die Fläche, die nach Abzug der halben Breite des Hauptgangs an allen vier Seiten noch zur Verfügung steht. Sie ist beschrieben durch die Länge und Breite entsprechend Gleichung 5.18 und Gleichung 5.19.

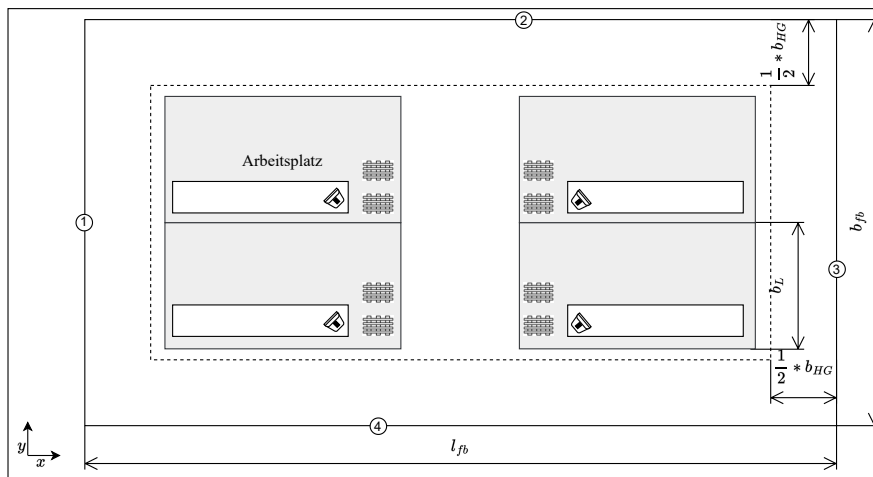


Abb. 5.16 Schematische Darstellung des Funktionsbereichs Verpackung in einer Lern- und Testumgebung

Im Verpackungsbereich kommen drei bereits beschriebene Randomisierungsfaktoren zum Einsatz: die zufällige Wahl der Gangbreite zwischen zwei Arbeitsplätzen (Randomisierungsfaktor 5), die zufällige Wahl der Assets zur Darstellung der Ladeeinheiten (in diesem Fall Verpackungsarbeitsplätze, die nach der gleichen Logik generiert werden; Randomisierungsfaktor 6) und die zufällige Wahl der Achse, in deren Richtung die Gänge verlaufen (Randomisierungsfaktor 7). Die Gangbreiten zwischen den Arbeitsplätzen orientieren sich dabei an den Breiten des Palettenregals, da auch im Packbereich Gabelstapler rangieren können.

### Ladestation

Der Funktionsbereich der Ladestationen ist kein klassischer Funktionsbereich, wie er in Lehrbüchern in einem Materialflusssystem verortet ist. Dennoch ist es ein wichtiger Funktionsbereich, insbesondere in Systemen, die mit AMR betrieben werden. Diese Roboter und auch die meisten modernen manuell geführten Förderzeuge verfügen über elektrische Antriebe. Je nach Kapazität des Akkus müssen sie aufgeladen werden, häufig zum Ende einer Schicht. Zur technischen Ausstattung des Funktionsbereichs Ladestation gehört daher mindestens ein Ladeplatz (siehe Abbildung 5.17). Ebenso wie der Wareneingang und der Warenausgang muss der Funktionsbereich Ladestation an einer Wand des Materialflusssystems liegen. In realen Materialflusssystemen liegen die Ladestationen für elektrische Geräte wie Gabelstapler und Roboter häufig an einer Wand, da die Leitungen zur Versorgung der Stationen mit Strom so leichter zu erreichen sind. Außerdem sind die Stationen laut [VDI 2199, S. 37] leicht zugänglich und befahrbar zu gestalten, die Bereiche sollen gut lüftbar sein, um potenziell austretende Gase abzuführen, und das Eindringen von schädlichen Gasen, Dämpfen und Staub ist zu verhindern. All diese Punkte werden durch eine geschützte Position an der Wand und in der Nähe von Fenstern am ehesten erfüllt. Daher hat der Bereich Ladestation die zusätzliche Eigenschaft *Seite an Wand*, über die die Seite des Bereichs vermerkt wird, die an der Wand liegt.

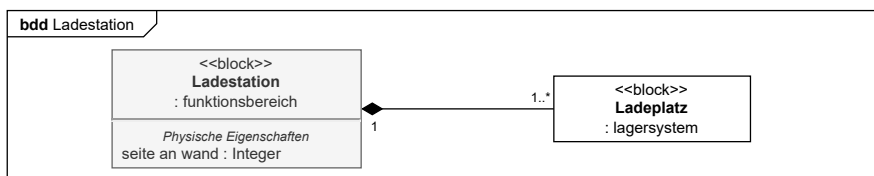


Abb. 5.17 Technische Ausstattung der Ladestation in einer Lern- und Testumgebung

Abbildung 5.18 zeigt eine schematische Darstellung des Funktionsbereichs Ladestation, wie er in diesem Generierungsalgorithmus erzeugt wird. Die Ladeplätze im Funktionsbereich Ladestation sind jeweils mit einer Ladesäule ausgestattet. Sie sind in einer Zeile entlang der Wand angeordnet. Um eine

realitätsnahe Lern- und Testumgebung zu generieren, werden zu ladende Fördermittel in den Funktionsbereich generiert. Dieser Generierungsschritt ist vergleichbar mit dem Füllen von Bodenlagerplätzen mit Ladeeinheiten und weist daher die zufälligen Faktoren aus Randomisierungsfaktor 6 auf.

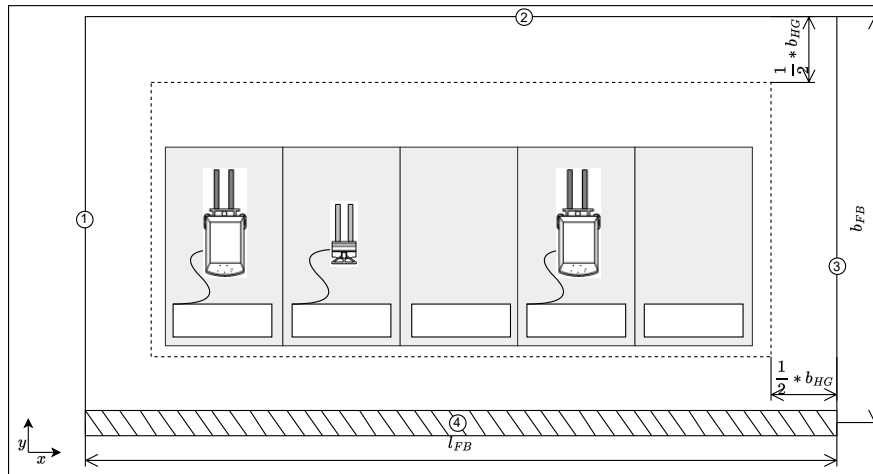


Abb. 5.18 Schematische Darstellung des Funktionsbereichs Ladestation in einer Lern- und Testumgebung

#### 5.2.2.4 Mobile Objekte

Neben den Elementen, die zur technischen Ausstattung eines Funktionsbereiches gehören und diesem fest zugeordnet sind, gibt es in einem Materialflusssystem mobile Objekte, die in verschiedenen Funktionsbereichen zum Einsatz kommen und sich daher an beliebigen Orten befinden können. Diese mobilen Objekte werden in den generierten Lern- und Testumgebungen an zufällig gewählten Orten im gesamten System oder einem Teil des Systems generiert. Das Ziel dabei ist es, dass die Objekte realistische Hindernisse für die zu trainierenden oder zu testenden Roboter bieten, denn auch in der Realität können diese Elemente Robotern an verschiedenen Orten begegnen und zum Beispiel einen zuvor geplanten Weg versperren. Trotz ihres im physischen System mobilen Charakters werden die Elemente in den Lern- und Testumgebungen statisch abgebildet. Sie verändern ihre Eigenschaften demnach nicht mit fortschreitenden Physikschritten der Simulation.

In Abbildung 5.19 ist die Modellierung mobiler Objekte inklusive ihrer Eigenschaften dargestellt. Es können verschiedene *Arten* mobiler Objekte generiert werden. Jedes mobile Objekt hat eine zufällig gewählte *Position* und eine *Orientierung*, mit denen sie generiert werden. Die Abmessungen, also *Länge*, *Breite* und *Höhe* des Objektes, sind relevant, um bei der zufälligen Generierung sicherstellen zu können, dass das Objekt kein bereits existierendes

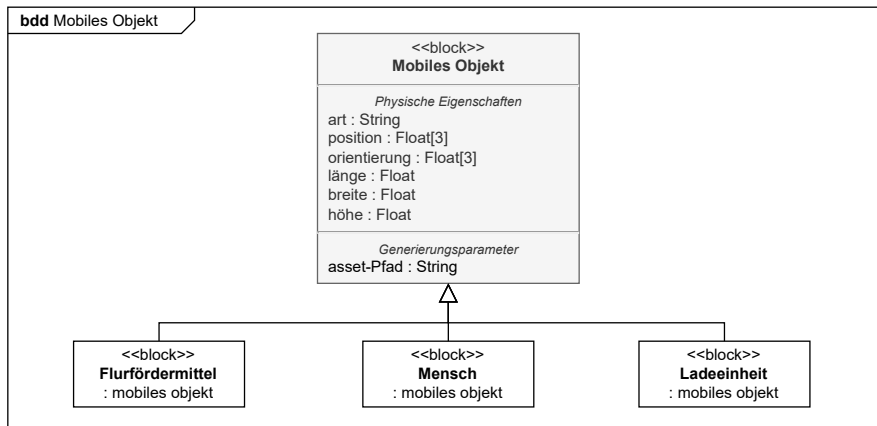


Abb. 5.19 Modellierung mobiler Objekte in einer Lern- und Testumgebung

Objekt überschneidet. Der *Asset-Pfad* schließlich gibt den Pfad zur Bibliothek der Assets an, aus dem das Objekt generiert werden soll.

Die mobilen Objekte in den generierten Materialflusssystemen können eingeteilt werden in die Kategorien *Fördermittel* wie Gabelstapler und Kommissionierwagen, *Personen* und einzelne *Ladeeinheiten*. Diese Objekte sind entweder keinem Funktionsbereich fest zugeordnet und können sich daher an einer beliebigen Stelle im Materialflusssystem befinden, oder sie können in einem Funktionsbereich an einer zufälligen Position generiert werden.

Die Objekte werden bei der Generierung zufällig aus einer Menge an qualifizierten Assets gewählt. Für die Qualifikation verantwortlich ist dabei die Art des Objektes: Soll ein Fördermittel randomisiert in einem Bereich generiert werden, wird zufällig ein Asset eines Fördermittels aus der Menge aller verfügbaren Assets ausgewählt.

#### Randomisierungsfaktor 9: Auswahl der mobilen Objekte

Aus einer Menge von  $n$  qualifizierten Assets zur Darstellung eines mobilen Objektes  $MO = \{mo_1, \dots, mo_n\}$  wird gleichverteilt zufällig ausgewählt und somit die Struktur und Darstellung des Materialflusssystems randomisiert.

$$P(mo) \sim U(1, n) \quad \forall mo \in MO \quad (5.22)$$

Die Positionen und Orientierungen der Objekte werden zufällig gewählt und stellen einen weiteren Randomisierungsfaktor in dem Generierungsalgorithmus dar.

**Randomisierungsfaktor 10: Position und Orientierung mobiler Objekte**

Die Position und die Orientierung eines mobilen Objektes werden so lange zufällig zwischen vorgegebenen Grenzwerten gewählt, bis sich das Objekt an einem Ort befindet, an dem es keine weiteren Objekte überschneidet.

Im Anschluss an die Generierung der mobilen Objekte in den Materialflusssystemen ist die Generierung der Struktur und der statischen Elemente in den Lern- und Testumgebungen abgeschlossen. Es folgt die Initialisierung der Aufgabe, die während des Simulationslaufs durch digitale Abbilder von AMR erlernt oder getestet werden soll, bevor die Simulation gestartet wird und die nächste Phase des Generierungsverfahrens, die Generierung der Materialflüsse, startet.

### 5.3 Generierung der Materialflüsse in Lern- und Testumgebungen

Das Verhalten der Materialflusssysteme in der Lern- und Testumgebung wird über die Transportaufträge im System abgebildet. Transportaufträge steuern das Verhalten von Materialflusssystemen. Sie resultieren direkt aus dem Kundenauftragsprofil und zeigen die Last, die auf einem System liegt. Die Leistung eines logistischen Systems wird beispielsweise durch den Durchsatz bemessen, der sich aus abgearbeiteten Transportaufträgen ergibt. Transportaufträge bieten sich daher als steuernde Instanz an, um verschiedene Lastszenarien abzubilden und die Reaktion des Systems auf diese Lastszenarien zu erlernen und zu bewerten.

Transportaufträge in Materialflusssystemen enthalten die Informationen über sämtliche Bewegungen von Ladeeinheiten zwischen Lagerplätzen in den Systemen [HS10, S. 258]. Über sie werden die Transportprozesse in dem System verwaltet und gesteuert. Transportaufträge können beispielsweise Einlager-, Auslager- oder Umlageraufträge sein. Auch Kommissionieraufträge und Umpackaufträge können als Transportaufträge definiert werden. Während ihrer Zeit im System können Transportaufträge verschiedene Status erreichen. In der Regel befindet sich ein Transportauftrag im Status *freigegeben*, sobald er im System eingelastet ist, *begonnen*, sobald ein Mitarbeiter oder Automatisierungstechnik den Transportauftrag übernimmt, und *beendet*, sobald der Transportprozess durchgeführt wurde.

Die Art und Weise, wie Transportaufträge in der vorliegenden Arbeit modelliert werden, ist in Abbildung 5.20 abgebildet. Jeder Transportauftrag hat eine eindeutige *ID* und eine eindeutige *Lern- und Testumgebung*, der er zugeordnet ist. Um einen Transportauftrag durchführen zu können, sind Informationen zum *Quell-Funktionsbereich* und zur *Quell-Koordinate* von

Bedeutung. An dieser Stelle wird die Ladeinheit, für die der Transportauftrag besteht, aufgenommen und anschließend im *Ziel-Funktionsbereich* zur *Ziel-Koordinate* transportiert. Sollte es sich bei dem Transportauftrag um einen Kommissionierauftrag handeln, ist auch die Information zur *Menge* der zu verbringenden Artikel von Bedeutung. Zudem hat ein Transportauftrag Informationen zum aktuellen *Status* und zum *Freigabezeitpunkt*, über die Auswertungen zur Leistung des Systems erstellt werden können.

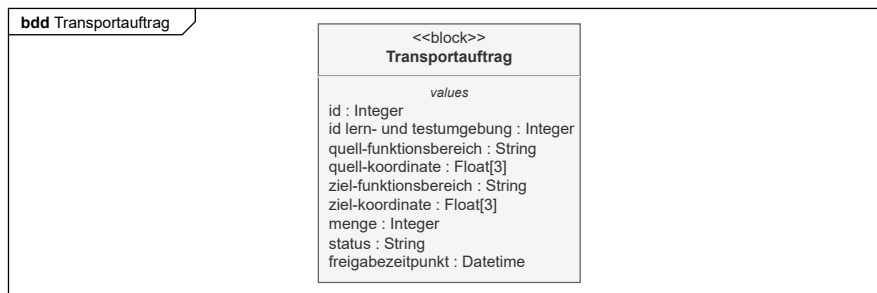


Abb. 5.20 Modellierung von Transportaufträgen in einer Lern- und Testumgebung

Die Transportaufträge werden in der vorliegenden Arbeit zentral durch einen Server erzeugt und zum Zeitpunkt ihrer Freigabe in die Lern- und Testumgebungen eingestellt. Die Art und Weise, wie diese Aufträge von den Akteuren im System abgearbeitet werden, ist bewusst nicht vorgegeben. Dadurch bleibt die konkrete Steuerungsarchitektur flexibel: In einem Szenario mit zentraler Steuerung kann ein übergeordneter Agent die Aufträge an verfügbare AMR verteilen. In Szenarien mit dezentraler, autonomer Entscheidungsfindung übernehmen die AMR die Entscheidung selbstständig, ob und wie ein Auftrag ausgeführt wird. Diese Offenheit ermöglicht es, verschiedene Steuerungsarchitekturen in den Umgebungen abbilden und untersuchen zu können.

Die Generierung von Transportaufträgen erfolgt über ein stochastisches Verfahren. Das Eintreffen neuer Aufträge wird durch eine Wahrscheinlichkeitsdichtefunktion modelliert, die die Zwischenankunftszeiten approximiert. Diese Vorgehensweise entspricht der gängigen Praxis in der Simulationsmodellierung [GG15]. Für die Zwischenankunftszeit wird in der vorliegenden Arbeit die Exponentialverteilung herangezogen, die insbesondere in der Warteschlangentheorie etabliert ist.

Die Exponentialverteilung beschreibt die Zeit zwischen unabhängigen, identisch verteilten Ereignissen in einem Poisson-Prozess und wird über die Dichtefunktion

$$f(x; \lambda) = \lambda e^{-\lambda} \quad (5.23)$$

definiert, wobei  $\lambda$  der Ratenparameter ist.

Die Poisson-Verteilung modelliert die Anzahl der Ereignisse in einem festen Zeitintervall, wobei die Zwischenankunftszeiten wiederum exponentiell verteilt sind. Für die Wahrscheinlichkeit, dass die Zeit bis zum nächsten Ereignis größer als  $t$  ist, gilt:

$$P(X > t) = e^{(-\lambda t)} \quad (5.24)$$

Da reale Ankunftsprozesse in Materialflusssystemen nicht immer diesen idealisierten Annahmen entsprechen, können auch alternative Verteilungen wie die Erlang-Verteilung oder Mischungen verschiedener Verteilungen herangezogen werden. Wenn reale Daten verfügbar sind, sollten diese zur empirischen Bestimmung der passenden Verteilung genutzt werden.

Die Basis zur Generierung der Transportaufträge bildet eine Materialflussmatrix. Sie gibt an, wie viele Aufträge pro Zeiteinheit von einem Funktionsbereich in einen anderen überführt werden. Die Werte der Matrix dienen als Mittelwert für das Sampling der Zwischenankunftszeiten.

Für jede Lern- und Testumgebung werden die Aufträge zu den gleichen Zeitpunkten eingelastet, die durch die gesampelten Zwischenankunftszeiten bestimmt sind. Dynamiken entstehen durch die stochastische Ziehung und durch die Bestimmung der Quell- und Zielkoordinaten. Die Quelle eines Transportauftrags wird durch eine zufällige Auswahl von Ladeeinheiten oder wahlweise durch eine zufällige Position innerhalb des definierten Quell-Funktionsbereichs bestimmt. Analog wird die Senke eines Transportauftrags durch die zufällige Wahl einer Position im Ziel-Funktionsbereich festgelegt. Durch dieses Vorgehen wird eine weitere Randomisierung der Transportaufträge erzeugt.

#### **Randomisierungsfaktor 11:** *Auswahl von Quellen und Zielen*

Die Quell- und Zielpositionen der Transportaufträge werden stochastisch bestimmt. Im Quell-Funktionsbereich wird zufällig eine Ladeeinheit oder eine Koordinate als Startpunkt gewählt, während im Ziel-Funktionsbereich eine zufällige Koordinate als Ziel festgelegt wird.

Die Generierung der Transportaufträge basiert somit auf einer zentralen Erzeugung von Aufträgen, deren Ankunftszeiten durch Wahrscheinlichkeitsverteilungen und eine Materialflussmatrix parametrisiert werden. Die konkrete Abarbeitung dieser Aufträge ist nicht festgelegt, sondern hängt von der gewählten Steuerungsarchitektur in der spezifischen Lern- und Testaufgabe ab. Damit lassen sich sowohl zentral koordinierte als auch dezentral autonome Steuerungsstrategien untersuchen. Durch die Kombination aus stochastischer Auftragserzeugung, Randomisierung von Quellen und Zielen

sowie flexibler Abarbeitung entsteht eine Grundlage für das Training und Testen autonomer Steuerungsstrategien im Digitalen Kontinuum.

## Kapitel 6

# Validierung

**Zusammenfassung** In diesem Kapitel wird die Validierung des entwickelten Generierungsverfahrens anhand einer Umsetzung in Nvidia Isaac Sim durchgeführt. Dazu werden die beiden zentralen Komponenten des Generierungsverfahrens, die Strukturgenerierung und die dynamische Generierung von Materialflüssen, in einer Anwendung implementiert und mit realitätsnahen Robotermodellen erprobt. Die Ergebnisse zeigen, dass variantenreiche, semantisch kohärente und physikalisch korrekte Materialflusssysteme generiert werden können, in denen Transportaufträge zur Simulationslaufzeit stochastisch bereitgestellt werden. Zusätzlich wird die Testfähigkeit robotischer Steuerungsstrategien untersucht, indem Navigationsaufgaben mit unterschiedlichen Planungsalgorithmen im ROS-Framework durchgeführt und miteinander verglichen werden. Die Reflexion der Ergebnisse verdeutlicht, dass die grundlegenden Anforderungen an Variabilität, Realitätsnähe und Automatisierungsgrad erfüllt werden, wenngleich Optimierungspotenziale hinsichtlich der Rechenlast, der weiteren Annäherung an reale Layouts und der Erweiterung um komplexere Materialflussszenarien bestehen.

## 6.1 Ziel und Vorgehen bei der Validierung

Ziel der Validierung ist es, zu bewerten, inwieweit das entwickelte Generierungsverfahren die in Abschnitt 4.1 dargestellten Anforderungen erfüllt. Dazu werden der Strukturgenerierungsalgorithmus und die Logik zur Generierung der Transportaufträge in der Simulationsumgebung Isaac Sim von Nvidia (Version 2023.1.1) umgesetzt, wie im weiteren Verlauf dieses Kapitels beschrieben. Isaac Sim ist eine auf Nvidia Omniverse basierende Plattform, die es ermöglicht, KI-gesteuerte Roboter in virtuellen, physikalisch akkuraten Umgebungen zu entwickeln, zu simulieren und zu testen (siehe [Mak+21]). Die Eignung dieser Plattform ergibt sich insbesondere aus der integrierten Physik-Engine, aus der Bereitstellung vorkonfigurierter, realistischer Robotermodelle mit ihren wesentlichen Komponenten wie Sensoren, Aktoren und Recheneinheiten, aus der Anbindung an gängige RL- und Robotiksteuerungs-Frameworks sowie aus der Verfügbarkeit optisch und physikalisch realistischer Assets aus dem Bereich der Materialflusssysteme.

Im weiteren Verlauf des Kapitels wird beschrieben, wie mit dem implementierten Generierungsalgorithmus Lern- und Testumgebungen erzeugt werden, anhand derer die Variabilität der Strukturgenerierung, die Generierung dynamischer Materialflüsse im Simulationsverlauf und die Einbindung robotischer Steuerungsalgorithmen für die Navigation demonstriert werden. Dadurch kann überprüft werden, ob alle Elemente für die Lern- und Testfähigkeit von Robotern, wie AMR es sind, in den generierten Umgebungen verfügbar sind. Abschließend erfolgt eine systematische Bewertung anhand der in Abbildung 4.1 zusammengefassten Anforderungen.

## 6.2 Implementierung des Generierungsverfahrens

Die Implementierung des entwickelten Generierungsverfahrens erfolgt als eigene Extension in Nvidia Isaac Sim. Die Architektur der Implementierung ist in Abbildung 6.1 dargestellt. Zentrales Element bildet das Modul *Stage Builder*, das als Bindeglied zwischen den beiden Generierungsalgorithmen (Struktur und Materialflüsse), der Benutzeroberfläche und der Simulationsumgebung fungiert. Darüber hinaus übernimmt der Stage Builder die Initialisierung des Weltmodells, das die Grundlage für alle weiteren Schritte bildet.

Der Stage Builder verwaltet die Eingaben der Nutzerinnen und Nutzer über die Benutzeroberfläche. Diese Eingaben können ebenso durch einen KI-Agenten im Digitalen Kontinuum vorgegeben werden. Sie bilden die Grundlage für die Strukturgenerierung, die den Aufbau der Szene vor Start der Simulation übernimmt. Dabei wird für jede angelegte Lern- und Testumgebung ein Materialflusssystem generiert. Neben den Nutzereingaben werden hierfür eine Konfigurationsdatei, in der zulässige Gestaltungsparameter wie Gangbreiten oder qualifizierte Assets für logistische Objekte in

Funktionsbereichen hinterlegt sind, sowie die Asset-Datenbank von Nvidia Omniverse zugrunde gelegt.

Vor Start des Simulationslaufs werden auch die Roboter in die Lern- und Testumgebungen integriert, ebenfalls gesteuert durch den Stage Builder. Die Roboter werden gemeinsam mit ihrer jeweiligen Lern- oder Testaufgabe initialisiert, die zugleich in der Simulationsumgebung registriert wird. Der Stage Builder übernimmt darüber hinaus die Initialisierung der Transportauftragsgenerierung.

Mit Start der Simulation beginnt einerseits die Ausführung der Lern- oder Testaufgabe der Roboter. Im Rahmen von Testaufgaben werden dabei die Sensordaten der Roboter durch die zu testenden Steuerungsstrategien direkt in Aktionen überführt. Bei Lernaufgaben hingegen werden die Daten an ein angeschlossenes Lernframework übergeben und dort für RL-basierte Lernprozesse genutzt. Parallel dazu werden durch den Transportauftragsgenerator Transportaufträge erzeugt, die sich an einer hinterlegten Materialflussmatrix orientieren. Diese Matrix definiert die Anzahl von Transporten zwischen einzelnen Funktionsbereichen und dient damit als Basis für die dynamische Materialflussgenerierung.

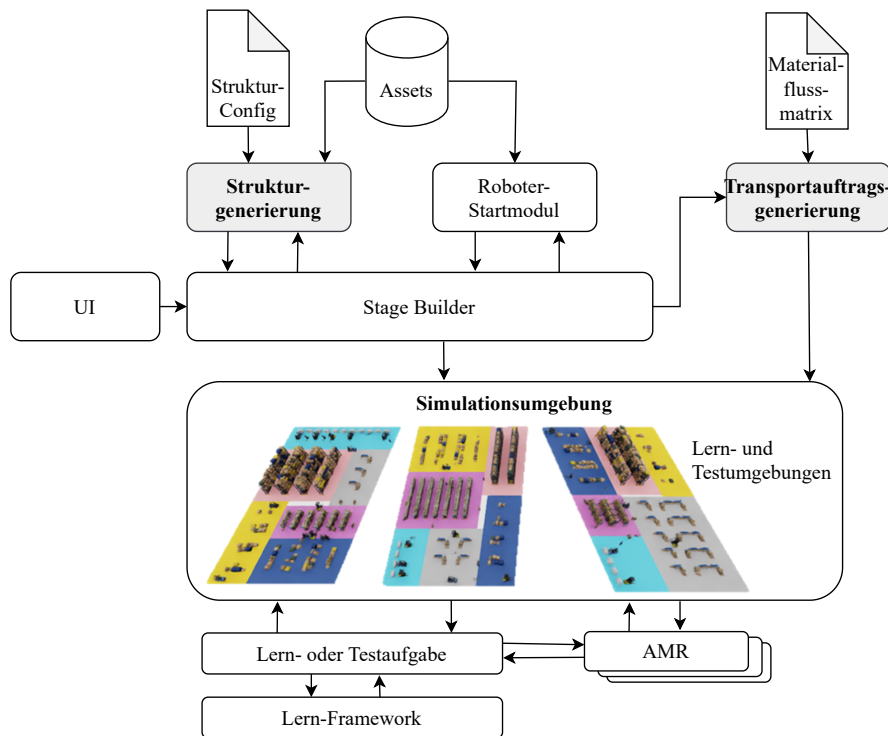


Abb. 6.1 Implementierungsarchitektur des Generierungsverfahrens in Nvidia Isaac Sim

Die Benutzeroberfläche der entwickelten Extension ist in Abbildung 6.2 dargestellt. Hier können Nutzerinnen und Nutzer die Spezifizierung der Generierungsanforderungen vornehmen. Für die Strukturgenerierung lassen sich unter anderem die Anzahl der Lern- und Testumgebungen, deren Abmessungen sowie die Anzahl der Funktionsbereiche verschiedener Typen definieren. Zusätzlich kann angegeben werden, wie viele mobile Objekte, beispielsweise Mitarbeitende, Fahrzeuge oder Ladeeinheiten, zufällig in den Umgebungen platziert werden sollen. Für die Verhaltensgenerierung können zudem die Art der eingesetzten Roboter und die Logik zur Erzeugung der Transportaufträge (an Ladeeinheiten oder zufällig im Funktionsbereich) festgelegt werden. In einem realisierten Digitalen Kontinuum werden diese Angaben durch einen KI-Agenten aus dem Kontext der aktuellen Lern- oder Testaufgabe erschlossen.

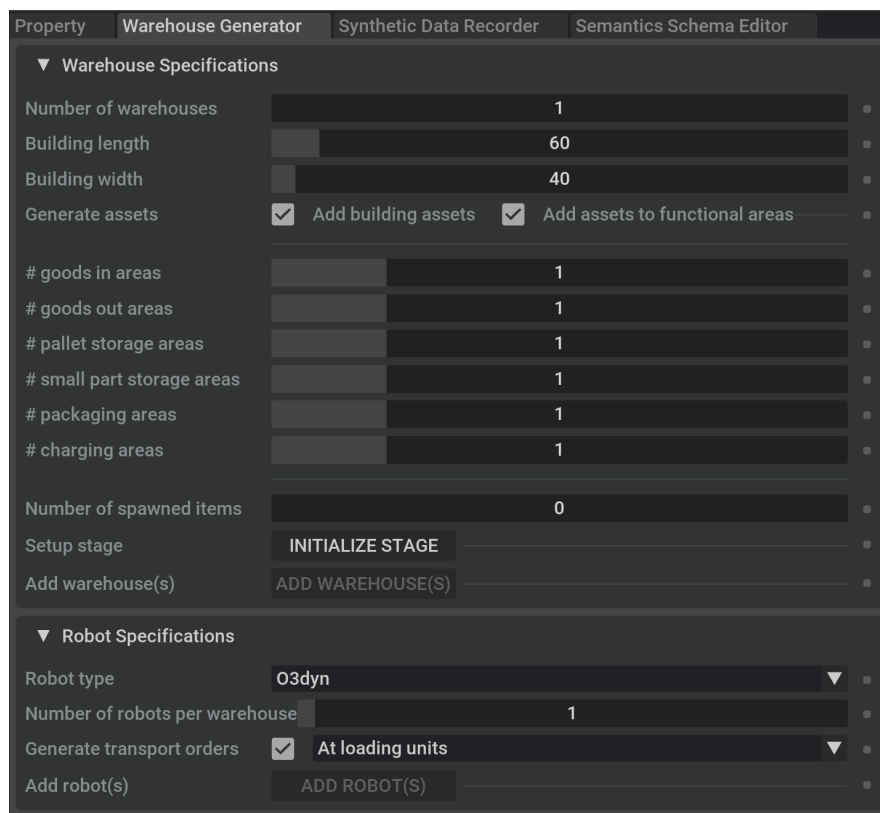


Abb. 6.2 Benutzeroberfläche der Generierungsanwendung in Nvidia Isaac Sim

Ein Beispiel für eine generierte Lern- und Testumgebung ist in Abbildung 6.3 abgebildet. Der Ausschnitt zeigt ein durch das entwickelte Generierungsverfahren automatisch erstelltes Simulationsmodell eines Materialflusssystems,

in das der Roboter O3dyn [Wie+24], entwickelt am Fraunhofer-Institut für Materialfluss und Logistik IML, integriert ist. Anhand von Umgebungen wie der dargestellten werden im Folgenden die Funktionalität der Strukturgenerierung, die Generierung dynamischer Transportaufträge und die möglichst realistische Einbindung von Robotern als Grundlage für das Lernen und Testen von Steuerungsstrategien validiert.



Abb. 6.3 Roboter in einem generierten Simulationsmodell in Nvidia Isaac Sim

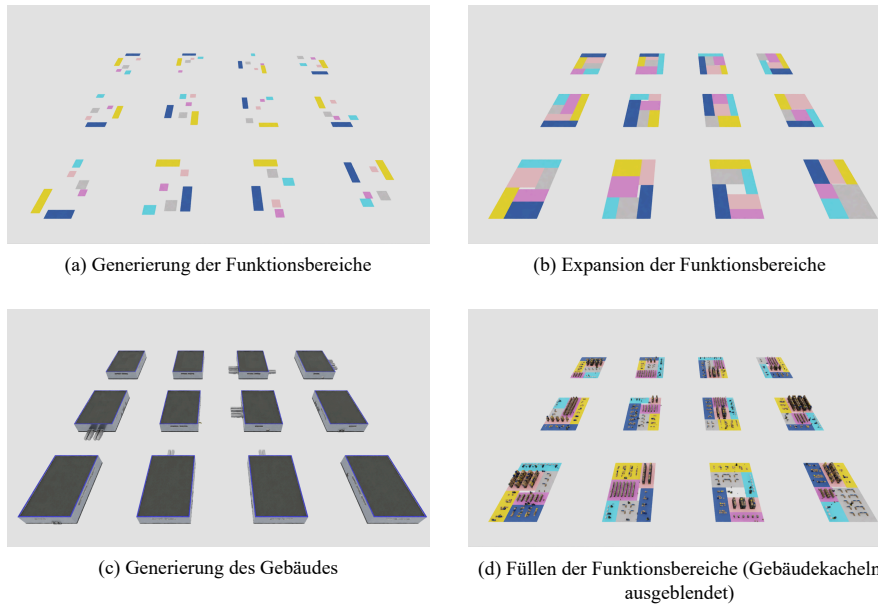
## 6.3 Darstellung der Ergebnisse

In diesem Abschnitt werden die Resultate der Umsetzung des entwickelten Generierungsalgorithmus vorgestellt. Die Darstellung erfolgt in drei Schritten: Zunächst wird die Strukturgenerierung analysiert, anschließend die Generierung der Materialflüsse betrachtet und schließlich die Eignung der erzeugten Umgebungen für das Lernen und Testen von Steuerungsstrategien von AMR bewertet.

### 6.3.1 Strukturgenerierung

Die Generierung der Struktur der Lern- und Testumgebungen ist der erste Teil des Generierungsalgorithmus. In diesem Teil wird vor Simulationsstart,

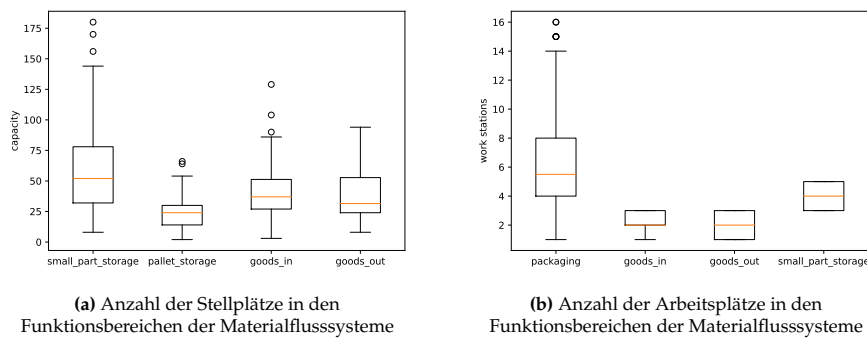
basierend auf den Generierungsanforderungen aus der Benutzeroberfläche und aus den Konfigurationsdateien, eine geforderte Menge an Materialflusssystemen in eine Szene in Isaac Sim generiert. Dazu werden die Positionen und Rotationen der Objekte in der Szene durch den in Abschnitt 5.2 entwickelten Algorithmus bestimmt und die Assets aus der Datenbank von Nvidia an den entsprechenden Stellen eingefügt.



**Abb. 6.4** Teilschritte der Strukturgenerierung in Nvidia Isaac Sim [MJV25, S. 2506]

Abbildung 6.4 zeigt die einzelnen Phasen des Generierungsalgorithmus, wie er in Isaac Sim realisiert ist. In dem hier dargestellten Fall werden zwölf Lern- und Testumgebungen in eine Szene generiert. Im ersten Schritt werden die Funktionsbereiche zufällig in den Flächen der Materialflusssysteme generiert (Abbildung 6.4(a)). Anschließend wachsen sie entsprechend den zufällig gewählten Wachstumsfaktoren, sodass die entstehenden Blocklayouts alle unterschiedlich strukturiert sind (Abbildung 6.4(b)). Dann wird das Gebäude generiert und die Gebäudekacheln mit Toren an den Wareneingängen und Warenausgängen positioniert (Abbildung 6.4(c)). Anschließend werden die Funktionsbereiche mit Assets logistischer Objekte gefüllt (Abbildung 6.4(d)). Die Schritte laufen dabei automatisch ab und sind hier zu Visualisierungszwecken einzeln dargestellt.

Um die Variabilität in den erzeugten Materialflusssystemen zu validieren, werden 100 verschiedene Modelle von Materialflusssystemen generiert. Dabei kommen insgesamt 19.489 einzelne Assets zum Einsatz. Jedes Lager



**Abb. 6.5** Boxplots der Strukturgenerierung in Isaac Sim [MJV25, S. 2507]

weist eine Grundfläche von 60 mal 40 Metern, auf und in jeder Instanz ist ein Exemplar der sechs vordefinierten Funktionsbereiche enthalten.

Die erzeugten Umgebungen werden als Dateien im Format Universal Scene Description (USD) gespeichert. Zusätzlich wird für jede Lern- und Testumgebung eine Matrix erzeugt, die die enthaltenen Funktionsbereiche, die Assets der ihnen zugeordneten logistischen Objekte und die zufällig platzierten Objekte formal beschreibt.

Zur Auswertung der generierten Umgebungen werden Metriken aus den Matrizen berechnet und in Boxplots dargestellt (siehe Abb. 6.5). Die Auswertung der Variabilität der Struktur, die der entwickelte Generierungsalgorithmus erreicht, wurde bereits in [MJV25] veröffentlicht.

Die bewerteten Metriken beziehen sich auf zwei zentrale Aspekte der Systemstruktur:

- Die *Kapazität pro Lager* ergibt sich aus der Anzahl der Regale in Kleinteile- und Palettenlagern sowie aus den möglichen Lagerplätzen in Wareneingangs- und Warenausgangsbereichen (Abbildung 6.5a).
- Die *Anzahl möglicher Arbeitsplätze pro Lager* ergibt sich aus den Arbeitsstationen im Verpackungsbereich sowie aus Kommissionier- und Buchungswagen in Kleinteilelagern, Wareneingang und Warenausgang (Abbildung 6.5b).

Die Darstellung dieser Metriken für die 100 generierten Materialflusssysteme zeigt, dass das Generierungsverfahren eine große Bandbreite unterschiedlicher Strukturen erzeugt. Lediglich die Anzahl der Arbeitsplätze im Wareneingang, Warenausgang und im Kleinteilelager sind nach oben und unten begrenzt. Für diese Metriken wurden im Generierungsalgorithmus Grenzen festgelegt, um den Realismus der generierten Materialflusssysteme beizubehalten. Im Vergleich zu den anderen Metriken sind sie daher nicht abhängig von der Größe der generierten Funktionsbereiche.

Neben der Variabilität konnte durch die auswertbaren Matrizen gezeigt werden, dass die generierten Modelle konsistent und formal beschreibbar

sind. Zudem sind die generierten logistischen Objekte durch die Nutzung der Assets in Nvidia Isaac Sim physikalisch akkurat, und die dem Generierungsalgorithmus zugrunde gelegten Regeln sorgen für semantisch kohärente Materialflusssysteme. Damit bildet die entwickelte Strukturgenerierung eine valide Grundlage für Lern- und Testumgebungen im Digitalen Kontinuum.

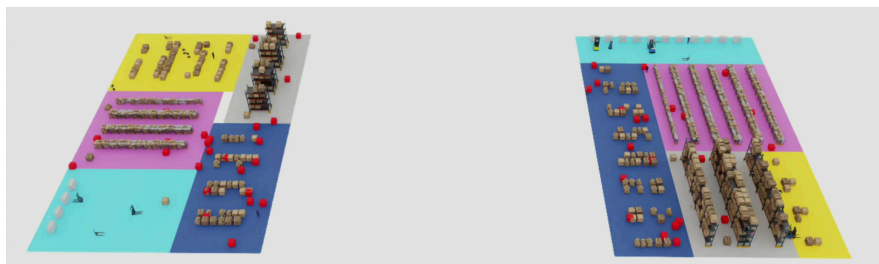
### 6.3.2 Materialflussgenerierung

Die Generierung der Materialflüsse, als dynamische Komponente des Generierungsalgorithmus, erfolgt zur Laufzeit der Simulation. Zur Validierung der Funktionsfähigkeit der Transportauftragsgenerierung wird das Transportauftragsprofil in Tabelle 6.1 in zwei Lern- und Testumgebungen simuliert, und anschließend werden die Zwischenankunftszeiten der Transportaufträge visualisiert. In der Materialflussmatrix stehen die Quell-Funktionsbereiche in den Zeilen und die Ziel-Funktionsbereiche in den Spalten.

**Tabelle 6.1** Materialflussmatrix zur Erzeugung der Transportaufträge [Ladeeinheiten/h]

	Wareneingang	Warenausgang	Palettenlager	Kleinteilelager
Wareneingang	0	0	100	100
Warenausgang	0	0	0	0
Palettenlager	0	0	0	50
Kleinteilelager	0	150	0	0

Abbildung 6.6 zeigt die anhand der Materialflussmatrix erzeugten Transportaufträge zur Simulationszeit 160 Sekunden in der Simulationsumgebung. Die roten Würfel zeigen die Quell-Positionen in den Lern- und Testumgebungen an, die in diesem Fall zur besseren Sichtbarkeit zufällig in den Flächen der Funktionsbereiche generiert wurden. Es wurden insgesamt 52 Transportaufträge erzeugt, 26 in jedem Materialflusssystem. Die Quellbereiche sind jeweils der Wareneingang (blau), das Paletten- und das Kleinteilelager (grau und fuchsia).



**Abb. 6.6** Generierte Transportaufträge in zwei Materialflusssystemen zur Simulationszeit 160 Sekunden

Abbildung 6.7 zeigt die Quellen (rot) und auch die Senken (dunkelblau) der erzeugten Transportaufträge in einem Streudiagramm. In Kombination mit Abbildung 6.6 ist zu erkennen, dass die Zielbereiche der Transportaufträge in den Funktionsbereichen Warenausgang, Palettenlager und Kleinteilelager liegen. Somit kann die Zuordnung der Transportaufträge zu Funktionsbereichen entsprechend der vorgegebenen Materialflussmatrix positiv validiert werden.

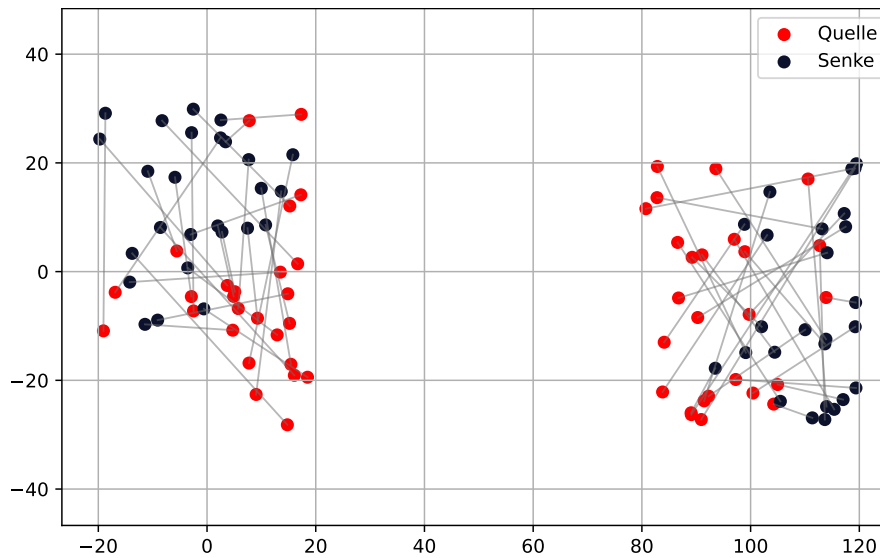


Abb. 6.7 Quellen (rot) und Senken (dunkelblau) der generierten Transportaufträge zur Simulationszeit 160 Sekunden

Die Zwischenankunftszeiten der generierten Transportaufträge aus beiden Materialflusssystemen in dem getesteten Szenario sind in Abbildung 6.8 dargestellt. In der Simulation werden die Zwischenankunftszeiten aus einer Exponentialverteilung erzeugt. Aufgrund der Kürze der Simulationszeit und der unterschiedlichen Mittelwerte der einzelnen Materialflussverbindungen zwischen den Funktionsbereichen lässt sich die Verteilung in den Diagrammen nicht direkt erkennen. Dennoch wird deutlich, dass die Zwischenankunftszeiten stochastisch generiert werden und variable, dynamische Lastszenarien simuliert werden können.

Insgesamt lässt sich feststellen, dass es möglich ist, durch den Ansatz einer zentralen Auftragsgenerierung Transportaufträge entsprechend einer Materialflussmatrix an variablen Koordinaten und zu variablen Zeitpunkten in den Simulationsumgebungen zu erzeugen. Diese stehen somit zur Verfügung, um auf ihrer Basis Steuerungsstrategien für AMR in verschiedenen Steuerungsarchitekturen zu lernen und zu testen.

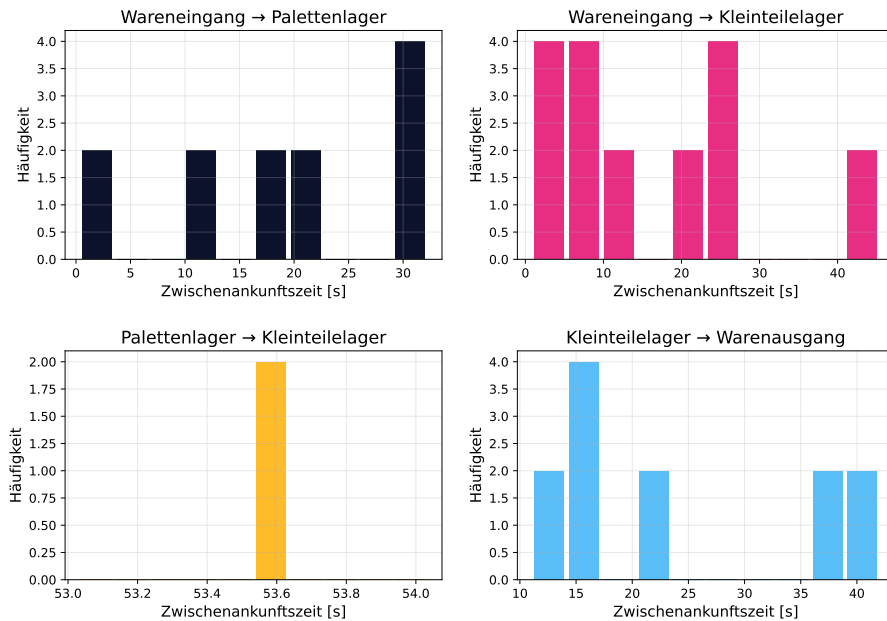


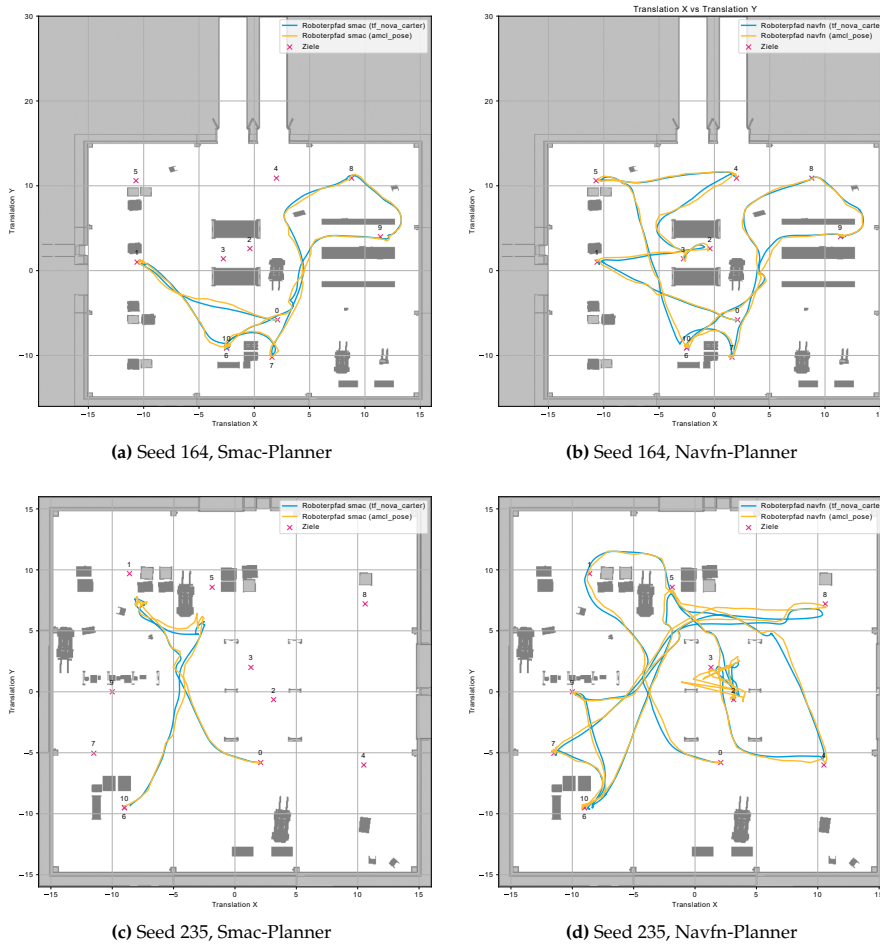
Abb. 6.8 Histogramme der Zwischenankunftszeiten der Transportaufträge

### 6.3.3 Lern- und Testfähigkeit robotischer Steuerungsstrategien

Schließlich wird die Lern- und Testfähigkeit robotischer Steuerungsalgorithmen in den erzeugten Umgebungen validiert. In den generierten Umgebungen sollen End-to-End-Steuerungsalgorithmen entwickelt und getestet werden können, die das Verhalten der Roboter basierend auf ihren Sensorwahrnehmungen bestimmen. Zur Validierung der Lern- und Testfähigkeit wird eine Navigationsaufgabe in vier generierten Materialflusssystemen durchgeführt. Hierbei werden fünf Transportaufträge mit jeweils fünf Quell- und fünf Zielkoordinaten abgearbeitet, die zuvor durch die Materialflussgenerierung bestimmt wurden.

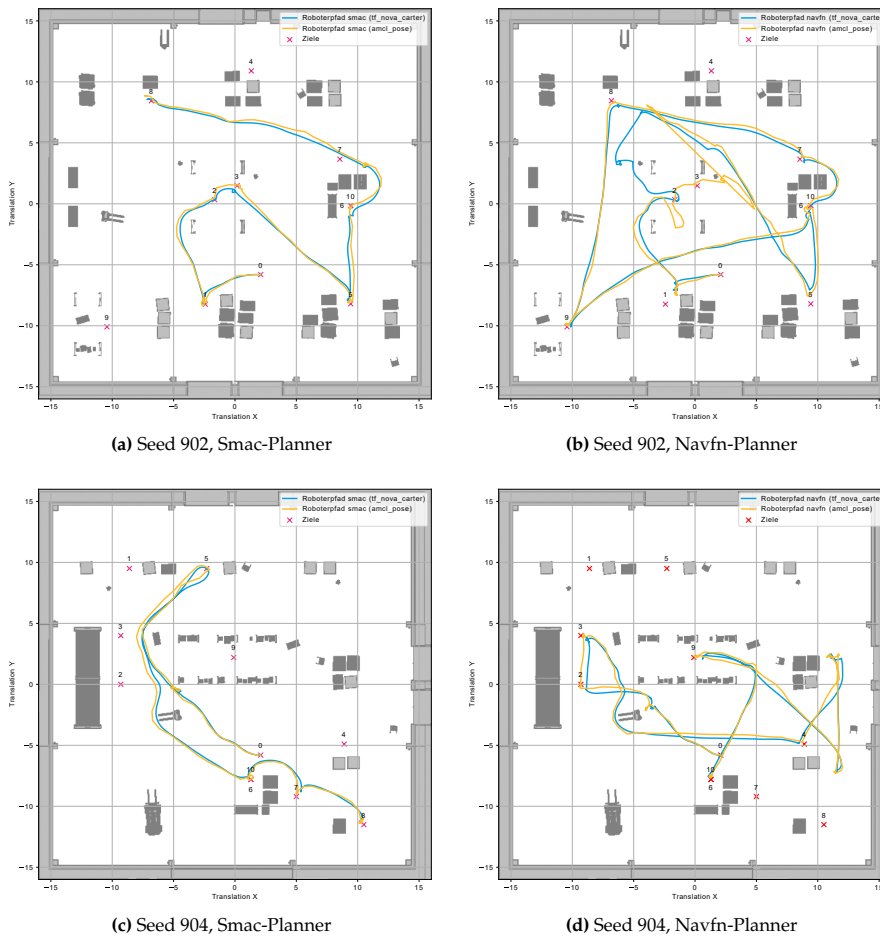
Für die Umsetzung wird das in Isaac Sim verfügbare Robotermodell Nova Carter eingesetzt. Nova Carter ist ein Referenzroboter, der auf der Plattform Nova Orin basiert und über verschiedene vorkonfigurierte Sensoren wie Stereo- und Fisheye-Kameras, 2D- und 3D-Lidar sowie inertielle Messeinheiten (engl. Inertial Measurement Unit (IMU)) verfügt [NVI25]. Diese Ausstattung erlaubt es, die Navigation mit einer realitätsnahen Roboterkonfiguration in der Simulationsumgebung zu validieren, ohne dass zusätzliche Anpassungen notwendig sind.

Zur Bewertung der Lern- und Testfähigkeit wird die Steuerung der Roboter für die Navigationsaufgabe direkt über die Einbindung des Frameworks Robot Operating System (ROS) in Isaac Sim realisiert. Die direkte Integration



**Abb. 6.9** Vergleich der gefahrenen und erkannten Wege mit dem Smac-Planner und dem Navfn-Planner, Teil 1

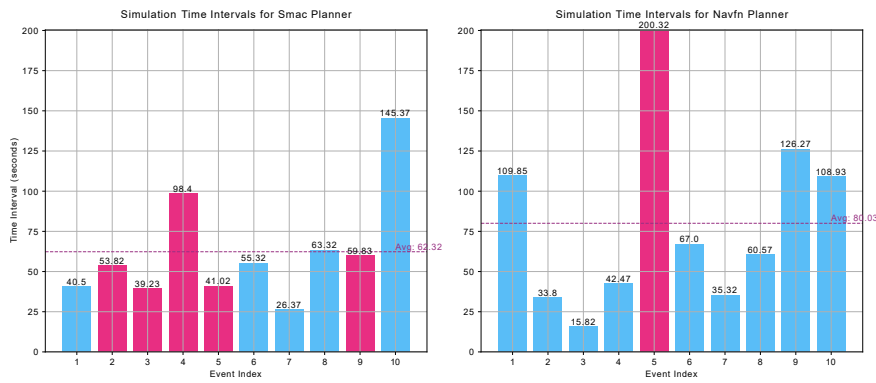
von ROS ist dabei von zentraler Bedeutung, da ROS in industriellen und wissenschaftlichen Anwendungen als De-facto-Standard für die robotische Steuerung gilt und die reale Softwarearchitektur von AMR widerspiegelt. Dadurch wird sichergestellt, dass die getesteten Steuerungsprozesse und Schnittstellen den realen Betriebsbedingungen möglichst nahekommen und die Übertragbarkeit der Ergebnisse aus der Simulation in die physische Anwendung unterstützt wird. Zudem ermöglicht die ROS-Kommunikation die Nutzung von Sensordatenströmen, wie sie auch in der Realität erzeugt werden, sowie klassischer Planungs-, Lokalisierungs- und Steuerungsframeworks ohne Zwischenschnittstellen, was die Aussagekraft der Validierung erhöht und den Sim-to-Real-Gap reduziert.



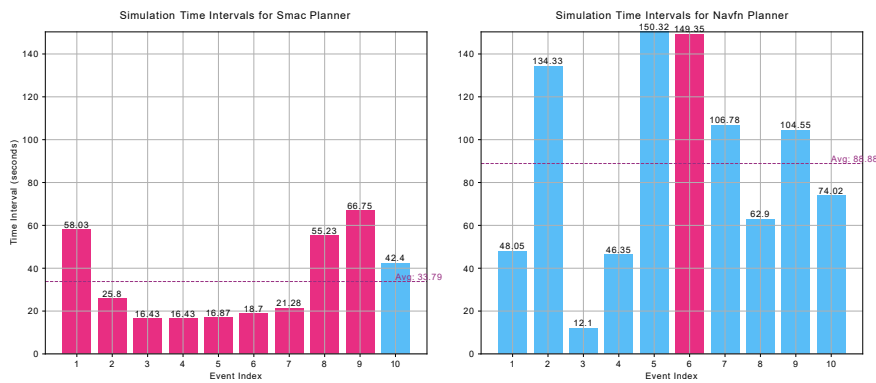
**Abb. 6.10** Vergleich der gefahrenen und erkannten Wege mit dem Smac-Planner und dem Navfn-Planner, Teil 2

Für die Navigationsaufgabe werden zwei unterschiedliche Wegeplanungsverfahren aus dem ROS-Navigation-Stack eingesetzt: der Smac-Planner [LLC25b] und der Navfn-Planner [LLC25a]. Es handelt sich dabei um vordefinierte, nicht lernbasierte Verfahren, die hier in ihren Standardkonfigurationen genutzt und miteinander verglichen werden. Beide Algorithmen arbeiten auf Basis der simulierten Lidar-Daten. Die Leistungsfähigkeit der Navigationsstrategien wird anhand zweier Kennzahlen bewertet: der Erfolgsrate (Anteil erfolgreich erreichter Ziele) und der durchschnittlichen Zeit bis zum Erreichen einer Zielposition.

In Abbildung 6.9 und Abbildung 6.10 sind die tatsächlich gefahrenen Trajektorien der Roboter in den vier unterschiedlichen, durch das Generierungsverfahren erzeugten Materialflusssystemen dargestellt. Die Kreuze



(a) Seed 164

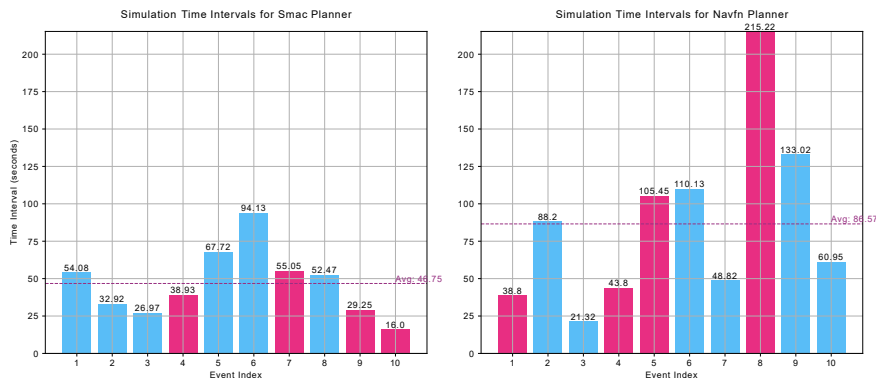


(b) Seed 235

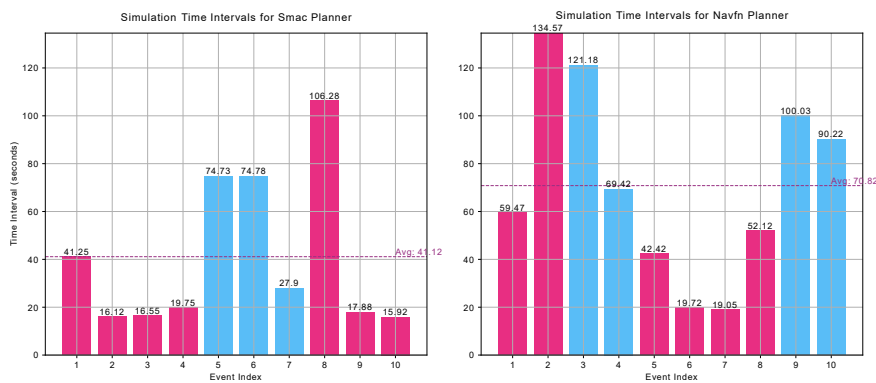
**Abb. 6.11** Vergleich der Simulationszeiten mit dem Smac-Planner und dem Navfn-Planner, Teil 1

markieren die Quell- und Zielkoordinaten der Transportaufträge, die es abzu-  
arbeiten gilt. Die in Blau eingezeichneten Pfade zeigen die durch die Roboter  
tatsächlich zurückgelegten Wege in der Simulationsumgebung, während die  
in Gelb markierten Wege die Positionen angeben, an denen der Roboter beim  
Einsatz der getesteten Verfahren lokalisiert wurde. In den Abbildungen sind  
zudem die verwendeten *Seeds* angegeben, also die Initialisierungswerte des  
Zufallsgenerators, die eine Wiederholbarkeit der Experimente sicherstellen.

In Abbildung 6.11 und Abbildung 6.12 sind ergänzend die benötigten  
Simulationszeiten für die Bearbeitung der Transportaufträge dargestellt. Je-  
der Balken repräsentiert die Zeit, die ein Roboter benötigt hat, um eine der  
geforderten Zielkoordinaten zu erreichen. Erfolgreich abgeschlossene Auf-  
träge sind durch blaue Balken gekennzeichnet, während Aufträge, bei denen  
das Ziel nicht erreicht werden konnte, in Fuchsia dargestellt sind. Auf diese  
Weise lässt sich sowohl die Erfolgsrate als auch die Zeitspanne zwischen den  
Transportaufträgen für die beiden Wegeplanungsverfahren vergleichen.



(a) Seed 902



(b) Seed 904

**Abb. 6.12** Vergleich der Simulationszeiten mit dem Smac-Planner und dem Navfn-Planner, Teil 2

Die Ergebnisse zeigen, dass es grundsätzlich möglich ist, End-to-End-Wegeplanungsalgorithmen in den generierten Umgebungen umzusetzen und ihre Leistungsfähigkeit zu vergleichen. Gleichzeitig wird deutlich, dass beide Wegeplanungsverfahren nicht optimal arbeiten: Die Lokalisierung ist häufig ungenau, und nicht alle Ziele können zuverlässig erreicht werden, da teilweise keine gültigen Pfade gefunden werden. Der Navfn-Planner schneidet dabei etwas besser ab als der Smac-Planner. Beim Einsatz im Digitalen Kontinuum wäre die Schlussfolgerung aus den durchgeführten Tests, die Wegeplanungsalgorithmen im nächsten Schritt anzupassen und noch einmal zu testen, bis die Ergebnisse passend sind.

Im Kontext dieser Arbeit stellen die Ergebnisse dennoch einen Erfolg dar, da sie zeigen, dass robotisches Verhalten über ROS in den generierten Umgebungen steuerbar ist und unterschiedliche Algorithmen unter realitätsnahen Bedingungen vergleichbar sind. Dadurch sind alle Elemente für eine generelle Lern- und Testfähigkeit in den Umgebungen gegeben (Wahrnehmung

der Umgebungszustände, Berechnung der nächsten Aktion, Ausführung der Aktion). Allerdings zeigen die Experimente auch die vergleichsweise hohen Rechenzeiten bei der Bearbeitung der Transportaufträge auf.

## 6.4 Reflexion der Ergebnisse

Aufbauend auf den im vorherigen Abschnitt dargestellten Ergebnissen des Generierungsvorgehens werden im Folgenden die Resultate im Hinblick auf die in Abschnitt 4.1 formulierten Anforderungen an den Algorithmus reflektiert.

### 6.4.1 Digitales Kontinuum

Die Anforderungen aus dem Bereich des Digitalen Kontinuums konnten in wesentlichen Punkten erfüllt werden.

Hinsichtlich des Automatisierungsgrads zeigt sich, dass das Verfahren die gestellten Anforderungen erfüllt. Nach wenigen Eingaben in der Benutzeroberfläche erfolgt die Generierung vollautomatisch, sodass der Eingabeaufwand minimal ist. Diese Angaben können im Digitalen Kontinuum durch einen KI-Agenten entsprechend dem Kontext der Lern- oder Testaufgabe bestimmt und dem Generierungsalgorithmus zur Verfügung gestellt werden.

Die Parallelisierung der Lern- und Testumgebungen konnte in der Strukturgenerierung und der Generierung der Materialflüsse gezeigt werden. Für die Robotersteuerung unter ROS-Anbindung ist die Parallelisierung aufgrund der Rechenlast der detaillierten Umgebungen in Kombination mit den Sensorwahrnehmungen zum aktuellen Zeitpunkt schwieriger. Durch die Weiterentwicklungen im Bereich der Rechenhardware und Algorithmen kann dies zukünftig möglich sein. Grundsätzlich ist der Generierungsalgorithmus für die Parallelisierung ausgelegt.

Die geforderten Datenstandards im Digitalen Kontinuum wurden durch ein in dieser Arbeit entwickeltes semantisches Modell umgesetzt, das speziell auf das vorgestellte Generierungsverfahren zugeschnitten ist. Für eine breitere Anwendbarkeit im Digitalen Kontinuum ist es zukünftig sinnvoll, dieses Modell an etablierte Standards für Digitale Zwillinge, wie die Verwaltungsschale [Gor+24], anzupassen. Dadurch ließe sich eine höhere Interoperabilität zwischen verschiedenen Technologien und Akteuren erreichen.

### 6.4.2 Strukturgenerierung

Die Anforderungen an die Strukturgenerierung umfassen sowohl die Randomisierung der erzeugten Modelle als auch deren Realitätsnähe.

Der Generierungsalgorithmus integriert alle relevanten Funktionsbereiche eines Materialflusssystem und erfüllt damit diese Anforderung an die Strukturgenerierung.

Zudem wird die Struktur der Materialflusssysteme durch realistische, optisch und physikalisch detaillierte Assets generiert, wobei die zugrundeliegende Logik des Aufbaus von Materialflusssystemen auf bewährten Regeln aus Praxis und Lehrbüchern basiert. Damit wird eine hohe Übereinstimmung mit den Grundprinzipien realer Materialflusssysteme erzielt. Um die Validität dieses Aspekts zukünftig weiter zu erhöhen, können zukünftige Arbeiten reale Layouts als Vergleichs- und Referenzbasis einbeziehen oder direkt als Eingabe für die Generierung verfügbar machen.

Weiterhin konnte eine breite Variabilität der Strukturgenerierung aufgezeigt werden. Die Randomisierung der Strukturen kann zukünftig weiter gesteigert werden, indem grundsätzlich die Abbildung weiterer Funktionsbereiche und Technologien eingebunden wird. So können beispielsweise neben manuellen technischen Systemen auch automatische Systeme in der Generierungslogik berücksichtigt werden. Während in manuellen Systemen statische Objekte abgebildet werden, bringen automatische Systeme durch eigene Steuerungslogiken eine zusätzliche Komplexitätsebene mit sich, die für das Training und Testen von AMR-Strategien von Relevanz ist.

Eine zusätzliche Steigerung der Vielfalt der generierten Strukturen kann zukünftig über die Abbildung verschiedener Formen der Gebäude und Funktionsbereiche erreicht werden, beispielsweise L- und U-Formen.

### 6.4.3 Materialflussgenerierung

Die Anforderungen an die Generierung der Materialflüsse werden durch den entwickelten Generierungsalgorithmus im Grundsatz erfüllt: Transportaufträge können dynamisch zur Simulationslaufzeit generiert und ohne Vorgabe der Steuerungsarchitektur der AMR bereitgestellt werden.

In dem aktuellen Verfahren werden die Transportaufträge für alle parallelen Lern- und Testumgebungen zum gleichen Zeitpunkt erzeugt. Hier kann zukünftig die Generierung verschiedener Lastszenarien in den parallelen Lern- und Testumgebungen für zusätzliche Variation sorgen.

Die Generierung der Transportaufträge basiert aktuell auf Materialflussmatrizen, die als Grundlage für die stochastische Verteilung der Transportaufträge dienen. Zukünftig kann die Realitätsnähe weiter gesteigert werden, indem reale Ankunftszeitverteilungen aus betrieblichen Systemen zur Erzeugung der Materialflüsse herangezogen werden. Darüber hinaus wäre die Einbeziehung zusätzlicher Szenarien, wie Störungen oder variierende Lastprofile im Tagesverlauf, sinnvoll, um die Bandbreite möglicher Einsatzsituationen weiter zu variieren und robustere Steuerungsstrategien entwickeln zu können.

#### **6.4.4 Lern- und Testfähigkeit robotischer Steuerungsstrategien**

Die Testfähigkeit robotischer Steuerungsstrategien konnte in der Validierung nachgewiesen werden. In den generierten Umgebungen lassen sich End-to-End-Steuerungsalgorithmen abbilden, die robotisches Verhalten basierend auf Sensorinformationen in Aktionen umsetzen. Es werden somit die relevanten Elemente für die Lern- und Testfähigkeit von robotischen Steuerungsstrategien bereitgestellt.

Die eingesetzte Robotiksimulationsumgebung ermöglicht eine realitätsnahe Abbildung der Roboterhardware, einschließlich Sensoren und Aktoren. Die Steuerung über ROS ist funktionsfähig und erlaubt die Integration verschiedener Steuerungsarchitekturen des Materialflusssystems.

Gleichzeitig erweist sich die Nutzung physikalisch exakter Umgebungen für RL-Frameworks und das Training in großen Szenen als rechenintensiv. Zukünftige Arbeiten können daher an der Entwicklung von Abstraktionsebenen ansetzen, die die Komplexität der Simulationsumgebungen gezielt an die zu erlernenden oder zu testenden Aufgaben anpassen.

#### **6.4.5 Schlussfolgerung der Validierung**

Zusammenfassend lässt sich feststellen, dass das entwickelte Verfahren die in Abschnitt 4.1 identifizierten Anforderungen an die Generierung variantenreicher, physikalisch realistischer Lern- und Testumgebungen für die Anwendung im Digitalen Kontinuum im Wesentlichen erfüllt. Die Strukturgenerierung basiert auf etablierten logistischen Gestaltungsprinzipien und liefert konsistente Ergebnisse auf einem passenden Detaillierungsgrad, während die Materialflussgenerierung dynamische und randomisierte Prozesse zur Laufzeit der Simulation ermöglicht. Auch die Lern- und Testfähigkeit robotischer Steuerungsstrategien konnte an einem Testszenario in der Simulation demonstriert werden.

Es bestehen dennoch Weiterentwicklungspotenziale. Dazu zählen die Integration standardisierter Datenmodelle, die Steigerung des Variantenreichtums der Strukturgenerierung, die stärkere Orientierung an realen Layouts und die Erweiterung der Materialflussgenerierung um Stör- und Lastszenarien. Außerdem sind Optimierungen im Bereich der Rechenlast erforderlich, beispielsweise durch die Einführung von Abstraktionsebenen der Umgebungen oder der eingesetzten Robotermodelle entsprechend der Lern- oder Testaufgabe.

Insgesamt bietet das Verfahren eine passende Grundlage für die Generierung von Lern- und Testumgebungen für AMR-Steuerungsstrategien im Digitalen Kontinuum. Es leistet damit einen wesentlichen Beitrag zur Realisierung autonomer, selbstoptimierender und effizienter Entwicklungs- und Betriebsprozesse logistischer Systeme.



## Kapitel 7

# Fazit und Ausblick

**Zusammenfassung** Dieses Kapitel fasst die zentralen Beiträge der Arbeit zusammen und zeigt Perspektiven für weitere Entwicklungen zur Realisierung des Digitalen Kontinuums auf. Es resümiert das Leitbild des Digitalen Kontinuums für die durchgängige, KI-basierte Entwicklung und den Betrieb hochautonomer logistischer Systeme und zieht ein Fazit zum entwickelten Generierungsverfahren für simulationsfähige Lern- und Testumgebungen für AMR, das eine wesentliche Unterbrechung in diesen Prozessen schließt. Im Ausblick werden anschließend potenzielle Forschungsansätze zur Verbindung der weiterhin existierenden Unterbrechungen im Digitalen Kontinuum aufgezeigt.

## 7.1 Zusammenfassung

Die vorliegende Arbeit verfolgt das Ziel, einen Beitrag zur Realisierung des Potenzials von KI in hochautonomen intralogistischen Systemen zu leisten. Ausgangspunkt ist die Beobachtung, dass Autonomie als zentraler Hebel zur Effizienzsteigerung in logistischen Systemen gilt, jedoch bislang nur eingeschränkt erreicht wurde. Obwohl KI vielfach als Schlüsseltechnologie zur Steigerung des Autonomiegrades identifiziert wird, existieren bisher keine Entwicklungs- und Betriebskonzepte, die die vollständige Integration dieser Technologie in den Mittelpunkt stellen.

Mit dem Leitbild des Digitalen Kontinuums stellt diese Arbeit erstmals systematisch und theoretisch fundiert einen konzeptionellen Rahmen auf, der eine durchgängige, tiefgreifende Integration von KI-Technologien in die Entwicklungs- und Betriebsprozesse autonomer Systeme in der Logistik anstrebt. Das Digitale Kontinuum basiert auf dem durch KI-Agenten gesteuerten Entwicklungszyklus zwischen physischer und digitaler Realität und vereint die Dimensionen vertikaler, horizontaler und kausaler Kontinuität sowie die befähigenden Schlüsseltechnologien zu einem ganzheitlichen Leitbild. Basierend auf diesem Leitbild werden Unterbrechungen identifiziert, die einer Realisierung des Digitalen Kontinuums heute noch im Wege stehen.

Ein zweiter wesentlicher Beitrag dieser Arbeit ist die Entwicklung eines Generierungsverfahrens für simulationsfähige Lern- und Testumgebungen für AMR in Materialflusssystemen für die Anwendung im Digitalen Kontinuum. Dieses Verfahren zählt auf die Verbindung einer zentralen Unterbrechung des Digitalen Kontinuums ein: die bislang manuelle und zeitaufwändige Erstellung virtueller Lern- und Testumgebungen. Das entwickelte Generierungsverfahren ermöglicht die automatisierte Erzeugung semantisch kohärenter und physikalisch detaillierter Strukturen in den Umgebungen sowie die dynamische Generierung von Materialflüssen. Es berücksichtigt realitätsnahe Steuerungsarchitekturen und technische Komponenten von AMR und eignet sich für End-to-End-basierte robotische Steuerungsansätze.

Mit der prototypischen Realisierung eines wesentlichen Teils des Entwicklungszyklus des Digitalen Kontinuums wurde in der Arbeit gezeigt, wie ein bislang manuell geprägter Prozessschritt automatisiert und so in einen kontinuierlichen und selbstoptimierenden Prozess integriert werden kann.

In Bezug auf die in Kapitel 1 definierten angestrebten Beiträge der Arbeit lässt sich somit folgern, dass

- die Einführung und Ausarbeitung des Digitalen Kontinuums als Leitbild für die KI-basierte Entwicklung und den Betrieb autonomer logistischer Systeme erfolgt ist,
- die Entwicklung eines Verfahrens zur automatisierten Generierung simulationsfähiger Lern- und Testumgebungen, das semantische Kohärenz,

physikalische Genauigkeit, Variantenvielfalt und dynamische Materialflussgenerierung vereint, erreicht wurde und zusätzlich

- seine Umsetzung und Evaluation in der Simulationsumgebung Isaac Sim von Nvidia und die Demonstration seiner Eignung zur Erzeugung variantenreicher Umgebungen für das Training und Testen robotischer Steuerungsstrategien erfolgt sind.

## 7.2 Ausblick

Die Ergebnisse dieser Arbeit schaffen die Grundlage für weiterführende Forschung sowohl im Bereich der Weiterentwicklung des Generierungsalgorithmus für Lern- und Testumgebungen für AMR in Materialflusssystemen als auch im Bereich der Realisierung des Digitalen Kontinuums.

Zukünftige Forschung kann zunächst bei der Weiterentwicklung des Generierungsverfahrens ansetzen. Ein wesentliches Potenzial liegt in der Integration komplexerer und realitätsnäherer Randbedingungen, beispielsweise von Strukturen basierend auf Layouts aus der Praxis, von Störungen oder unvorhergesehenen Ereignissen und von variierenden Lastszenarien. Durch die Abbildung solcher Situationen kann die Robustheit und Generalisierungsfähigkeit der entwickelten Steuerungsstrategien weiter erhöht werden. Ebenso können dynamische Verhaltensweisen von Objekten innerhalb der Simulation eingebunden werden, sodass Aspekte wie Kreuzungsverkehr oder menschliches Verhalten abgebildet werden können. Zudem ist die Entwicklung von Abstraktionsschichten der generierten Modelle eine vielversprechende Forschungsrichtung, um aufgabenabhängig zwischen Realitätsnähe und Rechenaufwand wählen zu können und das Spektrum zwischen Materialflusssimulation und Robotiksimulation fließend zu gestalten.

Ein weiterer Ansatz zur Umsetzung des Digitalen Kontinuums auf Basis der generierten Lern- und Testumgebungen ist die Prüfung, wie gut in ihnen entwickelte Steuerungsstrategien auf die Realität übertragbar sind. Beispielsweise kann durch die Kombination automatisch generierter Simulationsumgebungen mit physischen Testfeldern die Übertragbarkeit erlernter Strategien auf die Realität bewertet werden. Ebenso können die Erkenntnisse aus dem physischen Testfeld in die Generierung der Lernumgebungen einfließen. So kann eine erste Umsetzung des Digitalen Kontinuums in einer Laborumgebung entstehen.

Gleichzeitig wird deutlich, dass Teile des idealisierten Entwicklungszyklus des Digitalen Kontinuums weiterhin unterbrochen sind. Dazu gehört beispielsweise die automatisierte Formulierung neuer Lösungsansätze. An dieser Stelle können perspektivisch Ansätze untersucht werden, die die Generierung neuer Entscheidungsmechanismen nicht mehr primär durch menschliche Entwicklungsteams, sondern durch autonome KI-Agenten verfolgen. Ein erster Ansatz kann darin bestehen, das in dieser Arbeit entwi-

ckelte Verfahren durch KI-Agenten entsprechend des erkannten Kontextes parametrisieren und ausführen zu lassen. Mit der automatisierten Problembeschreibung, Erzeugung von Lösungsansätzen und deren Evaluation im Digitalen Kontinuum entsteht ein evolutionärer Entwicklungsprozess.

Dieser Wandel markiert eine grundlegende Verschiebung gegenüber traditionellen Entwicklungsprozessen. Während konventionelle Verfahren neue Steuerungslogiken in langen, iterativen Zyklen manuell konzipieren und testen, erlaubt das Digitale Kontinuum eine beschleunigte Entwicklungsdynamik. In diesem Leitbild entstehen Strategien fortlaufend aus datenbasiertem Feedback, eigenständiger Formulierung von Lösungsansätzen und experimenteller Überprüfung in simulierten und realen Umgebungen. Die vorliegende Arbeit liefert die methodische Grundlage für diesen Paradigmenwechsel und demonstriert ein erstes funktionales Element eines solchen Systems.

Insgesamt verdeutlicht diese Arbeit, dass das Digitale Kontinuum nicht nur ein theoretisches Leitbild darstellt, sondern den strukturellen Ausgangspunkt einer zukünftigen, sich selbst optimierenden logistischen Architektur bildet. Die hier entwickelte Methodik schließt eine zentrale Lücke auf dem Weg dorthin und macht deutlich, dass selbstlernende, agentische Steuerung in logistischen Systemen nicht mehr ausschließlich Vision ist, sondern zunehmend in Reichweite rückt. Darauf aufbauend eröffnet sich ein Forschungsfeld, das die Gestaltung intelligenter Wertschöpfungssysteme neu definiert und den Weg hin zu autonomer, adaptiver und sich kontinuierlich verbessernder Logistik ebnet.

## Literaturverzeichnis

- [Abd+21] Mohamed Abdelazeem u. a. „Multi-sensor point cloud data fusion for precise 3D mapping“. In: *The Egyptian Journal of Remote Sensing and Space Science* 24.3 (1. Dez. 2021), S. 835–844. ISSN: 1110-9823. DOI: 10.1016/j.ejrs.2021.06.002.
- [Aca11] Acatech - Deutsche Akademie der Technikwissenschaften, 2011, Hrsg. *Cyber-Physical Systems: Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion*. Acatech Position 11. Berlin, Heidelberg: Springer Berlin Heidelberg Imprint Springer, 2011. ISBN: 978-3-642-27567-8.
- [AG21] Fadi AlMahamid und Katarina Grolinger. „Reinforcement Learning Algorithms: An Overview and Classification“. In: *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2021, S. 1–7. DOI: 10.1109/CCECE53047.2021.9569056.
- [AKD25] Deepak Bhaskar Acharya, Karthigeyan Kuppan und B. Divya. „Agentic AI: Autonomous Intelligence for Complex Goals - A Comprehensive Survey“. In: *IEEE Access* 13 (2025), S. 18912–18936. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3532853.
- [AL12] Albert Albers und Quentin Lohmeyer. „Advanced Systems Engineering - Towards a Model-Based and Human-Centered Methodology“. In: *Proceedings of TMCE 2012*. 9th International Symposium on Tools and Methods of Competitive Engineering TMCE 2012. Karlsruhe, Germany, 2012, S. 407–416.
- [Alt12] Oliver Alt. *Modellbasierte Systementwicklung mit SysML*. München: Hanser, 2012. 204 S. ISBN: 978-3-446-43066-2.
- [Alu15] Rajeev Alur. *Principles of cyber-physical systems*. Cambridge (Mass.): The MIT press, 2015. ISBN: 978-0-262-02911-7.
- [Ame23] Geekplus America. *Tote-to-Person Roboshuttle Picking*. 2023. URL: <https://www.geekplus.com/product/roboshuttle> (besucht am 11. 05. 2024).
- [Ant+19] Gabriel Antoniu u. a. *A blueprint for the new Strategic Research Agenda for High Performance Computing*. Apr. 2019.
- [Arn06] Dieter Arnold, Hrsg. *Intralogistik: Potentiale, Perspektiven, Prognosen*. VDI-Buch. Berlin Heidelberg: Springer, 2006. ISBN: 978-3-540-29657-7.
- [Arn08] Dieter Arnold, Hrsg. *Handbuch Logistik*. 3., neu bearb. Aufl. VDI-[Buch]. Berlin Heidelberg: Springer, 2008. 1137 S. ISBN: 978-3-540-72929-7 978-3-540-72928-0.
- [Aru+17] Kai Arulkumaran u. a. „Deep Reinforcement Learning: A Brief Survey“. In: *IEEE Signal Processing Magazine* 34.6 (2017), S. 26–38. ISSN: 1558-0792. DOI: 10.1109/MSP.2017.2743240.

- [ASR A1.5] Arbeitsschutzausschüsse beim BMAS - Ausschuss für Arbeitsstätten, Hrsg. BGHW ASR A1.5:2022-03, *Technische Regeln für Arbeitsstätten - Fußböden*. 2022.
- [ASR A1.8] Arbeitsschutzausschüsse beim BMAS - Ausschuss für Arbeitsstätten, Hrsg. BGHW ASR A1.8:2022-03, *Technische Regeln für Arbeitsstätten - Verkehrswege*. 2022.
- [Bae+19] Hyansu Bae u. a. „Multi-Robot Path Planning Method Using Reinforcement Learning“. In: *Applied Sciences* 9.15 (29. Juli 2019), S. 3057. ISSN: 2076-3417. DOI: 10.3390/app9153057.
- [BEP24] German Barquero, Sergio Escalera und Cristina Palmero. *Seamless Human Motion Composition with Blended Positional Encodings*. 23. Feb. 2024. DOI: 10.48550/arXiv.2402.15509.
- [BFR14] Jenay M. Beer, Arthur D. Fisk und Wendy A. Rogers. „Toward a framework for levels of robot autonomy in human-robot interaction“. In: *Journal of human-robot interaction* 3.2 (10. Juli 2014), S. 74–99. DOI: 10.5898/JHRI.3.2.Beer.
- [Bio23] BionicHIVE. *BionicHIVE introduces SquID*. 2023. URL: <https://bionichive.com/solution/> (besucht am 11. 05. 2024).
- [BK23] Haci Bayhan und Pascal Kaiser. „Cyberphysische Systeme und Industrie 4.0“. In: *IT und autonome Systeme in der Logistik*. Hrsg. von Michael Ten Hompel. Fachwissen Logistik. Berlin [Heidelberg]: Springer Vieweg, 2023, S. 29–38. ISBN: 978-3-662-66938-9.
- [Bla14] Benjamin S. Blanchard, Hrsg. *Systems engineering and analysis*. 5th ed. Essex: Pearson, 2014. ISBN: 978-1-292-02597-1.
- [BM10] C. Browne und F. Maire. „Evolutionary Game Design“. In: *IEEE Transactions on Computational Intelligence and AI in Games* 2.1 (März 2010), S. 1–16. ISSN: 1943-068X, 1943-0698. DOI: 10.1109/TCLIAIG.2010.2041928.
- [BS10] Sören Bergmann und Steffen Strassburger. „Challenges for the Automatic Generation of Simulation Models for Production Systems“. In: *Proceedings of the 2010 Summer Computer Simulation Conference, Ottawa, Canada* (2010), S. 545–549.
- [Bun18] Bundesregierung. *Strategie Künstliche Intelligenz der Bundesregierung*. 2018. URL: <https://www.bundesregierung.de/resource/blob/997532/1550276/3f7d3c41c6e05695741273e78b8039f2/2018-11-15-ki-strategie-data.pdf> (besucht am 12. 04. 2024).
- [BXA24] Elia Bonetto, Chenghao Xu und Aamir Ahmad. *GRADE: Generating Realistic And Dynamic Environments for Robotics Research with Isaac Sim*. 22. Aug. 2024. DOI: 10.48550/arXiv.2303.04466.
- [Cas+20] Victor Casamayor-Pujol u. a. „Autonomous stock counting based on a stigmergic algorithm for multi-robot systems“. In: *Computers in Industry* 122 (1. Nov. 2020), S. 103259. ISSN: 0166-3615. DOI: 10.1016/j.compind.2020.103259.
- [Cha+00] Yun-Chih Chang u. a. „B\*-Trees: a new representation for non-slicing floorplans“. In: *Proceedings of the 37th conference on Design automation - DAC '00. the 37th conference*. Los Angeles, California, United States: ACM Press, 2000, S. 458–463. ISBN: 978-1-58113-187-1. DOI: 10.1145/337292.337541.

- [Cha+20] Konstantinos Chatzilygeroudis u. a. „A Survey on Policy Search Algorithms for Learning Robot Controllers in a Handful of Trials“. In: *IEEE Transactions on Robotics* 36.2 (Apr. 2020), S. 328–347. ISSN: 1941-0468. DOI: 10.1109/TRO.2019.2958211.
- [Che+23] Zoey Chen u. a. *GenAug: Retargeting behaviors to unseen situations via Generative Augmentation*. 22. Feb. 2023. DOI: 10.48550/arXiv.2302.06671.
- [Cho+21] HeeSun Choi u. a. „On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward“. In: *Proceedings of the National Academy of Sciences* 118.1 (5. Jan. 2021). ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1907856118.
- [Col+14] David Coleman u. a. *Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study*. 15. Apr. 2014. DOI: 10.48550/arXiv.1404.3785.
- [CR19] Jiasi Chen und Xukan Ran. „Deep Learning With Edge Computing: A Review“. In: *Proceedings of the IEEE* 107.8 (Aug. 2019), S. 1655–1674. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2019.2921977.
- [Da +17] Augusto Da Cunha Reis u. a. „Warehouse design: a systematic literature review“. In: *Brazilian Journal of Operations & Production Management* 14.4 (8. Dez. 2017), S. 542–555. ISSN: 2237-8960. DOI: 10.14488/BJOPM.2017.v14.n4.a10.
- [DIN 18225] DIN 18225:1988-06, *Industriebau: Verkehrswege in Industriebauten*.
- [DIN 30781-1] DIN 30781-1:1989-05, *Transportkette - Grundbegriffe*.
- [DIN 55405] DIN 55405:2014-12, *Verpackung - Terminologie - Begriffe*. DOI: 10.31030/2243088.
- [Dud25a] Duden. *Autonomie*. In: *Duden*. 2025. URL: <https://www.duden.de/rechtschreibung/Autonomie> (besucht am 03. 10. 2025).
- [Dud25b] Duden. *System*. In: *Duden*. 2025. URL: <https://www.duden.de/rechtschreibung/System> (besucht am 03. 10. 2025).
- [Dur+11] Phillip J. Durst u. a. „The Need for High-Fidelity Robotics Sensor Models“. In: *Journal of Robotics* 2011 (17. Nov. 2011). ISSN: 1687-9600. DOI: 10.1155/2011/679875.
- [DWF18] Roman Dumitrescu, Thorsten Westermann und Tommy Falkowski. „Autonome Systeme in der Produktion“. In: *Industrie 4.0 Management* 2018 (3. Dez. 2018), S. 17–20. DOI: 10.30844/I40M\_18-6\_17-20.
- [Dze08] Verena Dzeik. „Entwicklung eines Prozesskostenmodells zur Kalkulation von Verpackungskosten manueller Verpackungsarbeitsplätze“. Diss. Dortmund: Technische Universität Dortmund, 2008.
- [EI21] Ahmed Elfakharany und Zool Hilmi Ismail. „End-to-End Deep Reinforcement Learning for Decentralized Task Allocation and Navigation for a Multi-Robot System“. In: *Applied Sciences* 11.7 (24. März 2021), S. 2895. ISSN: 2076-3417. DOI: 10.3390/app11072895.
- [Eße+22] Julian Eßer u. a. „Guided Reinforcement Learning: A Review and Evaluation for Efficient and Effective Real-World Robotics“. In: *IEEE Robotics & Automation Magazine* (2022). Conference Name: IEEE Robotics & Automation Magazine, S. 2–22. ISSN: 1558-223X. DOI: 10.1109/MRA.2022.3207664.

- [EW11] John J Enright und Peter R Wurman. „Optimization and Coordinated Autonomy in Mobile Fulfillment Systems“. In: *Automated Action Planning for Autonomous Mobile Robots: Papers from the 2011 AAAI Workshop*. 2011.
- [Fab03] Remondino Fabio. „From Point Cloud to Surface: The Modeling and Visualization Problem“. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 34.5 (2003).
- [FE17] Jonas Freiknecht und Wolfgang Effelsberg. „A Survey on the Procedural Generation of Virtual Worlds“. In: *Multimodal Technologies and Interaction* 1.4 (Dez. 2017), S. 27. DOI: 10.3390/mti1040027.
- [FE20] Jonas Freiknecht und Wolfgang Effelsberg. „Procedural Generation of Multistory Buildings With Interior“. In: *IEEE Transactions on Games* 12.3 (Sep. 2020), S. 323–336. ISSN: 2475-1510. DOI: 10.1109/TG.2019.2957733.
- [FF20] Martin Fiedler und Benjamin Federmann. „Einsatzmöglichkeiten von Drohnen zur Inventur in der Logistik 4.0“. In: *Handbuch Industrie 4.0*. Hrsg. von Michael Ten Hompel, Thomas Bauernhansl und Birgit Vogel-Heuser. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, S. 477–491. ISBN: 978-3-662-58529-0 978-3-662-58530-6. DOI: 10.1007/978-3-662-58530-6\_109.
- [FHM05] Michael Franklin, Alon Halevy und David Maier. „From databases to dataspace: a new abstraction for information management“. In: *SIGMOD Rec.* 34.4 (2005), S. 27–33. ISSN: 0163-5808. DOI: 10.1145/1107499.1107502.
- [Fle19] Bernhard Fleischmann. „Begriffliche Grundlagen der Logistik“. In: *Handbuch Logistik*. Hrsg. von Kai Furmans u. a. Berlin, Heidelberg: Springer, 2019, S. 1–13. ISBN: 978-3-642-54476-7. DOI: 10.1007/978-3-642-54476-7\_1-1.
- [Fot+21] Johannes Fottner u. a. „Autonomous Systems in Intralogistics – State of the Art and Future Research Challenges“. In: *Logistics Research* 14.1 (4. Feb. 2021). DOI: 10.23773/2021\_2.
- [Fra+21] Giuseppe Fragapane u. a. „Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda“. In: *European Journal of Operational Research* 294.2 (16. Okt. 2021), S. 405–426. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2021.01.019.
- [FTD25] Mohamed Amine Ferrag, Norbert Tihanyi und Merouane Debbah. *From LLM Reasoning to Autonomous AI Agents: A Comprehensive Review*. 28. Apr. 2025. DOI: 10.48550/arXiv.2504.19678.
- [GG15] David Goldsman und Paul Goldsman. „Discrete-Event Simulation“. In: *Modeling and Simulation in the Systems Engineering Life Cycle*. Hrsg. von Margaret L. Loper. London: Springer London, 2015, S. 103–109. ISBN: 978-1-4471-5633-8 978-1-4471-5634-5. DOI: 10.1007/978-1-4471-5634-5\_10.
- [Goo23] Google. *MLOps: Continuous Delivery und Pipelines zur Automatisierung im maschinellen Lernen*. Google Cloud. 2023. URL: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning?hl=de> (besucht am 15.04.2024).
- [Gor+24] Alexander Gordt u. a. *Asset Administration Shell - Discovery in an Industrie 4.0 System Environment*. Plattform Industrie 4.0, 15. Apr. 2024.

- [Gri15] Michael Grieves. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*. 1. März 2015.
- [Gri16] Michael Grieves. *Origins of the Digital Twin Concept*. 2016. DOI: 10.13140/RG.2.2.26367.61609.
- [Gud12a] Timm Gudehus. *Logistik 1: Grundlagen, Verfahren und Strategien*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-29358-0 978-3-642-29359-7. DOI: 10.1007/978-3-642-29359-7.
- [Gud12b] Timm Gudehus. *Logistik 2: Netzwerke, Systeme und Lieferketten*. Studienausg. der 4. aktualis. Aufl. VDI-Buch. Berlin Heidelberg: Springer Vieweg, 2012. ISBN: 978-3-642-29375-7.
- [Guo+14] Anhong Guo u. a. „A Comparison of Order Picking Assisted by Head-up Display (HUD), Cart-Mounted Display (CMD), Light, and Paper Pick List“. In: *ACM International Symposium*. ACM Press, 2014, S. 71–78. ISBN: 978-1-4503-2969-9. DOI: 10.1145/2634317.2634321.
- [GV17] Michael Grieves und John Vickers. „Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems“. In: *Transdisciplinary Perspectives on Complex Systems*. Hrsg. von Franz-Josef Kahlen, Shannon Flumerfelt und Anabela Alves. Cham: Springer International Publishing, 2017, S. 85–113. ISBN: 978-3-319-38754-3 978-3-319-38756-7. DOI: 10.1007/978-3-319-38756-7\_4.
- [Hab+21] Reinhard Haberfellner u. a. *Systems engineering: fundamentals and applications*. Corrected publication. Cham, Switzerland: Birkhäuser, 2021. ISBN: 978-3-030-13431-0.
- [Hea24] Brian Heater. *Covariant is building ChatGPT for robots*. TechCrunch. 11. März 2024. URL: <https://techcrunch.com/2024/03/11/covariant-is-building-chatgpt-for-robots/> (besucht am 26. 04. 2024).
- [Hei+25] Christoph Heinbach u. a. „Auf dem Weg zur digitalen Föderation – Ein Vorgehensmodell für die Realisierung von dezentralen Datenräumen mit Gaia-X“. In: *HMD Praxis der Wirtschaftsinformatik* 62.3 (1. Juni 2025), S. 633–655. ISSN: 2198-2775. DOI: 10.1365/s40702-024-01125-9.
- [Hen+13] Mark Hendrikx u. a. „Procedural content generation for games: A survey“. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 9.1 (Feb. 2013), S. 1–22. ISSN: 1551-6857, 1551-6865. DOI: 10.1145/2422956.2422957.
- [HH11] Michael ten Hompel und Volker Heidenblut. *Taschenlexikon Logistik: Abkürzungen, Definitionen und Erläuterungen der wichtigsten Begriffe aus Materialfluss und Logistik*. Taschenlexikon Logistik. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2011. ISBN: 978-3-642-19945-5.
- [HK15] Michael ten Hompel und Sören Kerner. „Logistik 4.0“. In: *Informatik-Spektrum* 38.3 (2015), S. 176–182. DOI: 10.1007/s00287-015-0876-y.
- [HKR25] Michael ten Hompel, Alice Kirchheim und Moritz Roidl. „Logistik für ein posthumanes Zeitalter“. In: *VDI Nachrichten* (31. Okt. 2025), S. 26–27.
- [Hom+20] Michael ten Hompel u. a. „Technical Report: LoadRunner®, a new platform approach on collaborative logistics services“. In: *Volume 2020 (2020)*, Issue 10. DOI: 10.2195/LJ\_NOTREV\_TENHOMPEL\_EN\_202010\_01.

- [Hom+25] Michael ten Hompel u. a. *Das Digitale Kontinuum*. Whitepaper 39. Dortmund: Fraunhofer-Gesellschaft, 1. Juli 2025. DOI: 10.24406/publica-4922.
- [HS10] Michael ten Hompel und Thorsten Schmidt. *Warehouse Management: Organisation und Steuerung von Lager- und Kommissioniersystemen*. 4., neu bearb. Aufl. Intralogistik. Berlin: Springer, 2010. 354 S. ISBN: 978-3-642-03185-4 978-3-642-03184-7.
- [HSB11] Michael ten Hompel, Volker Sadowsky und Maria Beck. *Kommissionierung: Materialflusssysteme 2 - Planung und Berechnung der Kommissionierung in der Logistik*. Intralogistik. Springer, 2011. ISBN: 978-3-540-29940-0.
- [HSD18] Michael ten Hompel, Thorsten Schmidt und Johannes Dregger. *Materialflusssysteme: Förder- und Lagertechnik*. 4. Auflage. VDI-Buch. Springer, 2018. ISBN: 978-3-662-56181-2.
- [Hu+17] Haoyuan Hu u. a. *Solving a New 3D Bin Packing Problem with Deep Reinforcement Learning Method*. 19. Aug. 2017. DOI: 10.48550/arXiv.1708.05930. arXiv: 1708.05930.
- [Hua+04a] Hui-Min Huang u. a. „Autonomy Measures for Robots“. In: *Dynamic Systems and Control, Parts A and B*. ASME 2004 International Mechanical Engineering Congress and Exposition. Anaheim, California, USA: ASME, 1. Jan. 2004, S. 1241–1247. ISBN: 978-0-7918-4706-0. DOI: 10.1115/IMECE2004-61812.
- [Hua+04b] Hui-Min Huang u. a. „Specifying Autonomy Levels for Unmanned Systems: Interim Report“. In: *SPIE Defense and Security Symposium*. Orlando, Florida, USA, 2004.
- [Hua+05] Hui-Min Huang u. a. „Autonomy levels for unmanned systems (ALFUS) framework: an update“. In: *Proceedings of SPIE*. SPIE. Hrsg. von Grant R. Gerhart, Charles M. Shoemaker und Douglas W. Gage. Bd. 5804. Unmanned Ground Vehicle Technology VII. Bellingham, Washington State, USA, 27. Mai 2005, S. 439–448. DOI: 10.1117/12.603725.
- [Hua+07] Hui-Min Huang u. a. „Characterizing unmanned system autonomy: contextual autonomous capability and level of autonomy analyses“. In: *SPIE Defense and Security Symposium*. Hrsg. von Grant R. Gerhart, Douglas W. Gage und Charles M. Shoemaker. Orlando, Florida, USA, 27. Apr. 2007. DOI: 10.1117/12.719894.
- [Hua06] Hui-Min Huang. „The Autonomy Levels For Unmanned Systems (ALFUS) Framework: Interim Results“. In: *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*. Gaithersburg, Maryland, USA, Aug. 2006.
- [Hua07] Hui-Min Huang. „Autonomy levels for unmanned systems (ALFUS) framework: safety and application issues“. In: *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*. PerMIS07: Performance Metrics for Intelligent Systems 2007. Washington D.C.: ACM, 28. Aug. 2007, S. 48–53. ISBN: 978-1-59593-854-1. DOI: 10.1145/1660877.1660883.
- [Hün19] Felix Hüning. *Embedded Systems für IoT*. Berlin [Heidelberg]: Springer Vieweg, 2019. 195 S. ISBN: 978-3-662-57900-8.

- [Jes+21] Christian Jestel u. a. „Obtaining Robust Control and Navigation Policies for Multi-robot Navigation via Deep Reinforcement Learning“. In: *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. 2021 7th International Conference on Automation, Robotics and Applications (ICARA). Feb. 2021, S. 48–54. DOI: 10.1109/ICARA51699.2021.9376457.
- [Kai+21] Pascal Kaiser u. a. „Generating Highly Constrained Warehouse Layouts Using Answer Set Programming“. In: *Volume 2021 (2021)*. Medium: application/pdf Publisher: Wissenschaftliche Gesellschaft für Technische Logistik, Issue 17. DOI: 10.2195/LJ\_PROC\_KAISER\_EN\_202112\_01.
- [Kai+23] Pascal Kaiser u. a. „Towards Finding Optimal Solutions For Constrained Warehouse Layouts Using Answer Set Programming“. In: *4th Conference on Production Systems and Logistics CPSL 2023 (2023)*.
- [Kal+21] Fadil Kallat u. a. „Automatic Building of a Repository for Component-based Synthesis of Warehouse Simulation Models“. In: *Procedia CIRP* 104 (2021), S. 1440–1445. ISSN: 22128271. DOI: 10.1016/j.procir.2021.11.243.
- [Kha+20] Ahmed Khalifa u. a. *PCGRL: Procedural Content Generation via Reinforcement Learning*. 13. Aug. 2020. DOI: 10.48550/arXiv.2001.09212.
- [KKH23] Dominik Kreuzberger, Niklas Kühl und Sebastian Hirschl. „Machine Learning Operations (MLOps): Overview, Definition, and Architecture“. In: *IEEE Access* 11 (2023), S. 31866–31879. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3262138.
- [Klo+23] Patrick Klokowski u. a. „evoBOT – Design and Learning-Based Control of a Two-Wheeled Compound Inverted Pendulum Robot“. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Detroit, MI, USA, 1. Okt. 2023, S. 10425–10432. ISBN: 978-1-66549-190-7. DOI: 10.1109/IROS55552.2023.10342128.
- [KS17] Isaac Karth und Adam M. Smith. „WaveFunctionCollapse is constraint solving in the wild“. In: *Proceedings of the 12th International Conference on the Foundations of Digital Games*. FDG'17: International Conference on the Foundations of Digital Games 2017. Hyannis Massachusetts: ACM, 14. Aug. 2017, S. 1–10. ISBN: 978-1-4503-5319-9. DOI: 10.1145/3102071.3110566.
- [KW22] Henning Kagermann und Wolfgang Wahlster. „Ten Years of Industrie 4.0“. In: *Sci* 4.26 (2022). ISSN: 2413-4155. DOI: 10.3390/sci4030026.
- [Laß22] Dietmar Laß. *Fakt oder Fiktion? Technologien und Use Cases für das (Industrial) Metaverse*. Berlin: Fraunhofer-Verbund IUK-Technologie, 2022.
- [Li+24] Kunpeng Li u. a. „A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses“. In: *Transportation Research Part E: Logistics and Transportation Review* 185 (1. Mai 2024), S. 103518. ISSN: 1366-5545. DOI: 10.1016/j.tre.2024.103518.
- [Lin+19] Juntong Lin u. a. „End-to-end Decentralized Multi-robot Navigation in Unknown Complex Environments via Deep Reinforcement Learning“. In: *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2019 IEEE International Conference on Mechatronics and

- Automation (ICMA). ISSN: 2152-744X. Aug. 2019, S. 2493–2500. DOI: 10.1109/ICMA.2019.8816208.
- [Liu+21] Jialin Liu u. a. „Deep learning for procedural content generation“. In: *Neural Computing and Applications* 33.1 (1. Jan. 2021), S. 19–37. ISSN: 1433-3058. DOI: 10.1007/s00521-020-05383-8.
- [Liu+25] Bang Liu u. a. *Advances and Challenges in Foundation Agents: From Brain-Inspired Intelligence to Evolutionary, Collaborative, and Safe Systems*. 2. Aug. 2025. DOI: 10.48550/arXiv.2504.01990.
- [LL20] Bo Li und Hongbin Liang. „Multi-Robot Path Planning Method Based on Prior Knowledge and Q-learning Algorithms“. In: *Journal of Physics: Conference Series* 1624.4 (1. Okt. 2020), S. 042008. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1624/4/042008.
- [LLC25a] Open Navigation LLC. *NavFn Planner — Nav2 1.0.0 documentation*. 2025. URL: <https://docs.nav2.org/configuration/packages/configuring-navfn.html> (besucht am 12. 11. 2025).
- [LLC25b] Open Navigation LLC. *Smac Planner — Nav2 1.0.0 documentation*. 2025. URL: <https://docs.nav2.org/configuration/packages/configuring-smac-planner.html> (besucht am 12. 11. 2025).
- [LN21] C. Karen Liu und Dan Negrut. „The Role of Physics-Based Simulators in Robotics“. In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.1 (3. Mai 2021), S. 35–58. ISSN: 2573-5144, 2573-5144. DOI: 10.1146/annurev-control-072220-093055.
- [Lóp+20] Christian E. López u. a. „Deep Reinforcement Learning for Procedural Content Generation of 3D Virtual Environments“. In: *Journal of Computing and Information Science in Engineering* 20.5 (1. Okt. 2020), S. 051005. ISSN: 1530-9827, 1944-7078. DOI: 10.1115/1.4046293.
- [LR11] Christian Laviola und Sima Rustom. *Planungsleitfaden Zukunft Industriebau. Teil E: Strukturen zukunftsfähiger Industriebauten*. Bd. 3. Forschungsinitiative ZukunftBau 2756. Stuttgart: Fraunhofer-IRB-Verlag, 2011. 366 S. ISBN: 978-3-8167-8518-7.
- [Lu+24] Jack Lu u. a. „SceneControl: Diffusion for Controllable Traffic Scene Generation“. In: 2024 IEEE International Conference on Robotics and Automation (ICRA 2024). Yokohama, Japan, Mai 2024, S. 16908–16914.
- [Mak+21] Viktor Makoviychuk u. a. *Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning*. 25. Aug. 2021. DOI: 10.48550/arXiv.2108.10470.
- [Mar16] Heinrich Martin. *Transport- und Lagerlogistik: Systematik, Planung, Einsatz und Wirtschaftlichkeit*. 10. Auflage. Wiesbaden: Springer Vieweg, 2016. 567 S. ISBN: 978-3-658-14551-4. DOI: 10.1007/978-3-658-14552-1.
- [McC+24] Robert McCarthy u. a. *Towards Generalist Robot Learning from Internet Video: A Survey*. 30. Apr. 2024.
- [MDP22] Mario Angos Mediavilla, Fabian Dietrich und Daniel Palm. „Review and analysis of artificial intelligence methods for demand forecasting in supply chain management“. In: *Procedia CIRP*. Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022 107 (1. Jan. 2022), S. 1126–1131. ISSN: 2212-8271. DOI: 10.1016/j.procir.2022.05.119.

- [Meh+20] Bhairav Mehta u. a. „Active Domain Randomization“. In: *Proceedings of the Conference on Robot Learning*. Conference on Robot Learning. ISSN: 2640-3498. PMLR, 12. Mai 2020, S. 1162–1176.
- [Mir+23] Aymen Mir u. a. *Generating Continual Human Motion in Diverse 3D Scenes*. 30. Okt. 2023. DOI: 10.48550/arXiv.2304.02061.
- [MJV25] Anike Murrenhoff, Jan Philipp Jäkel und Uday Venkatadri. „Towards a Dataset of Realistic 3D Intralogistics Scenes for AI Applications“. In: *IFAC-PapersOnLine* 59.10 (2025), S. 2503–2508. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2025.09.421.
- [MK98] Hiroshi Murata und Ernest S. Kuh. „Sequence-pair based placement method for hard /soft/pre-placed modules“. In: *Proceedings of the 1998 international symposium on Physical design*. ISPD '98. New York, NY, USA: Association for Computing Machinery, 1. Apr. 1998, S. 167–172. ISBN: 978-1-58113-021-8. DOI: 10.1145/274535.274560.
- [MKV20] Andreja Malus, Dominik Kozjek und Rok Vrabič. „Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning“. In: *CIRP Annals* 69.1 (1. Jan. 2020), S. 397–400. ISSN: 0007-8506. DOI: 10.1016/j.cirp.2020.04.001.
- [MM89] Friedemann Mattern und Horst Mehl. „Diskrete Simulation - Prinzipien und Probleme der Effizienzsteigerung durch Parallelisierung“. In: *Informatik Spektrum* 12.4 (1989). Publisher: Springer-Verlag, S. 198–210.
- [Moc21] Andreas Mockenhaupt. „Maschinelles Lernen“. In: *Digitalisierung und Künstliche Intelligenz in der Produktion*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, S. 133–163. ISBN: 978-3-658-32772-9. DOI: 10.1007/978-3-658-32773-6\_6.
- [Mos17] Christian Mosblech. *Beitrag zur automatisierten Layoutsynthese intralogistischer Materialflusssysteme*. Logistik für die Praxis. OCLC: 1029450557. Dortmund: Verlag Praxiswissen, 2017. 310 S. ISBN: 978-3-86975-128-3.
- [Mul+22] Marcel Muller u. a. „Towards Deadlock Handling with Machine Learning in a Simulation-Based Learning Environment“. In: 2022 Winter Simulation Conference (WSC). Singapore: IEEE, 11. Dez. 2022, S. 1485–1496. ISBN: 978-1-66547-661-4. DOI: 10.1109/WSC57314.2022.10015270.
- [Mur+22] Fabio Muratore u. a. „Robot Learning From Randomized Simulations: A Review“. In: *Frontiers in Robotics and AI* 9 (2022). ISSN: 2296-9144.
- [Mur23] Anike Murrenhoff. „Requirements for Generating Learning Environments for Autonomous Systems Behavior in a Digital Continuum“. In: *Progress in Material Handling Research*. International Material Handling Research Colloquium (IMHRC). Bd. 16th Proceedings. Dresden, 2023. DOI: 10.24406/publica-2029.
- [Mys22] Stylianos Mystakidis. „Metaverse“. In: *Encyclopedia* 2.1 (März 2022), S. 486–497. ISSN: 2673-8392. DOI: 10.3390/encyclopedia2010031.
- [Neu51] John Von Neumann. „The General and Logical Theory of Automata“. In: John von Neumann – Collected Works 5 (1951). Hrsg. von Q. H. Taub. Pergamon Press 1963, S. 288–328.
- [NT25] Uche Nweje und Moyosore Taiwo. „Leveraging Artificial Intelligence for predictive supply chain management, focus on how AI- driven tools are revolutionizing demand forecasting and inventory optimization“.

- In: *International Journal of Science and Research Archive* 14.1 (30. Jan. 2025), S. 230–250. ISSN: 2582-8185. DOI: 10.30574/ijra.2025.14.1.0027.
- [NVI23] NVIDIA. 8. *Replicator SceneBlox tutorial — Omniverse Robotics documentation*. 23. März 2023. URL: [https://docs.omniverse.nvidia.com/app\\_isaacsim/app\\_isaacsim/tutorial\\_replicator\\_sceneblox.html](https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/tutorial_replicator_sceneblox.html) (besucht am 23. 03. 2023).
- [NVI25] NVIDIA. *Nova Carter — Isaac Sim Documentation*. 10. Nov. 2025. URL: [https://docs.isaacsim.omniverse.nvidia.com/5.1.0/assets/nova\\_carter\\_landing\\_page.html](https://docs.isaacsim.omniverse.nvidia.com/5.1.0/assets/nova_carter_landing_page.html) (besucht am 12. 11. 2025).
- [ÖGM14] Ö. Öztürkoglu, K. R. Gue und R. D. Meller. „A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points“. In: *European Journal of Operational Research* 236.1 (1. Juli 2014), S. 382–394. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2013.12.023.
- [OMG17] Object Management Group® (OMG®). *OMG® Unified Modeling Language® (OMG UML®)*. Version v2.5.1. Dez. 2017. URL: <https://www.omg.org/spec/UML/>.
- [OMG23] Object Management Group® (OMG®). *OMG Systems Modeling Language™ (SysML®) - Part 1: Language Specification*. Version Version 2.0 Beta 1. Juni 2023. URL: <https://www.omg.org/spec/SysML/2.0/Beta1/Language/PDF>.
- [Ope18a] OpenAI. *Introduction to Reinforcement Learning - Part 1: Key Concepts in RL*. Spinning Up documentation. 2018. URL: [https://spinningup.openai.com/en/latest/spinningup/rl\\_intro.html](https://spinningup.openai.com/en/latest/spinningup/rl_intro.html) (besucht am 13. 07. 2023).
- [Ope18b] OpenAI. *Introduction to Reinforcement Learning - Part 2: Kinds of RL Algorithms*. Spinning Up documentation. 2018. URL: [https://spinningup.openai.com/en/latest/spinningup/rl\\_intro2.html](https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html) (besucht am 13. 07. 2023).
- [Ope22] OpenAI. *MuJoCo - Gym Documentation*. 2022. URL: <https://www.gymnasium.dev/environments/mujoco/> (besucht am 12. 04. 2024).
- [Pef+07] Ken Peffers u. a. „A Design Science Research Methodology for Information Systems Research“. In: *Journal of Management Information Systems* 24.3 (Dez. 2007), S. 45–77. ISSN: 0742-1222, 1557-928X. DOI: 10.2753/MIS0742-1222240302.
- [Per22] Peraton. *The Digital Continuum of Business Transformation*. Reston, Virginia, USA, 2022.
- [Per85] Ken Perlin. „An image synthesizer“. In: *ACM SIGGRAPH Computer Graphics* 19.3 (Juli 1985), S. 287–296. ISSN: 0097-8930. DOI: 10.1145/325165.325247.
- [PMD21] Pablo Pérez-Gosende, Josefa Mula und Manuel Díaz-Madroñero. „Facility layout planning. An extended literature review“. In: *International Journal of Production Research* 59.12 (18. Juni 2021), S. 3777–3816. ISSN: 0020-7543, 1366-588X. DOI: 10.1080/00207543.2021.1897176.
- [Pri23] Irfan Fachrudin Priyanta. „Towards 6G-Driven Digital Continuum in Logistics [Auf dem Weg zu einem 6G-gestützten digitalen Kontinuum in der Logistik]“. In: *Logistics Journal: Proceedings* (2023). ISSN: 2192-9084. DOI: 10.2195/lj\_proc\_priyanta\_en\_202310.

- [Pru86] Przemyslaw Prusinkiewicz. „Graphical applications of Lsystems“. In: *Proceedings of Graphics Interface '86 / Vision Interface '86* (1986), S. 247–253.
- [RE423] RE4DY. D2.2: *Digital 4.0 Continuum Reference Framework - v1*. Horizon Europe Grant Agreement ID: 101058384. RE4DY Manufacturing Data Networks, 7. Apr. 2023.
- [RM22] Moritz Roidl und Anike Murrenhoff. „Digitaler und Cyberphysischer Zwillling“. In: *Silicon Economy*. Hrsg. von Michael ten Hompel, Michael Henke und Boris Otto. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, S. 193–228. ISBN: 978-3-662-63955-9. DOI: 10.1007/978-3-662-63956-6\_10.
- [RND10] Stuart J. Russell, Peter Norvig und Ernest Davis. *Artificial intelligence: a modern approach*. 3rd ed. Prentice Hall series in artificial intelligence. Upper Saddle River: Prentice Hall, 2010. 1132 S. ISBN: 978-0-13-604259-4.
- [Roi22] Moritz Johannes Roidl. „Zur innerbetrieblichen Logistik - Axiomatik und Betrachtung als kinodynamisches System“. Diss. Dortmund: Technischen Universität Dortmund, 2022.
- [Rom+24] Bernardino Romera-Paredes u. a. „Mathematical discoveries from program search with large language models“. In: *Nature* 625.7995 (18. Jan. 2024), S. 468–475. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06924-6.
- [Rop09] Günter Ropohl. *Allgemeine Technologie : eine Systemtheorie der Technik*. KIT Scientific Publishing, 2009. ISBN: 978-3-86644-374-7. DOI: 10.26530/OAPEN\_422388.
- [RP04] Timothy Roden und Ian Parberry. „From Artistry to Automation: A Structured Methodology for Procedural Content Creation“. In: Bd. 3166. 1. Sep. 2004, S. 151–156. ISBN: 978-3-540-22947-6. DOI: 10.1007/978-3-540-28643-1\_19.
- [RP24] Reidar Riveland und Alexandre Pouget. „Natural language instructions induce compositional generalization in networks of neurons“. In: *Nature Neuroscience* (18. März 2024). ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-024-01607-5.
- [RT20] Sebastian Risi und Julian Togelius. „Increasing generality in machine learning through procedural content generation“. In: *Nature Machine Intelligence* 2.8 (3. Aug. 2020), S. 428–436. ISSN: 2522-5839. DOI: 10.1038/s42256-020-0208-z.
- [Rue+20] Laura von Rueden u. a. „Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions“. In: *Advances in Intelligent Data Analysis XVIII*. Hrsg. von Michael R. Berthold, Ad Feelders und Georg Krempel. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, S. 548–560. ISBN: 978-3-030-44584-3. DOI: 10.1007/978-3-030-44584-3\_43.
- [RWP19] Heiner Reinhardt, Marek Weber und Matthias Putz. „A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing“. In: *Procedia CIRP*. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019 81 (1. Jan. 2019), S. 121–126. ISSN: 2212-8271. DOI: 10.1016/j.procir.2019.03.022.

- [SB18] Richard S. Sutton und Andrew G. Barto. *Reinforcement learning: an introduction*. Second edition. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018. 526 S. ISBN: 978-0-262-03924-6.
- [Sch18] Michael Schmidt. *Distribution Center Design Process: ein systemtechnikorientiertes Vorgehensmodell zur Konzeptplanung von Logistikzentren*. Logistik für die Praxis. Dortmund: Verlag Praxiswissen, 2018. 219 S. ISBN: 978-3-86975-129-0.
- [Sch20] Steffen Schieweck. *Beitrag zur Antwortmengenprogrammierung in der Intralogistik*. Hrsg. von Michael ten Hompel. 1. Auflage. Logistik für die Praxis. Dortmund: Verlag Praxiswissen, 2020. 245 S. ISBN: 978-3-86975-152-8.
- [SFN21] Iqbal H. Sarker, Md Hasan Furhad und Raza Nowrozy. „AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions“. In: *SN Computer Science* 2.3 (26. März 2021), S. 173. ISSN: 2661-8907. DOI: 10.1007/s42979-021-00557-0.
- [SGM25] Omid Safari, Maxi Grobis und Marc Münnich. „Design criteria for automated simulation model generation“. In: *Simulation in Produktion und Logistik 2025 21. ASIM-Fachtagung*. 21. ASIM-Fachtagung Simulation in Produktion und Logistik. Technische Universität Dresden, 2025. DOI: 10.25368/2025.233.
- [Sha+10] Mike Shafto u. a. *DRAFT Modeling, Simulation, Information Technology & Processing Roadmap*. 11. NASA National Aeronautics und Space Administration, 2010.
- [Sha+24] Atefeh Shahroudnejad u. a. *A Novel Framework for Automated Warehouse Layout Generation*. 12. Juli 2024. DOI: 10.48550/arXiv.2407.08633.
- [Sie11] Roland Siegart. *Introduction to autonomous mobile robots*. Unter Mitarb. von Illah Reza Nourbakhsh und Davide Scaramuzza. 2nd ed. Intelligent robotics and autonomous agents series. Cambridge, Massachusetts: MIT Press, 2011. 1 S. ISBN: 978-0-262-01535-6 978-0-262-29532-1.
- [Sim25] Simio. *The Digital Continuum: The Digital Transformation Roadmap*. Simio. 2025. URL: <https://www.simio.com/de/whitepapers/the-digital-continuum-the-digital-transformation-roadmap/> (besucht am 20.09.2025).
- [Sme+11] R.M. Smelik u. a. „A declarative approach to procedural modeling of virtual worlds“. In: *Computers & Graphics* 35.2 (Apr. 2011), S. 352–363. ISSN: 00978493. DOI: 10.1016/j.cag.2010.11.011.
- [Smi+24] Igor G. Smit u. a. *Learning Efficient and Fair Policies for Uncertainty-Aware Collaborative Human-Robot Order Picking*. 9. Apr. 2024. DOI: 10.48550/arXiv.2404.08006.
- [Sol20] Matthews Automation Solutions. *How to Successfully Implement Autonomous Mobile Robots into Order Fulfillment*. Whitepaper. Waukesha, WI, USA, 2020.
- [SP21] Marian Süße und Matthias Putz. „Generative design in factory layout planning“. In: *Procedia CIRP* 99 (2021), S. 9–14. ISSN: 22128271. DOI: 10.1016/j.procir.2021.03.002.

- [Spr+20] Timothy Sprock u. a. *Theory of Discrete Event Logistics Systems (DELS) specification*. NIST IR 8262. Gaithersburg, MD: National Institute of Standards and Technology, Juni 2020. DOI: 10.6028/NIST.IR.8262.
- [Sta73] Herbert Stachowiak. *Allgemeine Modelltheorie*. Wien New York: Springer, 1973. 494 S. ISBN: 978-3-211-81106-1 978-0-387-81106-2.
- [Ste+24] Lennart M. Steinbacher u. a. „Automatic model generation for material flow simulations of Third-Party Logistics“. In: *Journal of Intelligent Manufacturing* 35.8 (1. Dez. 2024), S. 3857–3874. ISSN: 1572-8145. DOI: 10.1007/s10845-023-02257-3.
- [STN16] Noor Shaker, Julian Togelius und Mark J. Nelson. *Procedural Content Generation in Games*. Computational Synthesis and Creative Systems. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-42714-0 978-3-319-42716-4. DOI: 10.1007/978-3-319-42716-4.
- [Str21] Eliza Strickland. *The Turbulent Past and Uncertain Future of Artificial Intelligence - Is there a way out of AI's boom-and-bust cycle?* IEEE Spectrum. 30. Sep. 2021. URL: <https://spectrum.ieee.org/history-of-ai> (besucht am 04.08.2025).
- [Sur+20] Hartmut Surmann u. a. *Deep Reinforcement learning for real autonomous mobile robot navigation in indoor environments*. 28. Mai 2020. DOI: 10.48550/arXiv.2005.13857.
- [Sym+22] Georgios Symeonidis u. a. „MLOps - Definitions, Tools and Challenges“. In: *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas, NV, USA: IEEE, 26. Jan. 2022, S. 0453–0460. ISBN: 978-1-6654-8303-2. DOI: 10.1109/CCWC54503.2022.9720902.
- [Tha+18] Shantanu Thakar u. a. „Towards Time-Optimal Trajectory Planning for Pick-and-Transport Operation with a Mobile Manipulator“. In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). ISSN: 2161-8089. Aug. 2018, S. 981–987. DOI: 10.1109/COASE.2018.8560446.
- [The+24] Andre Thevapalan u. a. „A Hybrid System for Warehouse Layout Planning Based on Answer Set Programming and Conditional Expert Knowledge“. In: *TAASP 2024 Workshop on Trends and Applications of Answer Set Programming*. Klagenfurt, Austria, 26. Nov. 2024.
- [Tog+11] Julian Togelius u. a. „Search-Based Procedural Content Generation: A Taxonomy and Survey“. In: *IEEE Transactions on Computational Intelligence and AI in Games* 3.3 (Sep. 2011), S. 172–186. ISSN: 1943-068X, 1943-0698. DOI: 10.1109/TCIAIG.2011.2148116.
- [Tor+19] Ruben Rodriguez Torrado u. a. *Bootstrapping Conditional GANs for Video Game Level Generation*. 3. Okt. 2019.
- [TRN02] Jose Tavares, Carlos Ramos und Jose Neves. „Addressing The Facilities Layout Design Problem Through Constraint Logic Programming“. In: *Journal for Manufacturing Science and Production* 4.3 (März 2002), S. 149–168. ISSN: 2191-0375, 0793-6648. DOI: 10.1515/IJMSP.2002.4.3.149.

- [Tru+22] Joanne Truong u. a. „Rethinking Sim2Real: Lower Fidelity Simulation Leads to Higher Sim2Real Transfer in Navigation“. In: 6th Conference on Robot Learning (CoRL 2022). Auckland, New Zealand, 2022.
- [Tsi+21] Efthimios Tsiogas u. a. „Pallet detection and docking strategy for autonomous pallet truck AGV operation“. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic: IEEE, 27. Sep. 2021, S. 3444–3451. ISBN: 978-1-66541-714-3. DOI: 10.1109/IROS51168.2021.9636270.
- [UB21] Hendrik Unger und Frank Börner. „Reinforcement Learning for Layout Planning – Modelling the Layout Problem as MDP“. In: *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*. Hrsg. von Alexandre Dolgui u. a. Bd. 632. Cham: Springer International Publishing, 2021, S. 471–479.
- [VDI 2199] Verein Deutscher Ingenieure (VDI), Hrsg. *VDI 2199:2012-01, Empfehlungen für bauliche Planungen beim Einsatz von Flurförderzeugen*. 2012.
- [VDI 2411] Verein Deutscher Ingenieure (VDI), Hrsg. *VDI 2411 - Begriffe und Erläuterungen im Förderwesen*. 1970.
- [VDI 2510-1] Verein Deutscher Ingenieure (VDI), Hrsg. *VDI 2510 Blatt 1:2009-12, Infrastruktur und periphere Einrichtungen für Fahrerlose Transportsysteme (FTS)*. 2009.
- [VDI 3590-1] Verein Deutscher Ingenieure (VDI), Hrsg. *VDI 3590 Blatt 1:1994-04, Kommissioniersysteme; Grundlagen*. 1994.
- [VDI 3633-1] Verein Deutscher Ingenieure (VDI), Hrsg. *VDI 3633 Blatt 1:2014-12, Simulation von Logistik-, Materialfluss- und Produktionssystemen - Grundlagen*. 2014.
- [VDI 4490] Verein Deutscher Ingenieure (VDI), Hrsg. *VDI 4490:2007-05, Operative Logistikkennzahlen von Wareneingang bis Versand*. 2007.
- [Ver+20] Richa Verma u. a. *A Generalized Reinforcement Learning Algorithm for Online 3D Bin-Packing*. 1. Juli 2020. DOI: 10.48550/arXiv.2007.00463.
- [Wah17] Wolfgang Wahlster. „Künstliche Intelligenz als Grundlage autonomer Systeme“. In: *Informatik-Spektrum* 40.5 (Okt. 2017), S. 409–418. ISSN: 0170-6012, 1432-122X. DOI: 10.1007/s00287-017-1049-y.
- [WCW24] Shao-Ci Wu, Wei-Yu Chiu und Chien-Feng Wu. „Deep Reinforcement Learning for Task Assignment and Shelf Reallocation in Smart Warehouses“. In: *IEEE Access* 12 (2024), S. 58915–58926. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3392752.
- [Wen+19] Sigrid Wenzel u. a. „Trends in Automatic Composition of Structures for Simulation Models in Production and Logistics“. In: *2019 Winter Simulation Conference (WSC)*. 2019 Winter Simulation Conference (WSC). National Harbor, MD, USA: IEEE, Dez. 2019, S. 2190–2200. ISBN: 978-1-72813-283-9. DOI: 10.1109/WSC40007.2019.9004959.
- [Wen18] Sigrid Wenzel. „Simulation logistischer Systeme“. In: *Modellierung logistischer Systeme*. Hrsg. von Horst Tempelmeier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, S. 1–34. ISBN: 978-3-662-57770-7 978-3-662-57771-4. DOI: 10.1007/978-3-662-57771-4\_1.
- [WHK22] Mike Wesselhöft, Johannes Hinckeldeyn und Jochen Kreuzfeldt. „Controlling Fleets of Autonomous Mobile Robots with Reinforcement Learning“.

- ning: A Brief Survey“. In: *Robotics* 11.5 (Okt. 2022), S. 85. ISSN: 2218-6581. DOI: 10.3390/robotics11050085.
- [Wie+24] Marvin Wiedemann u. a. *O3dynSimModel · GitLab*. GitLab. 23. Aug. 2024. URL: <https://git.openlogisticsfoundation.org/silicon-economy/simulation-model/o3dynsimmodel> (besucht am 08. 04. 2024).
- [Wir18] Gabler Wirtschaftslexikon. *Technologiekonvergenz*. In: 14. Feb. 2018. URL: <https://wirtschaftslexikon.gabler.de/definition/technologiekonvergenz-54383/version-277422>.
- [Xia+23] Zhou Xian u. a. *Towards Generalist Robots: A Promising Paradigm via Generative Simulation*. 29. Aug. 2023. DOI: 10.48550/arXiv.2305.10455.
- [YDY24] Shenglu Yu, Shimin Du und Chang Yang. „A Deep Reinforcement Learning Floorplanning Algorithm Based on Sequence Pairs“. In: *Applied Sciences* 14.7 (Jan. 2024), S. 2905. ISSN: 2076-3417. DOI: 10.3390/app14072905.
- [Yu+23] Tianhe Yu u. a. *Scaling Robot Learning with Semantically Imagined Experience*. 22. Feb. 2023. DOI: 10.48550/arXiv.2302.11550.
- [Zha+21] Hang Zhao u. a. *Online 3D Bin Packing with Constrained Deep Reinforcement Learning*. 4. März 2021. DOI: 10.48550/arXiv.2006.14978.
- [Zha+24] Mingyuan Zhang u. a. „MotionDiffuse: Text-Driven Human Motion Generation With Diffusion Model“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.6 (Juni 2024), S. 4115–4128. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2024.3355414.
- [Zho+23] Zhehua Zhou u. a. *Towards Building AI-CPS with NVIDIA Isaac Sim: An Industrial Benchmark and Case Study for Robotics Manipulation*. 31. Juli 2023.
- [Zhu+23] Wentao Zhu u. a. *Human Motion Generation: A Survey*. 15. Nov. 2023. DOI: 10.48550/arXiv.2307.10894.
- [ZQW20] Wenshuai Zhao, Jorge Pena Queralta und Tomi Westerlund. „Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey“. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2020 IEEE Symposium Series on Computational Intelligence (SSCI). Canberra, ACT, Australia: IEEE, 1. Dez. 2020, S. 737–744. ISBN: 978-1-72812-547-3. DOI: 10.1109/SSCI47803.2020.9308468.



# Anhang

## A.1 Literatur zu Steuerungsproblemen von AMR in der Intralogistik

Der erste Teil des Anhangs liefert eine Übersicht über Literatur zu Steuerungsproblemen von AMR in der Intralogistik, bei denen RL-Ansätze zum Einsatz kommen.

**Tabelle A.1** Übersicht über Literatur zu Steuerungsproblemen von AMR in Materialflusssystemen

Nr.	Quelle	Steuerungsprobleme	Architektur	Agentenmodell	Domäne
1	[MKV20]	Dispatching	zentrales Lernen, dezentrale Ausführung	MARL	Produktion
2	[Bae+19]	Routing	zentrales Lernen, dezentrale Ausführung	SARL	Computerspiel
3	[Lin+19]	Routing, Navigation	zentrales Lernen, dezentrale Ausführung	SARL	offen
4	[LL20]	Routing	zentrales Lernen, dezentrale Ausführung	MARL	offen
5	[Sur+20]	Routing, Navigation	zentrales Lernen, dezentrale Ausführung	MARL	offen
6	[Jes+21]	Routing, Navigation	zentrales Lernen, dezentrale Ausführung	MARL	offen
7	[Mul+22]	Deadlockvermeidung und -auflösung	zentrales Lernen, zentrale Ausführung	SARL	Produktion
8	[Hu+17]	Bin packing	zentrales Lernen, zentrale Ausführung	SARL	offen
9	[Ver+20]	Bin packing	zentrales Lernen, zentrale Ausführung	SARL	offen
10	[Zha+21]	Bin packing	zentrales Lernen, zentrale Ausführung	SARL	offen
11	[Smi+24]	Zuweisung Roboter - Mensch	zentrales Lernen, zentrale Ausführung	SARL	Lager
12	[EI21]	Dispatching, Navigation	zentrales Lernen, dezentrale Ausführung	MARL	offen
13	[Li+24]	Dispatching, Routing, Kollisionsvermeidung	zentrales Lernen, zentrale Ausführung	SARL	Lager
14	[WCW24]	Dispatching, Kollisionsvermeidung, Lagerplatzzuweisung	zentrales Lernen, zentrale Ausführung	SARL	Lager

## A.2 Vorveröffentlichungen und Forschungsprojekte

Teile der Ergebnisse dieser Arbeit wurden durch die Autorin bereits im Vorfeld veröffentlicht. Die Vorveröffentlichungen sind innerhalb der Arbeit referenziert und in das allgemeine Literaturverzeichnis integriert:

- Moritz Roidl und Anike Murrenhoff. „Digitaler und Cyberphysischer Zwilling“. In: *Silicon Economy*. Hrsg. von Michael ten Hompel, Michael Henke und Boris Otto. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, S. 193–228. ISBN: 978-3-662-63955-9. DOI: 10.1007/978-3-662-63956-6\_10
- Anike Murrenhoff. „Requirements for Generating Learning Environments for Autonomous Systems Behavior in a Digital Continuum“. In: *Progress in Material Handling Research*. International Material Handling Research Colloquium (IMHRC). Bd. 16th Proceedings. Dresden, 2023. DOI: 10.24406/publica-2029
- Anike Murrenhoff, Jan Philipp Jäkel und Uday Venkatadri. „Towards a Dataset of Realistic 3D Intralogistics Scenes for AI Applications“. In: *IFAC-PapersOnLine* 59.10 (2025), S. 2503–2508. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2025.09.421
- Michael ten Hompel u. a. *Das Digitale Kontinuum*. Whitepaper 39. Dortmund: Fraunhofer-Gesellschaft, 1. Juli 2025. DOI: 10.24406/publica-4922

Alle Publikationen wurden gemäß der Promotionsordnung als Vorveröffentlichung von der Autorin in Absprache mit dem Erstgutachter bei der Fakultät Maschinenbau der Technischen Universität Dortmund angezeigt.

Die Vorveröffentlichungen geschahen im Rahmen folgender Forschungsprojekte, an denen die Autorin der vorliegenden Arbeit mitgewirkt hat:

- Silicon Economy (gefördert durch das Bundesministerium für Digitales und Verkehr)
- Lamarr Institut für Maschinelles Lernen und Künstliche Intelligenz (institutionell gefördert durch das Bundesministerium für Forschung, Technologie und Raumfahrt und das Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen)

### A.3 An der Forschungsarbeit beteiligte Personen

An Teilen der Forschungsarbeiten, welche in diese Arbeit eingeflossen sind bzw. auf die sie Bezug nimmt, waren neben der Autorin dieser Arbeit weitere Personen im Rahmen der Forschungsprojekte beteiligt. Die Form ihrer Beiträge ist in folgender Tabelle aufgeführt.

**Tabelle A.2** An der Forschungsarbeit beteiligte Personen

<b>Beitrag</b>	<b>Neben der Autorin beteiligte Personen</b>
Unterstützung bei der Implementierung des Generierungsverfahrens	Mia Bedarf, Sabrina Hartmann, Jan Jäkel, Julius Mackowiak
Diskussionen zum Digitalen Kontinuum	Prof. Dr. Dr. h. c. Michael ten Hompel, Univ.-Prof. Dr. habil. Dr. h. c. Michael Henke, Dr.-Ing. Moritz Roidl, Dr.-Ing. Jana Jost, Dr.-Ing. Sören Kerner, Marco Motta, Dr.-Ing. Michael Schmidt, Dr.-Ing. Sebastian Wibbeling

## A.4 Studentische Arbeiten im Kontext der Forschungsarbeit

Von der Autorin betreute Arbeiten lieferten wertvolle Eingaben, Vorarbeiten und Anregungen. Alle studentischen Arbeiten, die im Kontext der vorliegenden Ausarbeitung entstanden, sind im Folgenden chronologisch aufsteigend nach Abgabedatum aufgeführt.

Datum	Titel	Typ	Vorname	Nachname
19.07.2022	Erstellung elementarer Bausteine zur prozeduralen Generierung eines Intralogistiksystems validiert in einer 3D Simulationsumgebung	MA	Stefanie	Rempel

PA: Projektarbeit, BA: Bachelorarbeit, MA: Masterarbeit