

Beitrag zur Einarbeitung in ereignisdiskrete Simulation zur Neu- und Umplanung von Materialflusssystemen

vorgelegt von
M.Sc.Eng.
Bastian Clemens Schumacher
ORCID: 0000-0002-1661-9719

an der Fakultät V - Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Holger Kohl
Gutachter: Prof. Dr.-Ing. Günther Seliger
Gutachter: Prof. Dr. Dr. habil. Dmitry Ivanov

Tag der wissenschaftlichen Aussprache: 12. Dezember 2019

Berlin 2020

Danksagung

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkzeugmaschinen und Fabrikbetrieb der TU Berlin. Mein besonderer Dank gilt dem ehemaligen Leiter des Lehrstuhls für Montagetechnik und Fabrikbetrieb, Herrn em. Prof. Dr.-Ing. Günther Seliger. Bereits zu einem sehr frühen Zeitpunkt unterstützte er mich bei der Entfaltung meiner Interessen für die ereignisdiskrete Simulation und die Hochschullehre. Ich danke ihm für die zahlreichen kritischen Auseinandersetzungen mit meiner Arbeit während der vergangenen Jahre. Bei Herrn Prof. Dr.-Ing. Holger Kohl möchte ich mich für die Übernahme des Vorsitzes sowie für die sehr erfolgreiche Zusammenarbeit im Studiengang Global Production Engineering (GPE) bedanken. Herrn Prof. Dr. Dr. habil. Dmitry Ivanov danke ich für das der Arbeit entgegengebrachte Interesse sowie deren Begutachtung.

Bei meinen Kollegen aus der Arbeitsgruppe „Learnstruments“ möchte ich mich für die vielen kontroversen Diskussionen bedanken. Der fachliche Austausch mit Anja Steinbach, Bernd Muschard, Felix Sieckmann, Jan Menn und Mustafa Severengiz hat die Arbeit an vielen Stellen bereichert. Den Studierenden Felix Balzar und Shehab Al-Eryani danke ich für ihre Beiträge zur Umsetzung der App PSIMA. Thomas Stoffel möchte ich für unsere Diskussionen über die Potenziale der ereignisdiskreten Simulation im Gasturbinenbau danken. Arne Glodde und Uli Ringeisen gilt mein Dank für die Bereitstellung und auch manchmal sehr kurzfristige Wartung der IT-Infrastruktur.

Diese Arbeit wäre kaum gelungen, wenn meine Kolleginnen vom Studiengang GPE mir im Tagesgeschäft mit den Anliegen von zeitweise über 200 Studierenden nicht so beherzt den Rücken freigehalten hätten. Danke Anneliese Ulrich, Marina Nagel, Irina Ferreira Range und Sigrid Treptau. Danke auch an Sabine Lange für die Unterstützung bei allen organisatorischen Anliegen.

Mein größter Dank gilt den Menschen in meinem privaten Umfeld. Sarah, Deine immerwährende Unterstützung und ständige Motivation war die Basis für diese Arbeit. Ich weiß, dass ich Dich an unzähligen Tagen nahezu in den Wahnsinn getrieben habe und danke Dir für Deine endlose Geduld. Fritz, Deine Geburt war eine sehr willkommene Ablenkung. Dich wachsen zu sehen, macht mir jeden Tag aufs Neue das wirklich Wichtige bewusst. Meinen Eltern Sonja und Wolfgang danke ich für die Unterstützung bei wichtigen Entscheidungen, die mir diesen Weg geebnet haben. Danke Sarah und Sonja für das Lektorat der Arbeit.

Kurzfassung

Die vorliegende Arbeit zeigt Wege auf, wie Studierende in handelsübliche flexible ereignisdiskrete Simulationsinstrumente zur Durchführung von Simulationsstudien im Anwendungsbereich der Neu- und Umplanung von Materialflüssen eingearbeitet werden können.

Materialflüsse in Produktionssystemen werden durch Informationsflüsse gesteuert. Eine Vielzahl an Elementen wie Produkte, Maschinen oder Betriebsmittel des Materialflusses und Relationen wie Rückflüsse und mehrere mögliche Nachfolger führen schnell zu einer hohen Kompliziertheit. Dynamische und stochastische Effekte bei der Förderung, Handhabung, Lagerung und Be- und Entladung führen zu komplexen Problemen bei der Planung. Planer von Materialflusssystemen müssen in diesem Umfeld Entscheidungen mit Blick auf eine hohe Wirtschaftlichkeit und Zuverlässigkeit treffen. Simulationsstudien, die unter Nutzung eines ereignisdiskreten Simulationsinstruments durchgeführt werden, können die Planer von Materialflusssystemen dabei unterstützen. Fehlentscheidungen lassen sich so frühzeitig erkennen und korrigieren. Planungsprojekte werden in der Folge oft schneller abgeschlossen und Fehlinvestitionen vermieden.

Chancen von Simulationsstudien könnten stärker genutzt werden. Häufig haben die Planer von Materialflusssystemen keine Erfahrung in der Durchführung von Simulationsstudien. Besonders die Abstraktion durch Reduktion im Sinne des Verzichts auf die Abbildung von Details oder Idealisierung im Sinne der Vereinfachung realer Gegebenheiten sowie die Interpretation von Simulationsergebnissen erfordern Erfahrung. Weiterhin setzt die Anwendung von komplizierten Simulationsinstrumenten eine Schulung voraus, die Planer von Materialflusssystemen häufig nicht erhielten. Der Wissenserwerb über Bausteine mit innerer Ablauflogik und Objekte ohne innere Ablauflogik durch deren mehrmalige exemplarische Anwendung sowie die Einarbeitung in die Syntax und Semantik von Programmiersprachen zur Implementierung von Simulationsmodellen stehen hier im Fokus.

Neuartige Lehr- und Lernmethoden haben das Potenzial, die Produktivität des Lehrens und Lernens bei der Durchführung von Simulationsstudien zu erhöhen. Sie ermöglichen eine Entwicklung vom Frontalunterricht hin zu einer orts- und zeitunabhängigen Instruktion mit interaktiven Inhalten. Die Vorteile der effektiven Nutzung von Lizenzen für Software, der hohen Verfügbarkeit und Skalierbarkeit lassen sich durch digitale Ansätze der Instruktion ausschöpfen. Sie fördern die Gruppenarbeit. Studierende stehen im Fokus der Entwicklungen.

Die in dieser Arbeit entwickelte Lernumgebung entsteht durch die Anwendung eines Vorgehens zur schrittweisen Einarbeitung in die Nutzung von Bausteinen und Objekten, einer visuellen blockbasierten Pro-

grammiersprache zur Implementierung von Simulationsmodellen sowie eines Vorgehens zur praxisnahen Einarbeitung in die Durchführung von Simulationsstudien.

Sogenannte inkrementell komplizierter werdende Simulationsmodelle (IKS) ermöglichen die Einarbeitung des Lernenden in die Bausteine und Objekte und in den grafisch-interaktiven Modellaufbau. In schrittweise komplizierter werdenden Lerneinheiten werden an den Lernenden immer neue Aufgaben gestellt, zu deren Lösung er vorher unbekannte Objekte und Bausteine anwenden muss. In der visuellen blockbasierten Programmiersprache zur Implementierung von Simulationsmodellen (VBPIS) sind die grafischen Eigenschaften von visuellen blockbasierten Programmiersprachen mit den Funktionen von Programmiersprachen zur Implementierung von Simulationsmodellen kombiniert. Die VBPIS unterstützt das Erlernen von Syntax und Semantik von Programmiersprachen zur Implementierung von Simulationsmodellen. IKS und VBPIS sind mit Elementen des Spielens gestaltet. Bei Spielelementen handelt es sich um motivationsförderliche Anreizmechanismen wie sogenannte Level, Abzeichen oder Punkte. Die Spielelemente motivieren den Spieler zu einer häufigen Interaktion mit der Lernumgebung und ermöglichen ein unmittelbares Feedback. Die IKS und VBPIS sind in einer mobilen App gestaltet. Die App ermöglicht einen orts- und zeitunabhängigen sowie kostengünstigen Zugriff und eine hohe Verfügbarkeit von IKS und VBPIS. Am Beispiel des kommerziell verfügbaren Simulationsinstruments Tecnomatix Plant Simulation ist die prototypische App Production Simulation Application (PSIMA) partiell implementiert worden. Es wurde gezeigt, dass Studierende PSIMA mehrmals in kurzen Zeitabständen begleitend zur Lehrveranstaltung nutzten. Sie wurden in die Bedienung des ereignisdiskreten Simulationsinstruments eingearbeitet.

Zur praxisnahen Einarbeitung in die Durchführung von Simulationsstudien wurde das Vorgehen zum erfahrungsbasierten Erlernen der Durchführung von Simulationsstudien (EEDS) entwickelt. Bei diesem Vorgehen dient eine sogenannte Lernfabrik den Studierenden zur vollständigen Durchführung einer Simulationsstudie von der Datenaufnahme und Abstraktion über den Modellaufbau und das Experimentieren hin zur Ergebnisanalyse. Dabei erkannte Verbesserungen werden in der Lernfabrik vorgenommen. Mehrere Stationen, ein veränderlicher Aufbau, ein didaktischer Ansatz, die Herstellung eines physischen Produkts und realitätsnahe Prozesse kennzeichnen solch eine Lernfabrik. Die Erprobung des Vorgehens zum EEDS zeigte, dass Studierende die Bedeutung der Abstraktion und Ergebnisinterpretation bei der Durchführung von Simulationsstudien in einer realitätsnahen Umgebung erfahren. Am Beispiel der Gestaltung des Materialflusses im Nacharbeitsbereich einer Montagelinie für Motoren wird illustriert, dass PSIMA im industriellen Einsatz „Training on the Job“ unterstützt. Am Beispiel eines Materialflusses im Gasturbinenbau werden Perspektiven eröffnet, wie die Lernumgebung hin zu einer mobilen App zur Analyse von Materialflüssen weiterentwickelt werden kann. Der Beitrag der Lernumgebung zur hohen Qualität der Planung wird auf Basis der materialflusstechnischen Anwendungen diskutiert.

Abstract

This thesis shows pathways how students can be familiarized with commercial flexible discrete-event simulation tools for executing simulation studies for the planning and re-planning of material flow systems.

Information flows in production systems control material flows. A high number of elements such as products, machines or equipment for material flow and relations such as reverse flows and more than one possible successor leads soon to a high complexity. Dynamic and stochastic effects during transport, handling storage and (un-)loading lead to complex planning problems. In this area, material flow systems planners need to take decisions considering a high profitability and reliability. Simulation studies which are conducted by means of a discrete-event simulation tool can support planners in decision making. Wrong decisions can be recognized earlier and be corrected. As a result, the duration of planning projects can be decreased and investments based on wrong planning can be avoided.

Chances of simulation studies could be exploited more intensive. Often the planners of material flow systems have no experience in conducting simulation studies. Particularly, the abstraction through reduction in the sense of omission of unimportant details or idealizations in the sense of simplification of indispensable details requires experience. The application of complicated simulation tools requires training, which planners have often not experienced. The knowledge gain of building blocks with internal structural logic and objects without internal structural logic through repeated exemplary application as well as the familiarization with the syntax and semantics of programming language for the implementation of simulation models is focused.

New methods of learning and teaching have the potential to increase the productivity of teaching and learning during the conduction of simulation studies. They enable the development from teacher-centered instruction to an instruction with interactive contents regardless of time and place. Advantages such as the effective utilization of licenses and the high availability and scalability can be exploited through digital approaches of instruction. They foster group work. Students are focused in this work.

The learning environment developed in this thesis emerges from a methodology for short and self-contained learning units, that build upon each other to introduce the learner into the use of building blocks and objects, a visual block-based programming language for the implementation of simulation models and a procedure for the practical introduction to conduct simulation studies.

The so-called self-contained learning units that build upon each other (IKS) enable the introduction of the learner in the building blocks and objects as well as the graphical-interactive model formalization. In IKS

the learner must solve tasks that require the usage of previously unknown building blocks and objects. The visual block-based programming language (VBPIs) combines the graphical properties of visual block-based programming languages with functions of programming languages for the implementation of simulation models. The VBPIs supports the learning of syntax and semantic of programming languages for the implementation of simulation models. IKS and VBPIs are shaped through game elements. Game elements are motivation conducive incentive mechanisms such as so-called levels, badges and points. Game elements motivate the learners to a frequent interaction with the learning environment and enable immediate feedback. IKS and VBPIs are part of a mobile App. The App has a high availability and enables a cheap interaction with IKS and VBPIs regardless of time and place. The App Production Simulation Application (PSIMA) was prototypical implemented on the example of the commercially available simulation tool Tecnomatix Plant Simulation. It was shown that students used PSIMA repeatedly in short intervals beyond a teaching course. They have been introduced into the application of a discrete-event simulation tool.

For the practical introduction into the execution of simulation studies the procedure for experiential learning to conduct simulation studies (EEDS) was developed. A so-called learning factory enables learners to completely execute a simulation study. These studies start with the definition of goals, data collection and preparation, and then range from abstraction and modeling, the experimentation with the digital model to the interpretation of results. Identified improvements are implemented in the learning factory. A learning factory comprises more than one station, a changeable setting, a didactic approach, the production of a physical product and realistic processes. In tests of the EEDS procedure it was found out that students experienced the meaning of the abstraction and result interpretation during the conduction of the simulation study in a realistic environment. On the example of the material flow in a rework area of an assembly line for engines it is illustrated that PSIMA supports training on the job in industrial environments. On the example of the material flow in gas turbine manufacturing perspectives are given, how the learning environment can be further developed towards a mobile App for the analysis of material flows. The contribution of the learning environment to a high-quality planning is discussed based on the material flow technical applications.

Inhaltsverzeichnis

1	Einleitung und Aufbau	1
2	Stand der Forschung und Technik	8
2.1	Planung von Materialflusssystemen	8
2.1.1	Ganzheitliche Produktionssysteme	11
2.1.2	Cyber-Physical Production Systems und Multiagentensysteme	11
2.2	Simulation von Materialflusssystemen	12
2.2.1	Modellierung und Simulation	13
2.2.2	Simulationswerkzeuge	14
2.2.3	Multiagentensimulation	16
2.2.4	Durchführung von Simulationsstudien	18
2.3	Qualifizierung von Simulations-Anwendern	20
2.3.1	Wissensdomänen und Simulationsinstrumente	20
2.3.2	Konzepte zur Modellierung	23
2.3.2.1	Bausteine und der grafisch-interaktive Modell Aufbau	26
2.3.2.2	Programmiersprachen zur Implementierung von Simulationsmodellen	27
2.3.2.3	Potenzial der (teil-)automatischen Modellgenerierung	28
2.3.2.4	Animation und Potenzial von Simulation in der Cloud	29
2.4	Erhöhung der Lehr- und Lernproduktivität	30
2.4.1	Instruktionsdesign	31
2.4.2	Gamification und Serious Games	31
2.4.2.1	Spieler-Typen	34
2.4.2.2	Motivation und Flow	35
2.4.2.3	Spielentstehung	37
2.4.2.4	Gamification im Fabrikbetrieb	38
2.4.3	Lernfabriken und erfahrungsbasiertes Lernen	39
2.4.4	Visuelle blockbasierte Programmiersprache	41
2.5	Vorarbeiten zur Einarbeitung in ereignisdiskrete Simulation	43
3	Handlungsbedarf	51
3.1	Anforderungen	51
3.2	Bewertung und Zwischenfazit	53

4	Lernumgebung für ereignisdiskrete Simulation	58
4.1	Einarbeitung in Simulationsinstrument	59
4.1.1	Cloud-IT-Architektur	59
4.1.2	Lernförderliche Elemente	62
4.1.2.1	Inkrementell komplizierter werdende Simulationsmodelle	62
4.1.2.2	Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen	67
4.2	Einarbeitung in die Durchführung von Simulationsstudien	70
4.3	Verlauf des Spielens und Lernens	72
5	Prototypische Implementierung und Erprobung	74
5.1	Gamifizierte App PSIMA	74
5.1.1	Verwendete Software-Werkzeuge und Cloud-IT-Architektur	74
5.1.2	Aufbau und Funktionen	77
5.1.3	Inkrementell komplizierter werdende Simulationsmodelle	81
5.1.4	Multiagentensimulation	84
5.1.5	(Teil-)automatische Modellgenerierung im Simulationsinstrument	87
5.1.6	Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen	88
5.1.7	Animation	91
5.1.8	Erprobung und Chancen	93
5.2	Beispiele materialflusstechnischer Anwendungen	101
5.2.1	Universitäres Versuchsfeld	101
5.2.2	Gasturbinen	108
5.2.3	Motorradantriebe	110
5.2.4	Planungsqualität	113
6	Zusammenfassung	115
	Literaturverzeichnis	118
	Anhang	138
A	Marktgängige Simulationsinstrumente	138
B	Skripte und Tabellen Umsetzung Cloud-IT-Architektur	141
C	Implementierung von PSIMA	144
D	Kurzbeschreibung der exemplarischen Level von PSIMA	149
E	Abzeichen in PSIMA	154
F	Skript Umsetzung Multiagentensimulation	157
G	Skripte, Beispiel und Tabellen Umsetzung VBPI	159
H	Skripte und Tabellen Umsetzung Animation	164
I	Ablauf der Montage in der Lernfabrik	168
J	Veröffentlichungen des Autors und Beiträge von Studierenden	170
	Glossar	172

Abbildungsverzeichnis

1.1	Investitions- und Zeitersparnis durch Simulation	2
1.2	Angestrebter Pfad der Qualifikation	5
1.3	Aufbau der Arbeit	7
2.1	Klassifikation von Simulationsmethoden	14
2.2	Simulationszyklus: Durchführung einer Simulationsstudie	19
2.3	Einordnung von Simulationswerkzeugen unter den Aspekten Flexibilität, Allgemeingültigkeit und Anwendungsbezug	21
2.4	Abgrenzung „Gaming“ vs. „Playing“	32
2.5	Flow-Theorie	36
2.6	Erfahrungsbasierter Lernzyklus	40
2.7	Beispiel der visuellen blockbasierten Programmiersprache Scratch	42
4.1	Überblick über die Bestandteile des Konzepts	58
4.2	Cloud-IT-Architektur	60
4.3	Thematische Untergliederung in sechs Levelpacks	64
4.4	VBPIS im Kontext der Cloud-IT-Architektur	69
4.5	Vorgehen zum EEDS	71
5.1	Cloud-IT-Architektur der prototypischen Implementierung	76
5.2	Bildschirmfotos der prototypischen App Teil 1	79
5.3	Bildschirmfotos der prototypischen App Teil 2	80
5.4	Status der Levelbearbeitung	84
5.5	Hierarchie der Implementierung der Multiagentensimulation im Simulationsinstrument	85
5.6	Beispiele für Multiagentensimulation unter Nutzung von anwenderdefinierten Bausteinen	86
5.7	Prinzip der Modellierung von Agenten in PSIMA	87
5.8	Bildschirmfoto der VBPIS	88
5.9	Bildschirmfoto der Benutzungsoberfläche für die VBPIS	89
5.10	Bildschirmfoto Simulationsmodell und Animation in Game-Engine	92
5.11	Statistische Verteilung der Anzahl durch aktive Nutzer erzeugter Simulationsmodelle	94
5.12	Eingereichte Simulationsmodelle je Level	95
5.13	Erfolgreich abgeschlossene Level	96
5.14	Summe aktiver Probanden je Stunde kumuliert über alle Tage des Studienzeitraums	97
5.15	Durchschnittliche Nutzungsintensität je Stunde pro aktivem Nutzer	98
5.16	Kumulierte Anzahl eingereichter Simulationsmodelle nach Wochentagen	98

5.17 U-Layout der Lernfabrik an der VGU und Produkte	102
5.18 Beispiel für die Verbesserung des Layouts der Lernfabrik	105
5.19 Materialflusssystem errichtet mit Lego als Alternative zur Lernfabrik beim EEDS	107
5.20 Anwendungsbeispiel Gasturbinenbau: Übersicht über die Produkte	109
5.21 Simulationsszenarien Gasturbinenbau	110
5.22 Prinzipskizze Layout Motormontage	111
5.23 Prinzipskizze Layout Nacharbeitsbereich Motormontage	112
5.24 Nacharbeitsbereich Motormontage modelliert in PSIMA	113
5.25 Gegenüberstellung Ablauf Simulationsstudie mit und ohne Lernumgebung	114
B.1 Datenbankmodell der Cloud-IT-Architektur	142
B.2 Ablauf der (teil-)automatischen Modellgenerierung im Simulationsinstrument	143
C.1 Menüführung der App PSIMA	145
C.2 Bildschirmfotos der prototypischen App: Vertiefende Darstellung	146
E.1 Abzeichen in PSIMA	156
G.1 VBPIs Minimalbeispiel Compiler: Block-Editor in der App PSIMA	160
H.1 Bildschirmfoto der Tabelle zur Animation generiert durch das Simulationsinstrument	164
H.2 Bildschirmfoto der stereoskopischen dreidimensionalen Animation	165

Tabellenverzeichnis

2.1	Ebenen der Fabrik mit beispielhaften Planungsaufgaben und geläufigen Bezeichnungen von Simulationsanwendungen	10
2.2	ADDIE-Modell	31
3.1	Kapitelverweise für Anforderungen	53
3.2	Handlungsbedarf	55
4.1	Lernziele mit VBPIS im Kontext der Levelpacks	68
4.2	Lernumgebung zur Einarbeitung in Simulation: Überblick über den Spiel- und Lernverlauf	73
5.1	Beschreibung von Level 1-1	81
5.2	Beschreibung von Level 2-5	82
5.3	Beschreibung von Level 4-4	82
5.4	Beschreibung von Level 5-1	83
5.5	VBPIS: In der Block-Bibliothek der App bereitgestellte Blöcke	90
5.6	Motormontage Ergebnisse Simulationsszenarien	112
A.1	Liste marktgängiger ereignisdiskreter Simulationsinstrumente	139
C.1	In PSIMA implementierte Objekte und Bausteine: Kurzbeschreibung der Eigenschaften .	147
C.2	In PSIMA implementierte Objekte und Bausteine: Symbole	148
D.1	PSIMA IKS Levelpack 1 - Materialfluss	150
D.2	PSIMA IKS Levelpack 2 - Informationsfluss	151
D.3	PSIMA IKS Levelpack 4 - Anwenderdefinierte Bausteine	152
D.4	PSIMA IKS Levelpack 5 - Story Modus	153
E.1	Übersicht über die Abzeichen in PSIMA	154
G.1	VBPIS Minimalbeispiel Compiler: Datenbank	161
G.2	Ausschnitt der generischen Struktur implementierter Blöcke zur Übersetzung der VBPIS in die Sprache SimTalk	163
I.1	Montageablauf in der Lernfabrik der VGU	168
I.2	Übersicht über den Ablauf der Übung in der Lernfabrik der VGU	169

Abkürzungsverzeichnis

ADDIE	Analyze, Design, Development, Implementation, Evaluation
App	Applikation
AR	Augmented Reality
ASIM	Arbeitsgemeinschaft Simulation
CIRP	Internationale Akademie für Produktionstechnik
CPPS	Cyber-Physical Production System
CPS	Cyber-Physical System
DES	Discrete-Event Simulation
EEDS	Erfahrungsbasiertes Erlernen der Durchführung von Simulationsstudien
EDU in SIM	Ausbildung im Modellaufbau und der Simulation
FTS	Fahrerloses Transportsystem
FTF	Fahrerloses Transportfahrzeug
FIFO	First In First Out
GPS	Ganzheitliche Produktionssysteme
GUI	Graphical User Interface
IKS	Inkrementell komplizierter werdende Simulationsmodelle
IT	Informationstechnik
KMU	Kleine und Mittlere Unternehmen
KPI	Key Performance Indicator
OP	Arbeitsstation
PSIMA	Production Simulation Application
SCM	Supply Chain Management
SFB	Sonderforschungsbereich
SIM in EDU	Simulation in/für Ausbildung
SQL	Structured Query Language
VBPIs	Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen
VDA	Verband der Automobilindustrie
VDI	Verein Deutscher Ingenieure e.V.
VGU	Vietnamesisch-Deutsche Universität
VR	Virtual Reality
Web	World Wide Web

1 Einleitung und Aufbau

Längst haben rechnergestützte Anwendungen Einzug in die Produktion gehalten und unterstützen Mitarbeiter bei Entscheidungen. Modelle, Methoden und Werkzeuge sind in der digitalen Fabrik durch ein durchgängiges Datenmanagement integriert [VDI-4499a, S.3]. Die Studie der Wirtschaftsprüfungsgesellschaft PricewaterhouseCoopers GmbH *Digital Factories 2020: Shaping the future of manufacturing* unterstreicht die Wichtigkeit der menschlichen Qualifikation im Kontext der digitalen Fabrik [Gei-17, S.34f.]. Im Jahr 2017 wurden 200 Entscheidungsträger in Unternehmen aus den Bereichen der Produktentwicklung, Produktion und Technologie befragt. Mit Blick auf die Digitalisierung in Fabriken erwarten 71% der Befragten ein hohes Investment in die Fortbildung ihrer Mitarbeiter. Dass ihre Mitarbeiter für innovative digitale Technologien noch nicht aufgeschlossen seien, führen 49% an. Der Auffassung, dass die Ausbildung in Universitäten nicht adäquat ist, um potenzielle Mitarbeiter hinsichtlich der digitalen Fabrik zu qualifizieren, sind 44%. Simulationen sind ein Bestandteil der digitalen Fabrik [VDI-4499a, S.3]. Komplexe Zusammenhänge lassen sich mit Simulation in kurzer Zeit darstellen und analysieren [Bis-15, S.268f.].

Aufgrund von zunehmend dezentral getroffenen Entscheidungen wird ein neuer „Komplexitätshub“ für die Planung und Steuerung von Produktionssystemen zu erwarten sein [May-17]. Ein Produktionssystem umschreibt „das Zusammenspiel aus Technik, Organisation und Mensch in der Produktion“ [Dom-15, S.3]. Materialflusssysteme sind ein Bestandteil von Produktionssystemen [Cag-14]. Transformationsprozesse von Gütern hinsichtlich Zeit, Ort, Menge, Zusammensetzung und Qualität durch Personal, Informationen und Produktionsmittel stehen im Fokus [Sch-08a]. Dezentrale Steuerungen, die häufig mit Industrie 4.0 und insbesondere mit den Cyber-Physical Production Systems (CPPS) in Verbindung gebracht werden, erhöhen die Komplexität der Planung von Materialflusssystemen [Kag-13]. Die Bedeutung der ereignisdiskreten Simulation zur Absicherung der Planungen steigt vor dem Hintergrund der Entwicklung von CPPS und der damit verbundenen Komplexität. Flexible ereignisdiskrete Simulationsinstrumente können zur Entscheidungsunterstützung bei der Planung von Materialflusssystemen¹ eingesetzt werden [Hom-18, S.371f.] [Uyg-15, S.116] [VDI-2689, S.13]. Mit ereignisdiskreten Simulationsinstrumenten lassen sich mit der Multiagentensimulation auch dezentrale Steuerungen autonomer logistischer Gegenstände des Materialflusses modellieren und simulieren [Bis-15, S.315] [Bra-18, S.132f.] [Fre-04] [Sch-06].

¹Die Aufgaben von Simulation bei der Neu- und Umplanung von Materialflusssystemen sind im Kapitel 2.1 beschrieben. Die begrifflichen Grundlagen der ereignisdiskreten Simulation finden sich im Kapitel 2.2.1. Ein Simulationsinstrument ist ein Werkzeug zur Simulation, das sich an Endanwender richtet [Wen-08b, S.82].

Simulation hat das Potenzial, Menschen bei Entscheidungen durch die Analyse von mehr Handlungsalternativen in kürzerer Zeit zu unterstützen. In diesem Kontext ist in der Abbildung 1.1 qualitativ die Verringerung von Investitionen durch den planungsbegleitenden Einsatz von Simulation aufgezeigt [Gut-17, S.47f.].

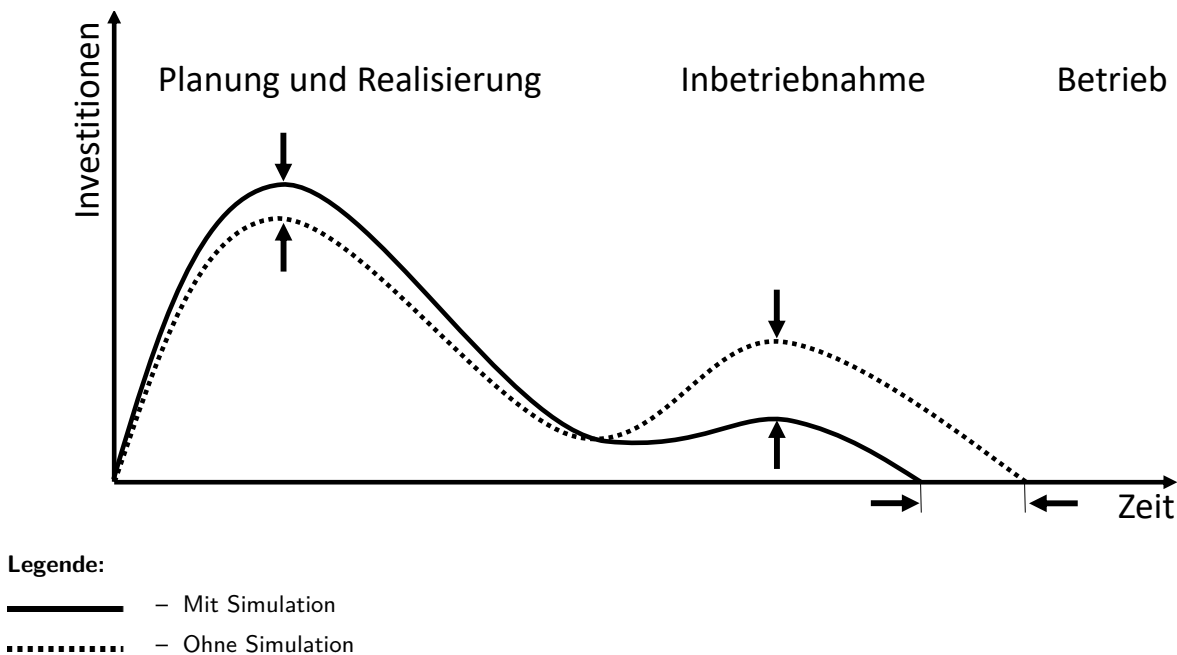


Abbildung 1.1: Investitions- und Zeitersparnis durch Simulation nach [Gut-17, S.48]

Zunächst sind höhere Investitionen in Kauf zu nehmen. Diese ergeben sich beispielsweise aus Lizenzgebühren für Software zur Simulation und für zusätzliches Personal zur Durchführung von Simulationsstudien. Dem steht eine Ersparnis durch frühzeitig erkannte Fehler gegenüber. Diese Fehler müssen folglich auch nicht in späteren Phasen des Projekts aufwändig behoben werden. Eine exakte Quantifizierung der Ersparnis durch den Einsatz von Simulation ist kaum möglich, da je Projekt nur eine der beiden Kurven ermittelt werden kann [Gut-17, S.47f.]. In der Richtlinie des VEREINS DEUTSCHER INGENIEURE E.V. (VDI) 3633 sind am Beispiel der Automobilindustrie Erfahrungswerte dokumentiert. So wird in diesem Industriezweig etwa 20% des Investitionsvolumens in Projekten vergeben, die mit sogenannten Simulationsanwendungen bearbeitet werden. Die erzielten Einsparungen sind mit zwischen zwei und vier Prozent des Investitionsaufwands beziffert. Circa 0,5% des Volumens ist als Kosten für die Vorbereitung, Ausführung und Bewertung von Simulationsstudien benannt [VDI-3633a, S.39ff.]. Kurz zusammengefasst, kann durch Simulation die Qualität der Planung von Materialflusssystemen erhöht werden. Ersparnisse ergeben sich aus der schnelleren Betriebsbereitschaft des Materialflusssystems sowie geringeren Investitionen in Nachbesserungen [Gut-17, S.48].

Die DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN E.V. sieht eine große Herausforderung darin, „der breiten Masse der Ingenieure die Potenziale von Modellen zu vermitteln und ihnen Methoden und Werkzeuge an die Hand zu geben, wie sie Systeme aus der realen Welt mittels geeigneter Modelle in der virtuellen Welt abbilden“ [Kag-13, S.47]. Simulation und die damit verbundene Modellierung erfordern qualifizierte Experten [Kag-13, S.48]. Die in dieser Arbeit betrachteten Handlungsfelder umfassen

die Einarbeitung in *Simulationsinstrumente* und *Simulationsstudien*. Marktgängige *Simulationsinstrumente* sind umfangreiche Expertensysteme [Bis-15, S.269] [Mos-13]. In der Literatur besteht Konsens darüber, dass es einer Ausbildung oder Schulung hinsichtlich der Anwendung von ereignisdiskreten Simulationsinstrumenten bedarf [Byr-15] [Mos-13] [Pad-14] [VDI-2689, S.13] [VDI-3633a, S.32]. Die Vorteile der Ersparnis von Zeit und Investitionen sind kaum für den Anwender erkennbar, wenn dieser nicht dazu qualifiziert ist, ein Simulationsinstrument anzuwenden [Byr-15]. Mit Blick auf die Durchführung von *Simulationsstudien* steht die Erfahrung des Simulationsanwenders beim Aufbau von Simulationsmodellen und der Ergebnisinterpretation im Vordergrund [Ban-10b, S.23f.] [Sha-98]. Der Erfolg einer Simulationsstudie ist im Wesentlichen von der Erfahrung des Durchführenden abhängig [May-12] [May-15a]. Für beide Handlungsfelder relevante Qualifikationen werden oft während des Studiums des Maschinenbaus, der Produktionstechnik und des Wirtschaftsingenieurwesens an Hochschulen erworben. Unternehmen investieren oft nur zurückhaltend in die Qualifizierung ihrer Mitarbeiter zur Durchführung von Simulationsstudien und zur Anwendung von ereignisdiskreten Simulationsinstrumenten [Ola-14]. Eine hohe Qualifikation von Hochschulabsolventinnen und -absolventen in der Durchführung von Simulationsstudien und der Anwendung von Simulationsinstrumenten scheint folglich einen großen Beitrag zur Verbreitung von ereignisdiskreter Simulation zu leisten. Damit verbunden lassen sich die Potenziale von ereignisdiskreter Simulation besser ausschöpfen.

Mit mobilen und spielerischen Ansätzen sowie Lernfabriken kann eine verbesserte Ausbildung in beiden Handlungsfeldern erreicht werden [Abe-17]. Mobile und spielerische Ansätze können insbesondere für junge Lernende ein geeignetes Mittel zur Instruktion sein. Diese Ansätze ermöglichen eine Interaktion von Spielern in einer Gruppe [McG-11, S.30ff.]. Spiele fördern die Zusammenarbeit und können gleichzeitig Konkurrenz zwischen Spielern schaffen [McG-11, S.228]. Häufig werden mit spielerischen Ansätzen Applikationen (Apps) in Verbindung gebracht. Eine App ist eine Anwendung auf einem neuartigen Endgerät mit Berührungsbildschirm [Grö-13]. Anwendungen, die auf World Wide Web (Web)-Browsern basieren, sind darin eingeschlossen. Im Jahr 2017 nutzten in Deutschland 95% der 14-29-jährigen und 97% der 30-49-jährigen ein Smartphone [Bit-17]. Nutzer können Apps von virtuellen Stores herunterladen. Im Jahr 2016 wurden weltweit 181,5 Milliarden Apps aus den Stores durch Benutzer heruntergeladen [Sta-16]. Das absteigende Ranking der Anzahl verfügbarer Apps nach Kategorie ist am Beispiel des App-Stores von Apple im Februar 2019 wie folgt sortiert [Poc-19]:

- Spiele: 820.307 Apps = 24,99%
- Business: 323.556 Apps = 9,79%
- Bildung: 282.254 Apps = 8,49%

Nach Angaben des VERBANDS DER DEUTSCHEN GAMES-BRANCHE spielen immer mehr Menschen auf Smartphones und Tablets [gam-18]. So spielten 18,2 Millionen Deutsche im Jahr 2018 auf Smartphones und 11,5 Millionen auf Tablet-PCs. Spiele auf mobilen Endgeräten sind in Deutschland am beliebtesten. Der Umsatz für Spiele-Apps stieg in Deutschland zwischen 2016 und 2017 um 21% auf 497 Millionen € [gam-18]. Spiele haben ein Ziel, das es unter Beachtung von Regeln zu erreichen gilt. Spieler erhalten durch sogenannte Spielelemente ein Feedback [McG-11, S.21]. Spielelemente sind Punkte, Level oder Abzeichen. Punkte dokumentieren sichtbar für eine Gruppe an Spielern den Fortschritt des einzelnen Spielers. Ein Level repräsentiert eine Lerneinheit, die eine durch den Lernenden zu lösende

Aufgabe umfasst. Abzeichen machen Erfolge im Spiel für den Spieler und Mitspieler sichtbar. Kurz zusammengefasst lässt sich neben der zunehmenden Relevanz von Apps im Alltag festhalten, dass Spiele-Apps das Potenzial haben, das menschliche Handeln zu beeinflussen. Insbesondere junge Lernende nutzen solche Apps regelmäßig.

Lernfabriken werden in der aktuellen wissenschaftlichen und praktischen Diskussion als ein vielversprechendes Werkzeug zur Vermittlung von Qualifikationen in Forschung und Lehre diskutiert. Lernfabriken sind ein Ort, an dem Studierende in einer physischen und/oder virtuellen Lernumgebung experimentell und problembasiert lernen. Realitätsnahe Prozesse, ein veränderlicher Aufbau, die Herstellung eines physischen Produkts und ein didaktisches Konzept charakterisieren eine Lernfabrik [Abe-15]. Lernfabriken haben das Potenzial, die Durchführung von Simulationsstudien realitätsnah für Lernende erlebbar zu machen. Für Lernende eröffnet dies die Chance, Erfahrungen im Aufbau von Simulationsmodellen und in der Ergebnisinterpretation in einer sicheren Umgebung zu sammeln. Gefahren wie Beschädigungen oder Fehlplanungen industrieller Materialflusssysteme und damit verbundene Einbußen in der Produktivität aufgrund mangelnder Qualifikation in der Durchführung von Simulationsstudien lassen sich so vermindern. In der Arbeit stehen neuartige lernförderliche Ansätze wie Spiele-Apps und Lernfabriken zur Erhöhung der Qualifikation von Studierenden für ereignisdiskrete Simulation im Fokus. Der Anwendungsbereich ist auf die Neu- und Umplanung von Materialflusssystemen ausgerichtet.

Die vorliegende Arbeit soll einen Beitrag dazu leisten, zeitdiskret-ereignisorientierte Simulation, oder kurz ereignisdiskrete Simulation, zur Entscheidungsunterstützung bei der Neu- und Umplanung von Materialflusssystemen zu verbreiten. Das Ziel besteht darin, die Aus- und Weiterbildung von Ingenieuren in ihrer Fähigkeit, Materialflusssysteme zu planen, zu verbessern. Eine mobile und spielerische Lernumgebung soll konzipiert und prototypisch umgesetzt werden, die es Anfängern ermöglicht, sich in ein Simulationsinstrument zur Durchführung von Simulationsstudien einzuarbeiten. Es leiten sich folgende Teil-Fragestellungen ab:

- Welche gängigen ereignisdiskreten Simulationsinstrumente existieren im Anwendungsbereich der Planung von Materialflusssystemen und welche Eigenschaften hinsichtlich des Modellierungskonzepts haben diese?
- Wie werden Simulationsstudien durchgeführt?
- Wie werden digitale Spiele konzipiert?
- Gibt es bereits Ansätze, die dazu geeignet sind, Studierende in Simulationsinstrumente oder die Durchführung von Simulationsstudien einzuarbeiten?
- Wie kann eine Lernumgebung zur Einarbeitung in ein Simulationsinstrument und die Durchführung von Simulationsstudien konzipiert und realisiert sein?

Die Lernumgebung richtet sich an Studierende mit geringer Erfahrung in der Durchführung von Simulationsstudien und der Anwendung eines ereignisdiskreten Simulationsinstruments. Experten in der Simulationstechnik stehen nicht im Fokus der Entwicklungen. Der angestrebte Pfad der Qualifikation von Quadrant IV zu I ist in Abbildung 1.2 skizziert.

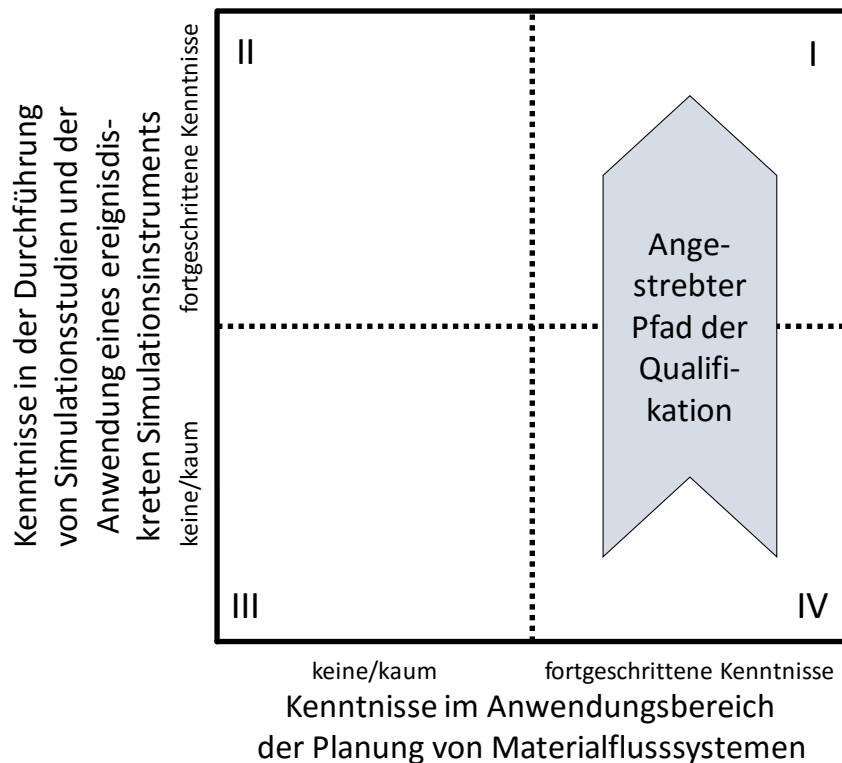


Abbildung 1.2: Angestrebter Pfad der Qualifikation

Der Aufbau der Arbeit ist in Abbildung 1.3 illustriert. Im Kapitel 2 ist der aktuelle Stand der Forschung und Technik beschrieben. Zunächst ist die begriffliche Basis für den Anwendungsbereich der Neu- und Umplanung von Materialflusssystemen im Abschnitt 2.1 ausgeführt. Im Kapitel 2.2 sind die relevanten Begrifflichkeiten der Modellierung und Simulation von Materialflusssystemen erarbeitet. Simulationswerkzeuge, Multiagentensimulation sowie der Ablauf einer Simulationsstudie sind hier diskutiert. Im Kapitel 2.3 ist die Qualifizierung von Simulations-Anwendern ausgeführt. Simulationswerkzeuge werden mit Wissensdomänen in Beziehung gesetzt. Eine Eingrenzung auf die attraktive Klasse der Simulationsinstrumente für spezifische Anwendungsbereiche wird vorgenommen. Gängige Arten der Modellbeschreibung in Simulationsinstrumenten und die Analyse von Potenzialen für eine attraktive Einarbeitung in diese Simulationsinstrumente schließen das Kapitel ab. Im Kapitel 2.4 sind mit Blick auf die Qualifizierung der Simulationsanwender die Potenziale mobiler spielerischer Ansätze, von Lernfabriken und visueller blockbasierter Programmiersprache herausgearbeitet. Relevante Ansätze aus der Praxis und Forschung sind im Kapitel 2.5 dokumentiert.

In dem Kapitel 3 sind zunächst Anforderungen an das Konzept auf Basis der Erkenntnisse aus dem Stand von Forschung und Technik herausgearbeitet. Diese sind mit den im Kapitel 2.5 identifizierten Ansätzen aus Praxis und Forschung abgeglichen. Hieraus leitet sich der Bedarf für

- ein Vorgehen für das erfahrungsbasierte Erlernen der Durchführung von Simulationsstudien (EEDS) unter Nutzung von Lernfabriken,

- Inkrementell komplizierter werdende Simulationsmodelle (IKS) unter Nutzung von Spielelementen sowie
- eine visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen (VBPIS) ab.

Im Abschnitt 4 ist die Lernumgebung konzipiert. Hinsichtlich des Handlungsfelds der Simulationsinstrumente werden im Kapitel 4.1 IKS und VBPIS entworfen. Diese sind in Form einer App in eine Cloud-IT-Architektur eingebettet. Hinsichtlich des Handlungsfelds der Simulationsstudie ist im Kapitel 4.2 das Vorgehen für das EEDS unter Nutzung von Lernfabriken erarbeitet. Im abschließenden Unterkapitel 4.3 sind die Entwicklungen im Sinne eines Spiel- und Lernverlaufs zusammengefasst.

Im Kapitel 5 ist zunächst im Abschnitt 5.1 die prototypische Implementierung der Cloud-IT-Architektur beschrieben. Die Cloud-IT-Architektur ist die informationstechnische Grundlage für die Umsetzung der mobilen App Production Simulation Application (PSIMA). Die App PSIMA integriert die IKS und VBPIS. Die Erprobung von PSIMA mit einer Gruppe Studierender schließt das Kapitel 5.1 ab. Im Kapitel 5.2 sind Beispiele für materialflusstechnische Anwendungen aufgezeigt. Im universitären Versuchsfeld ist die Erprobung des Vorgehens für das EEDS in einer Lernfabrik mit Studierenden beschrieben. Orientiert an den industriellen Beispielen der Fertigung von Gasturbinen und der Montage von Motorradantrieben sind Potenziale für eine hohe Planungsqualität von Materialflusssystemen durch die Entwicklungen in dieser Arbeit abgeleitet.

Schließlich sind die wesentlichen Erkenntnisse aus den vorangegangenen Kapiteln im Abschnitt 6 zusammengefasst.

Kapitel 2: Stand der Forschung und Technik

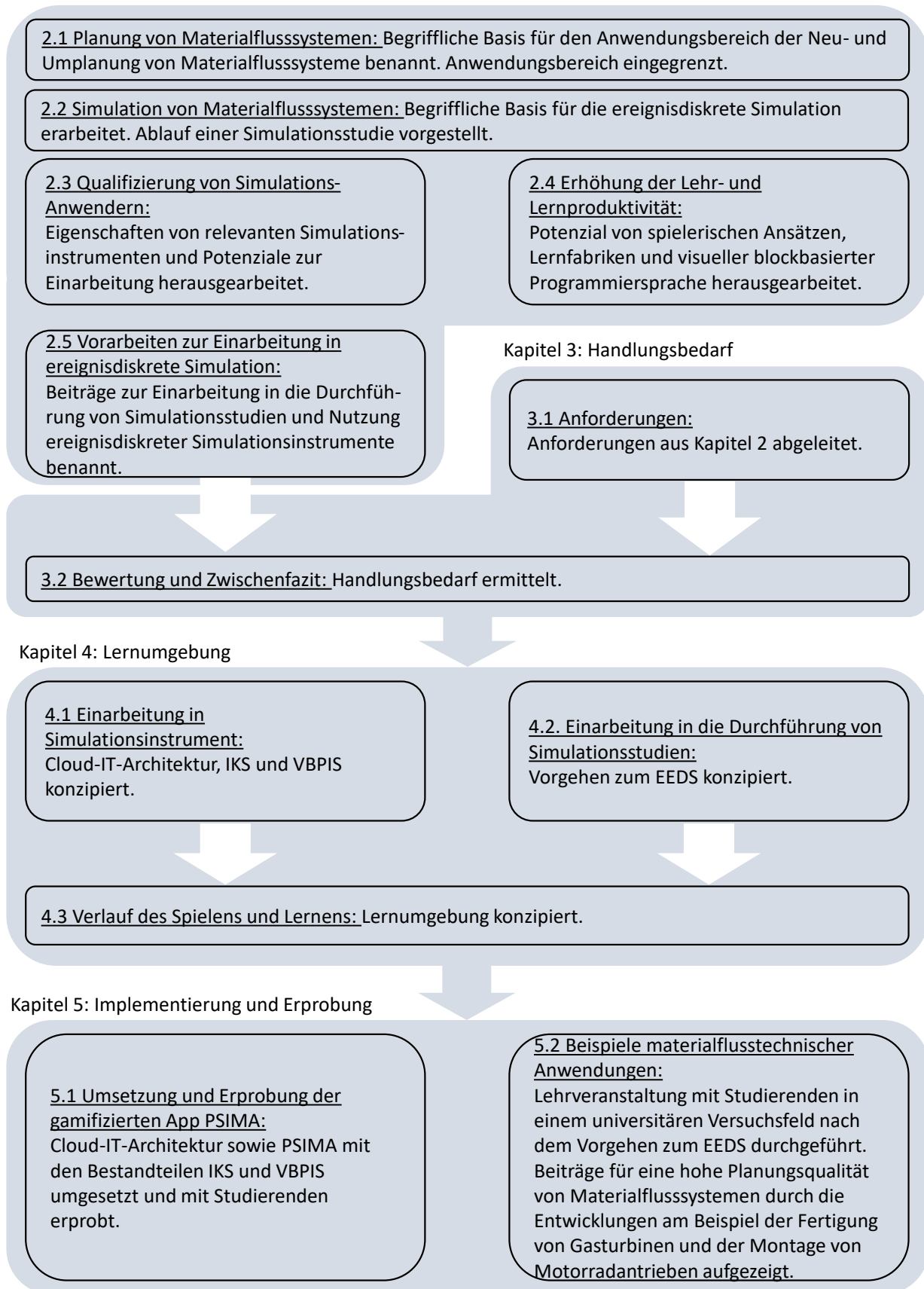


Abbildung 1.3: Aufbau der Arbeit

2 Stand der Forschung und Technik

In diesem Kapitel ist ein Überblick über den aktuellen Stand der Forschung und Technik dokumentiert. Zunächst sind im Kapitel 2.1 die Aufgaben bei der Neu- und Umplanung von Materialflusssystemen vor dem Hintergrund aktueller Diskussionen um Cyber-Physical Systems und Ganzheitliche Produktionssysteme spezifiziert. Im Kontext der Planung von Materialflusssystemen werden im darauffolgenden Kapitel 2.2 relevante Begriffe hinsichtlich der Modellierung und ereignisdiskreten Simulation definiert, der Aufbau von Simulationswerkzeugen beschrieben, die Relevanz von Multiagentensimulation dargestellt und die Durchführung von Simulationsstudien ausgeführt. Mit Blick auf die Qualifizierung von Simulations-Anwendern werden im Abschnitt 2.3 Simulationswerkzeuge im Kontext von Wissensdomänen betrachtet, um attraktive Werkzeuge für die Planung von Materialflusssystemen zu identifizieren. Relevante Modellierungskonzepte in etablierten Simulationsinstrumenten sowie Potenziale für die Qualifizierung von Simulations-Anwendern werden identifiziert. Chancen zur Einarbeitung in Simulationsinstrumente sind mit Blick auf spielerische Ansätze und didaktische Vorgehensweisen im Kapitel 2.4 ausgeführt. Schließlich sind im Abschnitt 2.5 die relevanten Vorarbeiten aus der Forschung und Praxis beschrieben, die einen Beitrag zur Einarbeitung in die Durchführung von Simulationsstudien und Anwendung ereignisdiskreter Simulationsinstrumente versprechen.

2.1 Planung von Materialflusssystemen

Der Begriff Materialfluss beschreibt die Bewegung von Gegenständen durch ein Produktionssystem [Din-14]. Es werden das Bearbeiten, das Prüfen, das Handhaben, das Fördern, der Aufenthalt, das Lagern, das Bilden von Ladeeinheiten, das Kommissionieren, das Montieren sowie das Be- und Entladen unterschieden [Sch-08a] [VDI-2689, S.2]. „Die Anordnung von mindestens zwei Gegenständen des Materialflusses, die im Rahmen eines Transformationsprozesses eine Veränderung des Systemzustandes von Gütern hinsichtlich Zeit, Ort, Menge, Zusammensetzung und Qualität ermöglichen, wird als Materialflusssystem bezeichnet“ [Sch-08a]. Die Gegenstände des Materialflusses umfassen nach SCHMIDT UND SCHNEIDER (2008) das Personal, die Informationen, die Produktionsmittel, die Materialflussmittel, die Informationsflussmittel, die Energie und die Infrastruktur. In einem Materialflusssystem sind der Materialfluss und der Informationsfluss untrennbar miteinander gekoppelt [VDI-2689, S.2].

Die Komplexität des Materialflusses ist nach LÖDDING (2016) von der Anzahl möglicher Vorgänger und Nachfolger eines Arbeitssystems sowie der Anzahl der Rückflüsse abhängig. Rückflüsse entstehen, wenn die Arbeitssysteme „nicht mehr in eine [...] Reihenfolge gebracht werden können, die alle Aufträge

in die gleiche Richtung durchlaufen“. Ein linearer Materialfluss, bei dem jedes Arbeitssystem einen Vorgänger und Nachfolger hat, weist die niedrigste Materialflusskomplexität auf. Diese Komplexität steigt mit der Anzahl möglicher Vorgänger und Nachfolger im Sinne von Verzweigungen und Rückflüssen [Löd-16, S.133f.]. Komplexitätsfördernd wirken Produktvarianten sowie stochastische Einflüsse. Da Materialflusssysteme in einer sich wandelnden und unsicheren Umwelt betrieben werden, sind sie als komplexe Systeme zu bezeichnen [EIM-12]. Dynamisches Verhalten und stochastische Einflüsse prägen diese Systeme.

Gemäß der Richtlinie VDI 5200 lassen sich fünf Ebenen der Planung in abnehmender Aggregation unterscheiden [VDI-5200, S.7]:

Produktionsnetz: Ein Produktionsnetz ist ein Verbund von Standorten eines oder mehrerer Unternehmen. Solch ein Netz kann lokal, regional, überregional oder international aufgestellt sein.

Werk: Ein Werk ist ein örtlich-räumlich geschlossener Produktionsbereich mit definierten Produktionsaufgaben an einem Standort. Ein Werk kann aus mehreren Gebäuden bestehen, die durch ein Wegenetz miteinander verbunden sind. Außenanlagen und eine Anbindung an die werksexterne Infrastruktur sind typisch.

Gebäude: Ein Gebäude ist ein architektonisch-räumlich geschlossener Produktionsbereich. Im engeren Sinne wird hierunter auch die Fabrik verstanden. Ein Gebäude kann mehrere Segmente umfassen.

Segment: Ein Segment ist ein geschlossener Produktionsbereich, der aus mehreren Arbeitsplätzen besteht.

Arbeitsplatz: Der Arbeitsplatz ist als elementarer Produktionsbereich mit Betriebsmitteln anzusehen.

In dieser Arbeit sind die Planungsfälle Neu- und Umplanung betrachtet [VDI-5200, S.4]. Die Neuplanung wird oft auch als Grüne-Wiese-Planung bezeichnet. Als bauliche Restriktionen sind lediglich das Grundstück sowie die vorhandene Infrastruktur zu beachten. Bei der Umplanung ist von einer bestehenden Fabrik auszugehen. Es gilt, bei der Planung Restriktionen eines eventuell laufenden Fabrikbetriebs zu berücksichtigen. Darüber hinaus sind vorhandene Produktionsfaktoren zu beachten. Typische Anlässe für die Analyse von Materialflusssystemen sind neue Produkte, Stückzahländerungen, geplante Investitionen, hohe Lagerbestände, unübersichtliche Transport- und Lagerverhältnisse oder Kapazitätsprobleme [VDI-2689, S.3].

Bei der Neu- und Umplanung von Materialflusssystemen steht meist die Maximierung der Wirtschaftlichkeit im Fokus. Entscheidungen bei der Planung müssen im Spannungsfeld von Durchlaufzeitminimierung, Terminabweichungsminimierung, Auslastungsmaximierung und Bestandsminimierung getroffen werden [VDI-3633a, S.9]. Grundlegende Gestaltungsalternativen für Materialflusssysteme ergeben sich aus dem Produktionsprinzip und den Fertigungsarten [Löd-16, S.123ff.]. Das Werkbank-, Baustellen-, Werkstätten-, Insel- oder Fließprinzip sowie die Fertigungsarten Einmalfertigung, Einzel- und Kleinserienfertigung, Serien- und Massenfertigung bestimmen maßgeblich den Materialfluss, der oft in Losen oder

als One Piece Flow organisiert ist. Tabelle 2.1 gibt einen Überblick über beispielhafte Planungsaufgaben mit Blick auf Materialflusssysteme der jeweiligen Planungsebene.

In dieser Arbeit werden zum besseren Verständnis überwiegend etablierte Gestaltungsprinzipien Ganzheitlicher Produktionssysteme zur Neu- und Umplanung von Materialflusssystemen fokussiert. Bei Ganzheitlichen Produktionssystemen handelt es sich um ein „unternehmensspezifisches, methodisches Regelwerk zur umfassenden und durchgängigen Gestaltung der Unternehmensprozesse“ [VDI-2870a, S.5].

Tabelle 2.1: Ebenen der Fabrik mit beispielhaften Planungsaufgaben und geläufigen Bezeichnungen von Simulationsanwendungen nach [Wie-14, S.139ff.] und [VDI-5200, S.7]

Ebene	Planungsaufgaben (Bsp.)	Bezeichnungen von ereignisdiskreten Simulationsanwendungen (Bsp.)
Produktionsnetz	Wahl des Verkehrsträgers	Supply Chain Simulation
	Erhöhung der Widerstandsfähigkeit der Supply Chain	Geschäftsprozesssimulation
	Bestimmung des Standorts von Werken und Lagern	Multiagentensimulation
	Gestaltung des Informationsflusses	
	Bestimmung des Kundenentkopplungspunkts	
	Gestaltung von Beschaffungs- und Distributionsnetzen	
Werk	Gestaltung des Layouts	Materialflusssimulation
	Gestaltung des Wertstroms	Geschäftsprozesssimulation
	Wahl des Produktionsprinzips	Simulationsbasierte Wertstromanalyse
	Gestaltung des Informationsflusses	Ablaufsimulation
	Wahl guter Steuerungsstrategien	Verkehrsflusssimulation [VDA-4810]
	Ermittlung von Störungsstrategien	Belieferungssimulation [VDA-4810]
	Gestaltung des Anlieferprinzips	Werkssimulation [VDA-4810]
	Kapazitätsabschätzungen	Multiagentensimulation
	Entscheidungsunterstützung für Eigenfertigung oder Fremdbezug	
Gebäude	Gestaltung des Layouts	Materialflusssimulation
	Gestaltung des Wertstroms	Simulationsbasierte Wertstromanalyse
	Gestaltung des Informationsflusses	Ablaufsimulation
	Ermittlung des Platzbedarfs	
Segment	Gestaltung des Layouts	Materialflusssimulation
	Gestaltung des Wertstroms	Ablaufsimulation
	Gestaltung des Informationsflusses	FTS-Simulation [VDI-2710]
	Bestimmung des Personaleinsatzplans	
	Ermittlung des Qualifikationsbedarfs	
	Auslastungserhöhung	
	Ermittlung von Störungsstrategien	
	Kapazitätsabschätzungen	
	Festlegung von Bestandsgrenzen	

Legende:

FTS – Fahrerloses Transportsystem

2.1.1 Ganzheitliche Produktionssysteme

Bei der Planung von Materialflusssystemen können Gestaltungsprinzipien der Ganzheitlichen Produktionssysteme zur Anwendung kommen. Diese Gestaltungsprinzipien sind umfangreich in der Richtlinie VDI 2870 beschrieben. Relevant für diese Arbeit sind im Kontext der Materialflusssysteme das *Fließprinzip* und das *Pull-Prinzip* [VDI-2870a]. Unter dem Fließprinzip sind Methoden zusammengefasst, die „einen schnellen, durchgängigen und turbulenzarmen Fluss von Materialien und Informationen“ ermöglichen sollen. Beim Pull-Prinzip wird eine Steuerung der Materialversorgung angestrebt, die sich aus dem Kundenbedarf ergibt. Bildlich formuliert zieht der Kunde an der Kette der Materialversorgung. Im Folgenden werden exemplarische Methoden kurz beschrieben, die sich bei der Planung von Materialflusssystemen nutzen lassen.

Beim *One Piece Flow* handelt es sich um eine Methode, die dem Fließprinzip zugeordnet ist [VDI-2870b, S.71f.]. Das Werkstück wird im Sinne von Losgröße Eins nach einem Bearbeitungsschritt auf einer Fertigungs- oder Montagestation direkt an die nachfolgende Fertigungs- oder Montagestation übergeben. Die Fertigungs- und Montagestationen sind in der Reihenfolge ihrer Bearbeitung angeordnet. Produkte folgen beim One Piece Flow, bedingt durch die starre Verkettung von Fertigungs- oder Montagestationen, dem First In First Out (FIFO)-Prinzip [Dro-15].

Die Methode *U-Layout* ist ebenso dem Fließprinzip zugeteilt [VDI-2870b, S.80f.]. Der Name der Methode ergibt sich aus den in der Draufsicht U-förmig angeordneten Fertigungs- oder Montagestationen. Der Anfang und das Ende des Materialflusses liegen sich gegenüber. Mitarbeiter arbeiten innerhalb der U-Form. Die Materialversorgung erfolgt in der Regel von außen.

Kanban zählt zum Pull-Prinzip [VDI-2870b, S.85f.]. Es handelt sich um einen selbststeuernden geschlossenen Regelkreis, bestehend aus einem Verbraucher von Teilen und dem Lieferanten. Mit Kanban Karten oder durch Informationstechnik (IT)-gestützte Verfahren werden Informationen über den Verbrauch an den Lieferanten weitergegeben. Der Lieferant produziert auf Basis der Information ein Los der benötigten Materialien und stellt dem Verbraucher diese in einem Supermarkt zur Verfügung.

Der *Supermarkt* dient der Entkopplung von Fertigungs- oder Montageprozessen durch Unterbrechung des Materialflusses zwischen Pull- und Push-Bereichen oder Pull- und Pull-Bereichen [VDI-2870b, S.95f.]. Es handelt sich um einen Puffer, der so auszulegen ist, dass nachfolgende Fertigungs- oder Montagebereiche ausreichend Teile zur Verfügung haben. Der sogenannte Kundenentkopplungspunkt gibt an, „ab wann aus einem kundenauftragsneutralen Fertigungsauftrag ein kundenauftragsspezifischer Fertigungsauftrag wird“ [Sch-08b, S.323ff.].

2.1.2 Cyber-Physical Production Systems und Multiagentensysteme

In der Diskussion um Industrie 4.0 werden häufig die sogenannten Cyber-Physical Systems (CPS) aufgegriffen [Kag-13] [May-17] [Spa-13]. „CPS umfassen eingebettete Systeme, Produktions-, Logistik-,

Engineering-, Koordinations- und Managementprozesse sowie Internetdienste, die mittels Sensoren unmittelbar physikalische Daten erfassen und mittels Aktoren auf physikalische Vorgänge einwirken, mittels digitaler Netze untereinander verbunden sind, weltweit verfügbare Daten und Dienste nutzen und über multimodale Mensch-Maschine-Schnittstellen verfügen“ [Kag-13, S.84]. Konkreter mit Blick auf die Produktion sind die CPPS beschrieben. CPPS fokussieren die Anwendung von CPS in der Industrie und befähigen „zur durchgängigen Betrachtung von Produkt, Produktionsmittel und Produktionssystem unter Berücksichtigung sich ändernder und geänderter Prozesse“ [Kag-13, S.84]. Bei den CPPS können intelligente Produkte und Werkstückträger mit Maschinen und Benutzern in ihrer Umgebung kommunizieren und teilweise autonom den Prozess zur Fertigung steuern [Bis-15, S.235]. Dezentrale Steuerungen erfreuen sich in Diskussionen um Industrie 4.0 steigender Beliebtheit [Kag-13] [Hey-16, S.9ff.] [May-17].

Zur Beschreibung dezentraler Steuerungen autonomer logistischer Gegenstände des Materialflusses findet sich in der Literatur häufig ein Bezug zu Agenten wieder [Fre-04] [Sch-06] [Bis-15, S.315] [Bra-18, S.132f.]. Ein Agent ist als autonome Einheit zu verstehen. Ein Agent kann seine Umgebung und darin befindliche Agenten wahrnehmen. Agenten haben neben dieser Fähigkeit Attribute und sind mit Regeln ausgestattet, die das Verhalten steuern [Law-15, S.694]. Als Agentensystem oder Multiagentensystem bezeichnet man eine Menge von interagierenden Agenten [Sch-06].

Systematisch wurde die Selbststeuerung logistischer Prozesse im Sonderforschungsbereich (SFB) 637 „Selbststeuerung logistischer Prozesse - Ein Paradigmenwechsel und seine Grenzen“ zwischen 2004 und 2012 untersucht [Fre-04] [DFG-19]. Die Selbststeuerung des Flusses von Material, das Management selbststeuernder logistischer Prozesse sowie die Realisierung der relevanten IT standen im Vordergrund [DFG-19]. Nach SCHOLZ-REITER UND HÖHNS (2006) finden sich Ansätze zur Beschreibung dezentraler Steuerungen durch Agenten in den Anwendungsbereichen der Produktionslogistik, Transportlogistik und dem Supply Chain Management (SCM) [Sch-06]. Beispiele sind die dezentrale Auftragsvergabe und Routenplanung selbstgesteuerter Fahrerloser Transportsysteme (FTS) [Sch-13], agentenbasierte Steuerungen für Elektrohängbahnsysteme [Chi-10], dezentral gesteuerte Kommissionierlager [Geh-10a] und agentenbasierte Staplerleitsysteme [Göh-10].

Nach der Einführung der Begrifflichkeiten zur Modellierung und Simulation werden die Multiagentensysteme im Kontext der ereignisdiskreten Simulation im Kapitel 2.2.3 erneut aufgegriffen.

2.2 Simulation von Materialflusssystemen

GUTENSWAGER ET AL. (2017) kommen in ihrer Literaturanalyse zum Ergebnis, dass ca. 80% der Anwendungsfälle von Simulation im Kontext der Produktion und Logistik in den Bereich der Planung fallen [Gut-17, S.44f.]. Es werden die Bereiche Planung, Realisierung und Betrieb unterschieden [VDI-3633a, S.6f.]. Die ereignisdiskrete Simulation ist dabei die beliebteste Simulationsmethode [Jah-10].

Die ereignisdiskrete Simulation kann zur Entscheidungsunterstützung bei der Neu- und Umplanung von Materialflusssystemen eingesetzt werden [Bra-11, S.122]. In Tabelle 2.1 auf Seite 10 sind gängige Be-

zeichnungen für Simulationsanwendungen benannt. Die Bezeichnungen wiederholen sich vereinzelt. Dies spiegelt die Breite des Anwendungsbereichs mancher Simulationswerkzeuge wider. In der Literatur sind vielfältig Hinweise darauf zu finden, dass ereignisdiskrete Simulation sich hervorragend zur Planungsunterstützung im Bereich der Ganzheitlichen Produktionssysteme eignet [Hom-18, S.371f.] [Uyg-15, S.116] [VDI-2689, S.13] [VDI-2870b, S.85].

Das „detaillierte physikalische, kinematische und kinetische Verhalten technischer Systeme“ wie beispielsweise beim Ur- und Umformen, Beschichten sowie Roboterbewegungen sind in dieser Arbeit nicht betrachtet [Wen-08c, S.1] [DIN-8580]. Menschmodelle, zum Zweck der Analyse der Ergonomie, werden ebenso nicht betrachtet [Gut-17, S.2] [Mär-11, S.6f.] [Sch-14c] [Wen-08c, S.1].

Im Folgenden werden relevante Begriffe aus dem Feld der Modellierung und Simulation benannt, der Aufbau und eine Klassifikation von Simulationswerkzeugen behandelt, die Referenz zur Multiagentensimulation hergestellt und die Durchführung von Simulationsstudien diskutiert.

2.2.1 Modellierung und Simulation

Die Planung von Materialflusssystemen kann durch Experimente mit einem digital nachgebildeten Modell des Materialflusssystems unterstützt werden. Ein Modell ist eine „vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System“ [VDI-3633b, S.11]. Es lassen sich physische und mathematische Modelle unterscheiden. Mathematische Modelle wiederum lassen sich in analytische Lösungen und Simulationen untergliedern [Law-15, S.4]. Ist die Zeit zur Lösung der Aufgabe mit analytischen Methoden wie der linearen Optimierung unangemessen hoch oder wenn stochastische Einflüsse berücksichtigt werden sollen, kommt eine Lösung durch eine Simulation infrage.

Der VDI *Fachausschuss 204 Modellierung und Simulation* definierte umfangreich Begriffe aus den Bereichen der Simulation von Logistik-, Materialfluss- und Produktionssystemen. Diese sind in der Richtlinienreihe VDI 3633 abgedruckt. Unter *Simulation* wird das „Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell [verstanden], um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind; insbesondere werden die Prozesse über die Zeit entwickelt. Anmerkung: Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden“ [VDI-3633a, S.3]. Ein *Simulationsmodell* ist eine „Abbildung eines realen oder geplanten Systems zum Zweck der Simulation“ [VDI-3633b, S.17].

Das Voranschreiten der Simulationszeit kann über verschiedene Vorgehensweisen realisiert werden. Abbildung 2.1 gibt einen Überblick über die sogenannten Simulationsmethoden [Mat-89] [Wen-08b, S.77ff.]. Im Gegensatz zu einem kontinuierlichen Zeitfortschrittsmechanismus, welcher i.d.R. durch Differenzialgleichungen gesteuert wird, finden in diskreten Modellen zu definierten Zeitpunkten Änderungen von Statusvariablen statt [Law-15, S.6ff., S.707ff.] [Ban-10b, S.32]. Der Zeitfortschritt in diskreten Modellen erfolgt entweder zeitgesteuert oder ereignisorientiert. WENZEL (2008) benennt die quasikontinu-

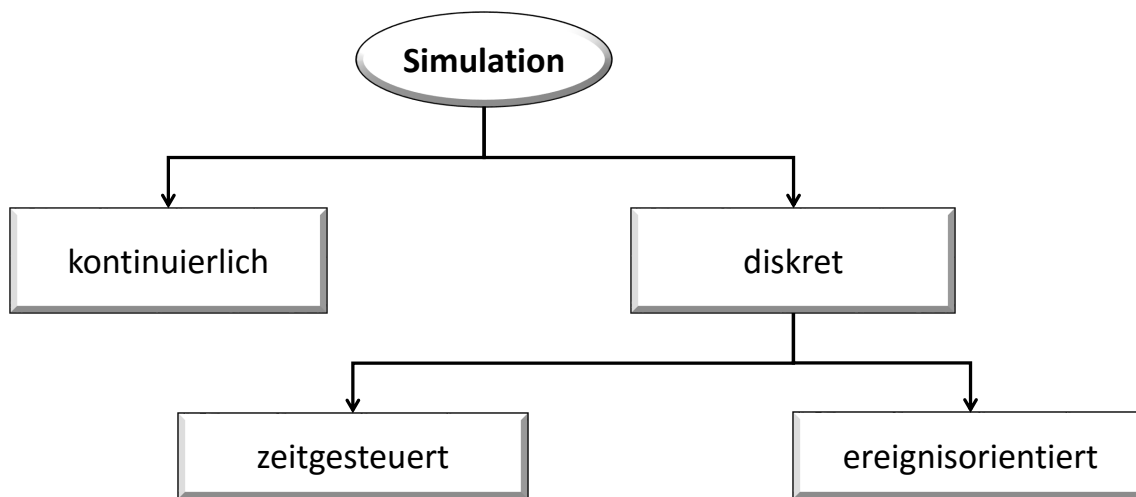


Abbildung 2.1: Klassifikation von Simulationsmethoden nach [Wen-08b, S.80] und [Mat-89, S.200]

ierliche Fortschreibung der Zeit synonym zur zeitgesteuerten; die Simulationszeit wird in festgelegten Zeitinkrementen Δt fortgeschrieben. Zustandsänderungen werden somit erst nach Erhöhung der Zeit durchgeführt. In einer zeitdiskret-ereignisorientierten Simulation, oder kurz ereignisdiskreten² Simulation, wird der Zeitfortschritt durch *Ereignisse*, *Aktivitäten* oder *Prozesse* bestimmt. Ein Ereignis ist grundsätzlich als atomar anzusehen und somit nicht weiter zerlegbar. Es verbraucht keine Simulationszeit. Ein über das Ereignis hinaus andauernder Zustand behält im Simulationsmodell bis zum nächsten Ereignis seine Gültigkeit [Wen-08b, S.73ff.] [Mat-89].

2.2.2 Simulationswerkzeuge

Simulationswerkzeuge werden dazu verwendet, die *Simulationszeit inklusive der Zustandsänderungen* fortzuschreiben sowie den *Aufbau und die Verwaltung eines Simulationsmodells*, das sogenannte *Modellierungskonzept*, zu unterstützen [Wen-08b, S.73ff.]. Ein Simulationswerkzeug ermöglicht eine softwaretechnische Nachbildung eines Systems in einem Modell. Simulationswerkzeuge vereinen in der Regel einen Simulatorkern, eine Datenverwaltung, eine Benutzungsoberfläche und verfügen über Schnittstellen zu externen Programmen [VDI-3633a, S.14f.]:

Simulatorkern: Hierunter wird der Programmteil verstanden, der die Modellwelt³ bereitstellt und die Erzeugung und Verarbeitung von Ereignissen ausführt. Der Simulatorkern verknüpft und

²In der Literatur finden sich synonyme Bezeichnungen, siehe auch Tabelle 2.1: Beispielsweise verwenden BRACHT ET AL. (2011) den Ausdruck der ereignisdiskreten Simulation [Bra-11, S.122]. Darüber hinaus geben die Autoren an, dass sich in der deutschen Literatur auch die Begriffe Ablaufsimulation und Materialflusssimulation wiederfinden. Auch die Begriffe diskret-ereignisorientierte Simulation, discrete-event simulation oder next-event simulation finden sich in Veröffentlichungen [Küh-06, S.46f.] [Poi-08, S.136] [Wen-08c, S.1].

³Unter der *Modellwelt* wird die „Summe der zur Verfügung stehenden Modellelemente eines Simulators“ verstanden [VDI-3633b, S.12]. Ein *Modellelement* wiederum kann als Komponente eines Modells angesehen werden, die nicht mehr weiter in untergeordnete Modellelemente zerlegt werden kann und/oder die sich hierarchisch übergeordnet ihrerseits aus Modellelementen zusammensetzt. Modellelemente können z.B. Klassen zugeordnet werden. Jedes Modellelement ist durch Attribute charakterisiert [VDI-3633b, S.12].

koordiniert die Prozesse einzelner Komponenten des Simulationswerkzeugs. Er kann als die zentrale Ablaufsteuerung verstanden werden und manipuliert die Modelldaten.

Datenverwaltung: Simulationswerkzeuge vereinigen in ihren Modellen die Eingabe- und Modelldaten mit den Attributen der Modellelemente und den zugehörigen Algorithmen, die aus Sicht der objektorientierten Programmierung auch Methoden genannt werden.

- Modelldaten sind bei der Modellentwicklung festgelegte Konstanten sowie während des Simulationslaufs berechnete Variablen. Modelldaten weisen einen dynamischen Charakter auf.
- Eingabedaten werden vom Nutzer des Modells bereitgestellt. Sie haben statischen Charakter.
- Darüber hinaus existieren Experimentdaten wie der Simulationszeitraum.
- Simulationsergebnisdaten werden während eines Simulationslaufs gesammelt, um während des Laufs oder danach genutzt zu werden.

Benutzungsoberfläche: Benutzungsoberflächen, im Englischen als Graphical User Interfaces (GUIs) bezeichnet, dienen als Schnittstelle zwischen der Software und dem Anwender zur Dateneingabe, Experimentdurchführung und Ausgabe von Simulationsergebnisdaten. Im einfachsten Fall ist eine Benutzungsoberfläche alphanumerisch menügeführt. Höher entwickelte Benutzungsoberflächen ermöglichen den grafisch-interaktiven Modellaufbau, z.B. durch Ziehen und Ablegen, im Englischen als *Drag and Drop* bezeichnet. In objektorientierten Simulationswerkzeugen ist ein Modellaufbau üblich, indem vordefinierte Klassenobjekte und -bausteine instanziiert und die Instanzen dann parametrisiert werden. Auch ist die Möglichkeit der Modellbeschreibung durch Programmiersprachen zur Implementierung von Simulationsmodellen oft gegeben.

Schnittstellen: Schnittstellen ermöglichen einen Datenaustausch mit anderen Datenbeständen. In der Regel werden durch Simulationswerkzeuge standardisierte Sprachen wie die Structured Query Language (SQL), die die Manipulation von Daten in Datenbanken ermöglicht, unterstützt.

Simulationswerkzeuge lassen sich in Simulationssprachen, Simulatorentwicklungsumgebungen und Simulatoren im Sinne von Programmpaketen einteilen [Wen-08b, S.73ff.]. Bei einer Simulationssprache handelt es sich um eine Programmiersprache, die den Problemstellungen der Simulation angepasst ist [VDI-3633b, S.18]. In der Regel stellen Simulationssprachen Erweiterungen von Programmiersprachen um simulationsspezifische Aspekte wie die Manipulation von Ereignissen dar [Wen-08b]. In Simulatorentwicklungsumgebungen kommen grafische Hilfsmittel zum Einsatz, die die Anwendung von strukturierten Techniken in der Software-Entwicklung erlauben [Noc-93, S.287]. Entscheidungsbäume, Entscheidungstabellen [DIN-66241] oder Nassi-Shneiderman Diagramme [DIN-66261] sind beispielhafte Diagramm-Techniken, die die Abbildung detaillierter logischer Entscheidungsstrukturen im Modellierungsprozess erlauben [Noc-93, S.287]. Im Allgemeinen richten sich Simulatorentwicklungsumgebungen an Simulatorentwickler und werden nicht dem Endanwender zur Verfügung gestellt [Wen-08b, S.82]. Simulatoren sind entweder unter Verwendung von Programmiersprachen oder von Simulatorentwicklungsumgebungen entstanden [VDI-3633b, S.18]. Simulatoren sind eigenständige Programmpakete mit den zuvor benannten Bestandteilen. Sie beinhalten in der Regel interaktive Modellierungshilfen und vordefinierte Bausteine [Wen-08b, S.82f.]. Angelehnt an WENZEL (2008) sind in dieser Arbeit Simulatoren,

die sich an Endanwender richten, als Simulationsinstrumente bezeichnet. Endanwender sind im Sinne dieser Arbeit Planer von Materialflusssystemen und Studierende der Produktionstechnik. Ereignisdiskrete Simulationsinstrumente stehen im Fokus dieser Arbeit.

2.2.3 Multiagentensimulation

Die Multiagentensimulation, synonym oft auch als agentenbasierte Simulation bezeichnet, kann dem Forschungsbereich der künstlichen Intelligenz zugeordnet werden [Bra-18, S.132]. Die Multiagentensimulation basiert auf der Idee der Multiagentensysteme. Die Multiagentensimulation ist ein Spezialfall der ereignisdiskreten Simulation. Das Systemverhalten wird maßgeblich durch die Interaktion von Agenten mit ihrer Umgebung und weiteren Agenten bestimmt [Law-15, S.694]. BRACHT, GECKLER UND WENZEL (2018) beschreiben Agenten als sozialfähig. Agenten handeln zielorientiert. Ziele verfolgen sie in Eigeninitiative. Sie verhalten sich interaktiv und kooperativ zu anderen Agenten [Bra-18, S.132f.].

Ein konkretes Beispiel für die Multiagentensimulation wurde am Beispiel der Baustellenlogistik veröffentlicht [Fra-13]. Ein Fallbeispiel aus dem Wareneingang an einer Baustelle ist als Petri-Netz⁴ abgebildet. Der Agent Stapler verhandelt mit Agenten aus zwei Kommissionierbereichen über die Priorisierung von Aufträgen. Die Implementierung des Szenarios erfolgte mit dem Simulationswerkzeug *renew* der Universität Hamburg [Kum-04].

HARJES UND SCHOLZ-REITER (2013) benennen ein Beispiel zur Simulation selbststeuernder Logistik im Umlaufmanagement von Verleihartikeln [Har-13]. Sie beschreiben die Entwicklung einer Multiagentensimulation an einem realen Beispiel aus der Veranstaltungslogistik. Produkte, Fahrzeuge und Personal werden in dem Ansatz als Agenten mit individuellen Zielen repräsentiert. Terminrestriktionen und Veranstaltungsorte werden im Modell der realen Situation nachgestellt. Die Agenten verhandeln kooperativ. Im Ergebnis liegt nach einem Simulationslauf eine Kommissionier-, Pack- und Ladeliste sowie die Tourenplanung für die Transportmittel vor. Die Autoren beschreiben, dass alle im Simulationsmodell implementierten Termine - wie auch in der Realität - eingehalten werden konnten.

Außerhalb der Produktion finden sich weitere Ansätze mit Potenzial zur Übertragung auf Materialflusssysteme. MEYER UND KLUSSMANN (2017) berichten über die Entwicklung eines ereignisdiskreten Simulationsmodells zur Evakuierung von Kreuzfahrtschiffen [Mey-17]. Sie zielen darauf ab, die Zeit zur Evakuierung sowie Engpässe ermitteln zu können. Die Autoren beschreiben die Modellierung eines Schiffsdecks. Dieses Deck wurde in sogenannte Zellen unterteilt. Eine Zelle entspricht der Standfläche eines Passagiers von $0,16\text{m}^2$ im dichten Gedränge. Es kann sich zu einem Zeitpunkt maximal ein Passagier in einer Zelle befinden. Die Passagiere wurden in demographische Gruppen eingeteilt. Mobilitätseingeschränkte Personen, Treppen und Sperrflächen wurden bei der Modellierung berücksichtigt. Jeder Grup-

⁴Mit Petri-Netzen lassen sich parallelläufige Prozesse beschreiben. Petri-Netze gehen auf die Dissertation von PETRI im Jahr 1962 zurück [Pet-62]. Sie bestehen aus Knoten, Plätzen und Transitionen. Plätze werden durch Kanten mit Transitionen verbunden und umgekehrt [VDI-4008a, S.5f.]. Petri-Netze sind Beschreibungsmittel. Beschreibungsmittel sind Werkzeuge für die Modellierung [Wie-87, S.31].

pe wurden andere Gehgeschwindigkeiten und Reaktionszeiten zugeordnet. Die Bildung von Staus vor Treppen und bei plötzlich auftretenden Hindernissen konnte durch die Simulation nachgestellt und analysiert werden. Einen ähnlichen Ansatz der Simulation von Personen präsentierten ENGEL UND NEHLS (2014). Sie untersuchten die Zeit zur Evakuierung des Bahnhofs Stuttgart 21 [Eng-14]. Beide Ansätze wurden in dem Simulationsinstrument Tecnomatix Plant Simulation umgesetzt. Dieses Simulationsinstrument ist dem Anwendungsbereich der Produktion und Logistik zuzuordnen. Beide Beispiele scheinen hinsichtlich der Entscheidungsfindung von autonomen logistischen Gegenständen des Materialflusses im Sinne der Wegfindung auch auf Materialflusssysteme übertragbar zu sein.

In der Literatur finden sich Hinweise auf spezialisierte Simulationswerkzeuge für die Simulation von Agenten. In dem SFB 637 an der Universität Bremen entstand das agentenbasierte Simulationswerkzeug PlaSMA (Platform for Simulations with Multiple Agents) [Geh-10b] [War-10] [Har-13] [DFG-19]. Bei dieser Software handelt es sich um die auf logistische Problemstellungen modifizierte Software JADE [Geh-10b]. Das ereignisdiskrete Simulationswerkzeug ermöglicht dem Benutzer die Konfiguration von Agenten, die Nachrichten senden und empfangen sowie Aktionen im Rahmen definierter Verhaltensweisen auslösen können [TZI-19]. Gemäß der Beschreibung auf der Website repräsentiert ein Agent entweder eine Umgebung, eine Organisation oder physische Objekte. Modelle, die mit der im SFB 637 auf Basis von UML (Unified Modeling Language) entwickelten Beschreibungsmethode „Autonomous Logistic Engineering Methodology“ formalisiert wurden, können automatisch in ausführbare Simulationsmodelle überführt werden [DFG-19].

Eine weitere führende Software für die Multiagentensimulation ist AnyLogic [Mac-14]. In ihrer Studie über Entwicklungsumgebungen für Multiagentensimulationen kommen KERWITZ UND CLASEN (2016) zum Ergebnis, dass sich mit der sehr flexiblen Software Probleme aus einem breiten Anwendungsfeld lösen lassen [Ker-16]. Zur Erstellung einfacher Modelle werden keine Programmierkenntnisse benötigt. Der Aufwand zur Einarbeitung in die kostenpflichtige Software AnyLogic wird von den Autoren kritisiert.

Zusammenfassend sind Multiagentensimulationen als mächtige Werkzeuge zur Analyse komplexer Zusammenhänge bei logistischen Fragestellungen einzuschätzen. Die Bedeutung dieser Methode scheint in Zeiten zunehmend dezentral gesteuerter Prozesse, wie diese bei den CPPS im Kontext der vierten industriellen Revolution diskutiert werden, zuzunehmen. Auch BRACHT, GECKLER UND WENZEL (2018) gehen von einer stärkeren Verbreitung von Multiagentensystemen im Rahmen von Industrie 4.0 aus [Bra-18, S.133]. Aus dem Stand der Wissenschaft und Technik lässt sich aber auch ableiten, dass die Werkzeuge zur Modellierung und Simulation von Agenten sehr umfangreich sind. Deren Bedienung erfordert sehr IT-affine Mitarbeiter. Die zunehmende Bedeutung der Multiagentensimulation soll in dieser Arbeit nicht außer Acht gelassen werden. Vor diesem Hintergrund werden flexible ereignisdiskrete Simulationsinstrumente betrachtet, die auch eine Modellierung und Simulation von Agenten im Kontext der Materialflusssysteme ermöglichen. Beispiele für die Multiagentensimulation finden sich in den Anwendungsbeispielen wieder.

2.2.4 Durchführung von Simulationsstudien

Eine Simulationsstudie ist ein „Projekt zur simulationsgestützten Untersuchung eines Systems“ [VDI-3633b, S.18]. Sie besteht maßgeblich aus der Vorbereitungs-, Ausführungs- und Bewertungsphase [VDI-3633a, S.18ff.] [Law-15, S.66ff.] [Küh-06, S.21]. Während der ersten Phase werden das Problem und Ziel der Untersuchung spezifiziert und relevante Daten gesammelt. In der Ausführungsphase ist ein initiales Simulationsmodell zu erstellen, zu verifizieren und zu validieren. Die Verifikation bezeichnet einen Prozess, bei dem die Korrektheit von Programmen oder Programmteilen formal nachgewiesen wird [VDI-3633b, S.21]. Eine Verifikation kann durch Syntax-Checks oder Konsistenz-Checks erfolgen [VDI-3633b, S.21]. Die Validierung bezeichnet die Prüfung auf hinreichende Übereinstimmung zwischen dem Modell und der Realität [VDI-3633a, S.37]. Laut der Richtlinie VDI 3633 ist sicherzustellen, dass das Modell das Verhalten des Materialflusssystems genau genug und fehlerfrei widerspiegelt. Nachdem die Simulationsexperimente gemäß einem Experimentplan durchgeführt wurden, sind in der Bewertungsphase die Ergebnisse der Experimente zu analysieren und Änderungen am Materialflusssystem vorzunehmen.

In der Literatur finden sich unterschiedliche Vorgehen zur Durchführung einer Simulationsstudie, die im Hinblick auf ihre Struktur ähnlich sind. In der Richtlinie VDI 3633 ist vorgeschlagen, dem Vorgehen von RABE ET AL. (2008) zu folgen [VDI-3633a] [Rab-08]. In dieser Vorgehensweise ist insbesondere die Validierung und Verifikation während der Simulationsstudie besonders hervorgehoben. LAW (2015) macht darauf aufmerksam, dass die Aufgaben bei der Durchführung einer Simulationsstudie iterativ bearbeitet werden können [Law-15, S.66ff.]. Der Verband der Automobilindustrie (VDA) veröffentlichte im Jahr 2013 mit der Richtlinie VDA 4811 ein Vorgehen zur Durchführung von Simulationsstudien im Hinblick auf die Besonderheiten in der Automobilindustrie. Das Befolgen dieser Empfehlung soll eine Mindestqualität bei der Durchführung von Simulationsstudien in der Automobilindustrie sicherstellen. Auch in diesem Vorgehen ist ein besonderer Fokus auf die Validierung und Verifikation gelegt [VDA-4811]. KÜHN (2006) stellt den Ablauf von Simulationsstudien in dem sogenannten Simulationszyklus dar [Küh-06]. Im Gegensatz zu dem in der Richtlinie VDI 3633 benannten Ablauf wird weniger der Modellaufbau in den Vordergrund gestellt, als vielmehr die Ergebnisanalyse und -interpretation. Der Zyklus ist in Abbildung 2.2 dargestellt. Er ist um die in der Richtlinie VDI 3633 und von RABE ET AL. (2008) benannten Schritte zum Aufbau von Simulationsmodellen detailliert abgebildet. Ständige und konsequente Validierung und Verifikation gilt auch für diesen Zyklus. Dieser kann mehrmals durchlaufen werden.

Das *Materialflusssystem* kann entweder ein existierendes Materialflusssystem oder ein gedankliches Modell eines Materialflusssystems sein. Das digitale *Simulationsmodell* wird vom Materialflusssystem abstrahiert. Abstraktion bezeichnet ein „Verfahren zur Reduzierung der Komplexität eines Problems durch Trennung der für bestimmte Aspekte der Problemlösung wichtigen und unwichtigen Details des Problems“ [VDI-3633b, S.3]. Es lassen sich zwei Verfahren unterscheiden [VDI-3633b, S.3]:

- Reduktion bzw. Abstraktion im engeren Sinne: Verzicht auf die Abbildung unwichtiger Details.
- Idealisierung: Vereinfachung von realen Gegebenheiten.

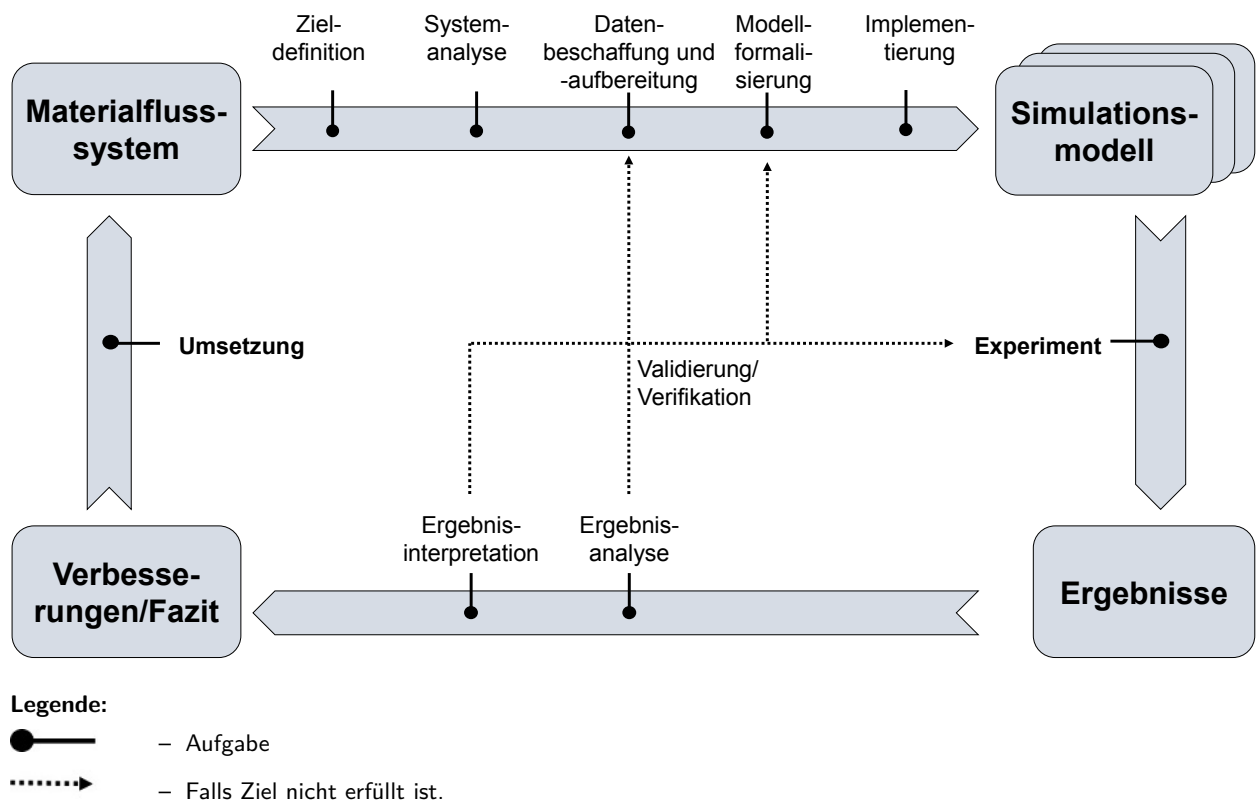


Abbildung 2.2: Simulationszyklus: Durchführung einer Simulationsstudie nach [Abd-13], [Küh-06, S.21], [Rab-08, S.5] und [VDI-3633a]

Folglich weist ein Modell lediglich eine beschränkte Gültigkeit auf. Anhand eines zuvor definierten Plans werden nun Experimente durchgeführt. Hier kommen häufig Sensitivitätsanalysen zum Einsatz. Bei diesen Analysen handelt es sich um eine „Untersuchung des Systemverhaltens bei Änderung einer einzelnen Einflussgröße, während alle anderen konstant bleiben“ [VDI-3633b, S.16]. Die *Ergebnisse* der Experimente werden im nächsten Schritt durch den Anwender im Hinblick auf das Untersuchungsziel analysiert und interpretiert. Da das digitale Modell eine Abstraktion der Realität darstellt, gilt es, die Simulationsergebnisdaten am Materialflusssystem zu kontextualisieren und so Verbesserungen abzuleiten. Das bedeutet, dass der Anwender prüft, ob reduzierte oder idealisierte Eigenschaften eine negative Auswirkung auf eine mögliche Umsetzung in der Realität haben. Die *Verbesserungen* werden schließlich in der Realität umgesetzt. Erneut kann der Zyklus durchlaufen werden.

Die Erfahrung des Simulationsanwenders spielt bei der Durchführung von Simulationsstudien eine wichtige Rolle für deren Erfolg [May-12] [May-15a]. Insbesondere die Abstraktion beim Aufbau von Simulationsmodellen und die Ergebnisinterpretation erfordern Erfahrung [Ban-10b, S.23f.] [Sha-98].

2.3 Qualifizierung von Simulations-Anwendern

Für unterschiedliche Klassen von Simulationswerkzeugen sind im Folgenden Anforderungen an die Qualifikation von Benutzern beschrieben. Gängige Konzepte zur Modellierung mit Simulationsinstrumenten werden daraufhin ermittelt. Die Relevanz von Bausteinen und von Programmiersprachen zur Implementierung von Simulationsmodellen mit Blick auf handelsübliche flexible Simulationsinstrumente ist dargestellt. Die Potenziale der (teil-)automatischen Modellgenerierung sowie von Simulation in der Cloud sind herausgearbeitet.

2.3.1 Wissensdomänen und Simulationsinstrumente

SCHMIDT beschrieb bereits im Jahr 1988 Ausbildungsinhalte für sogenannte Simulationstechniker. Konkret benennt er die Felder Simulationstechnik und Informatik. Weiterhin sind Kenntnisse in einem Anwendungsbereich wie der Montage erforderlich. Als Basis-Kenntnisse beschreibt er mathematische und erkenntnistheoretische Grundlagen [Sch-88, S.34ff.]:

Erkenntnistheorie: Grundlagen der Philosophie, allgemeine Systemtheorie, Modelltheorie.

Mathematik: Numerische Mathematik, Differentialgleichungen, lineare Algebra, diskrete algeb. Strukturen, Logik, Statistik und Wahrscheinlichkeit.

Informatik: Hardware, Softwaretechnologie, Sprache, Grafik.

Simulationstechnik: Simulatoren, Ablaufkontrolle, Systementwurf, Verfahren und Methoden wie die Erzeugung von Zufallszahlen, Integrationsverfahren und Optimierung.

Anwendungsbereich: Übersichtswissen, beispielsweise aus den Bereichen Produktionssysteme, Fertigung, Montage, Lager oder Transport.

Im Laufe der Jahre haben sich die Simulationswerkzeuge weiterentwickelt. Beispielsweise sind viele Werkzeuge mit leistungsstarken Modulen zur grafischen Animation ausgestattet. Einfache GUIs machen detailliertes Wissen des Anwenders von Simulationswerkzeugen über die Funktionsweise grafischer Animation weitestgehend überflüssig. In einer neueren Veröffentlichung benennen GUTENSCHWAGER ET AL. (2017) „Fachwissen aus der Statistik, der quantitativen Betriebswirtschaftslehre und der Informatik sowie aus Anwendungssicht aus den Ingenieurwissenschaften und dem Projektmanagement“ als Voraussetzung für eine sogenannte qualifizierte Anwendung der ereignisdiskreten Simulation [Gut-17, S.2].

Es empfiehlt sich, die erforderlichen Kenntnisse von Nutzern differenziert hinsichtlich auf dem Markt befindlicher Simulationswerkzeuge zu betrachten. In Anlehnung an SCHMIDT (1988) und NOCHE UND WENZEL (1991) veröffentlichte WENZEL (2008) eine Taxonomie, in welcher Simulationswerk-

zeuge nach Anwendungsbezug unterteilt sind, vgl. Abbildung 2.3 [Wen-08b, S.84] [Sch-88, S.19ff.] [Noc-91]. Die Taxonomie besteht aus fünf Ebenen. Dargestellt sind Klassen von Simulationswerkzeugen-

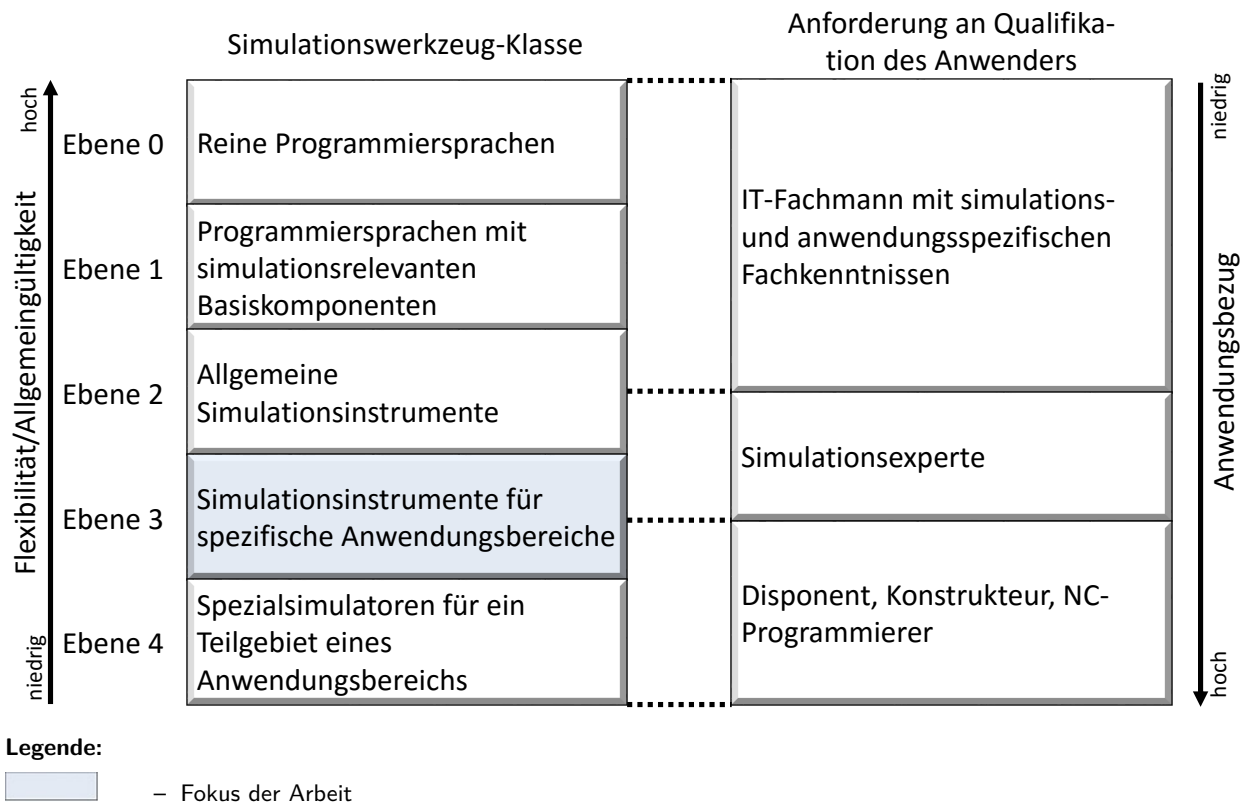


Abbildung 2.3: Einordnung von Simulationswerkzeugen unter den Aspekten Flexibilität, Allgemeingültigkeit und Anwendungsbezug nach [Wen-08b, S.84], [Sch-88, S.19ff.], [Noc-91] und [VDI-3633a, S.10, S.32]

gen, die mit steigender Ebene an Anwendungsbezug gewinnen und an Flexibilität und Allgemeingültigkeit verlieren. Dem gegenübergestellt ist die erforderliche Qualifikation des Anwenders. Mit steigender Allgemeingültigkeit der Simulationswerkzeuge hinsichtlich ihrer Anwendung ist spezifisches Wissen über die Simulationstechnik und IT erforderlich und umgekehrt. Die Ebenen bauen wie folgt aufeinander auf [Wen-08b, S.82f.] [Noc-93, S.287ff.]:

Ebene 0: Reine Programmiersprachen. Diese Sprachen weisen keine simulationsspezifischen Eigenschaften auf.⁵

Ebene 1: Programmiersprachen mit simulationsrelevanten Basiskomponenten, z.B. Klassenkonzepte oder Ereignisverwaltung bei ereignisdiskreten Werkzeugen.

Ebene 2: Wie Ebene eins, jedoch erweitert um für Modellklassen spezifische Komponenten. Beispielsweise bieten Petri-Netze und Sprachkonzepte die Basis für eine Vielzahl an Anwendungen.

⁵NOCHE ET AL. (1993) wählten die Ebene null bewusst als Beginn der Aufzählung, da Programmiersprachen wenig Unterstützung mit Blick auf simulationsspezifische Belange bieten. Simulationswerkzeuge werden erst auf Ebene eins relevant [Noc-93].

Ebene 3: Wie Ebene zwei, jedoch mit spezialisierten Komponenten im Hinblick auf einzelne Anwendungsbereiche, z.B. Produktion und Logistik, Computersysteme und Computer-Netzwerke [Ban-05, S.423ff.].

Ebene 4: Spezifische Simulationsinstrumente mit Komponenten zur Simulation von Fragestellungen aus einem Teilgebiet eines Anwendungsbereichs. Ein Beispiel sind Spezialsimulatoren zur Planung und Gestaltung von FTS. Zielstellungen können die Ermittlung der Anzahl erforderlicher FTS oder die Bestimmung der Wegfindungsstrategie sein [VDI-2710].

Ein Experte der Simulationstechnik, auch Simulationsexperte genannt, ist in der Lage, „die Aufbaustruktur und das Verhalten technischer Systeme, die Prozessabläufe sowie Organisations- und Steuerstrategien im Medium Software zu formulieren und auf diese Weise Werkzeuge für die Aufgabenlösung in Form problemspezifischer Simulationsmodelle zur Verfügung zu stellen“ [VDI-3633a, S.10]. „Der Modellierungs- und Simulationsexperte ist Know-how-Träger im Bezug auf die Modellierungswerkzeuge, die jeweiligen Modelle und die statistische Auswertung. Er formuliert den Datenbedarf, erstellt die Modelle, führt die Experimente durch und unterstützt bei der Interpretation der Ergebnisse“ [VDI-4465, S.21]. Die Modellierung durch Anwender mit geringen Kenntnissen in der Simulationstechnik und IT kann unter Nutzung von Werkzeugen mit hoher Anwendungsnähe ohne die Unterstützung eines Simulationsexperten vorgenommen werden. Die Anwender bleiben so nämlich in ihrer Vorstellungs- und Begriffswelt [VDI-3633a, S.32].

LAW (2015) vertritt eine ähnliche Auffassung unter Nutzung anderer Begrifflichkeiten. Er unterteilt Simulationswerkzeuge in sogenannte universelle Programmiersprachen und Simulationspakete [Law-15, S.182ff.]. Für universelle Programmiersprachen benennt er als Beispiele C, C++ und Java. Solche Sprachen erlauben dem Programmierer größere Flexibilität als Simulationspakete, erfordern jedoch Programmierkenntnisse und - verglichen mit Simulationspaketen - längere Entwicklungszeiten für Simulationsmodelle. Simulationspakete lassen sich nach LAW (2015) in universelle und anwendungsorientierte einteilen. Erstere können seinen Ausführungen nach sehr umfangreich für nicht näher spezifizierte Anwendungen genutzt werden und besondere Funktionalitäten für gewisse Anwendungen, z.B. in der Produktion, aufweisen. Anwendungsorientierte Simulationspakete können in definierten Anwendungsbereichen wie der Produktion, dem Gesundheitswesen oder den Kommunikationsnetzwerken eingesetzt werden.

Besonders geeignet zur Entscheidungsunterstützung bei der Neu- und Umplanung von Materialflusssystemen sind die Simulationsinstrumente der Ebene drei. Die Bedienung allgemeiner Simulationsinstrumente (Ebene 2), die kaum Anwendungsbezug aufweisen, ist eher komplex und erfordert daher in der Regel lange Einarbeitungszeiten [Wen-08b, S.82f.]. Spezialsimulatoren (Ebene 4) sind auf ein spezifisches Teilgebiet eines Anwendungsbereichs wie die Routenplanung von fahrerlosen Transportsystemen [VDI-2710], Gestaltung von Geschäftsprozessen oder die Auslegung von Staplersystemen begrenzt [Wen-08b, S.82f.] [Noc-93, S.289f.]. Ein Simulationsinstrument für spezifische Anwendungsbereiche (Ebene 3) ist zur Lösung unterschiedlicher Fragestellungen bei der Neu- und Umplanung von Materialflusssystemen geeignet. Die Begriffswelt des Anwendungsbereichs wird nicht gänzlich verlassen. Beispiele für die Flexibilität des Einsatzes dieser Simulationsinstrumente sind die Auslegung von Supermärkten im Sinne eines Kun-

denentkopplungspunkts im Kontext eines betriebsspezifischen Materialflusses beim Gasturbinenbau oder die Dimensionierung von Transportwegen unter Berücksichtigung des Zusammenspiels von Entscheidungen autonomer logistischer Gegenstände des Materialflusses. Es ist vorstellbar aufgrund des hohen Anwendungsbezugs vermehrt auf Spezialsimulatoren zu setzen, jedoch müssten für jede Planungsaufgabe ein neues Simulationsinstrument erstellt/beschafft und gewartet werden, Mitarbeiter geschult und eine redundante Modellierung und Datenhaltung in Kauf genommen werden.

Die in der Arbeit fokussierte Ebene drei ist als ein Kompromiss zwischen der Flexibilität einerseits und dem Anwendungsbezug sowie der Fachsprache andererseits zu sehen. Die Simulationsinstrumente der Ebene drei ermöglichen die flexible Modellierung und Simulation unterschiedlicher Methoden des Fließprinzips und des Pull-Prinzips sowie von autonomen logistischen Gegenständen des Materialflusses. Sie eignen sich zur Simulation von Materialflusssystemen unterschiedlicher Aggregation vom Produktionsnetz bis zum Arbeitsplatz [VDI-5200, S.7].

2.3.2 Konzepte zur Modellierung

Das Modellierungskonzept dient dem Aufbau und der Verwaltung des Simulationsmodells [Wen-08b, S.73ff.]. Die Beschreibung von Modellen kann gemäß der Richtlinie VDI 3633 in grafikorientiert, parameterorientiert und sprachorientiert klassifiziert werden [VDI-3633a, S.12]:

Grafikorientierte Modellbeschreibung: Standardisierte Symbole oder anwenderdefinierte Grafiken stehen als Bausteine zur Darstellung der Aufbau- und/oder Ablaufstruktur des Modells zur Verfügung. Parametereingaben und textliche Beschreibungen zur Programmierung der anwenderdefinierten Bausteineigenschaften können diese ergänzen. Beispiele für die grafikorientierte Modellbeschreibung können generische Konzepte und Bausteinkonzepte sein [Wen-08b]:

- *Generische Konzepte* sind durch allgemeine, vielfach verwendbare und hinsichtlich ihrer Beschreibungsmethodik anwendungsneutrale Modellkonstrukte gekennzeichnet. Diesen Konstrukten werden gewisse Fähigkeiten zugeordnet und sie können über Attribute näher charakterisiert werden. Zu dieser Kategorie zählen objektorientierte Modellierungskonzepte.
- Bei *Bausteinkonzepten* handelt es sich um ein anwendungsbezogenes Modellierungskonzept, da die Beschreibungsmittel sich begrifflich an den abzubildenden Systemen der Anwendung orientieren. Bausteine sind für einen bestimmten Anwendungsbereich topologisch, organisatorisch und/oder informatorisch zweckmäßig vordefiniert und aggregiert. Der Anwender kann diese parametrisieren. Die Anwendungsnähe dieses Modellierungskonzepts erlaubt nur eine eingeschränkte Modellsicht. Ein *Baustein* ist in der Richtlinie VDI 3633 als Modellelement „mit einer eigenen internen Ablauflogik“ definiert. Man unterscheidet physische und logische Bausteine. Ein *physischer Baustein* bildet körperlich vorhandene Gegenstände oder Lebewesen der Realität wie Maschinen oder Werker ab [VDI-3633a, S.26f.]. Physische Bausteine können dabei stationär, z.B. Lager oder Füllstation, oder mobil sein, z.B. mobiler Montageroboter oder Fahrzeug. Ein *logischer Baustein* bildet gedankliche Einheiten wie

Fertigungsbereiche oder Steuerungen ab [VDI-3633b, S.11]. Weiterhin kann in *vorkonfigurierte* und *anwenderdefinierte Bausteine* unterschieden werden. Ein *vorkonfigurierter Baustein* kann „durch Parametereingaben den konkreten Gegebenheiten des realen Systems angepasst werden“ [VDI-3633b, S.21]. Ein *anwenderdefinierter Baustein* ist ein vom „Anwender selbst erstellter Baustein“ [VDI-3633b, S.3]. Dieser kann bspw. durch Verknüpfung vorkonfigurierter oder anwenderdefinierter Bausteine erstellt werden. Modellelemente ohne eigene interne Ablauflogik werden *Objekte* genannt [VDI-3633a, S.26f.] [VDI-3633b, S.13]. Es werden physische Objekte und Informationsobjekte unterschieden, die je in temporärer oder permanenter Ausprägung auftreten können. Ein Beispiel für ein temporäres physisches Objekt ist ein Werkstück. Ein Beispiel für ein permanentes physisches Objekt ist ein Ladehilfsmittel. Ein temporäres Informationsobjekt kann ein Auftrag, ein permanentes ein Arbeitsplan sein.

Parameterorientierte Modellbeschreibung: Diese ist hauptsächlich für anwenderspezifisch entwickelte Werkzeuge sinnvoll. Beispiele hierfür können anwendungsorientierte Modellierungskonzepte und Referenzmodelle sein [Wen-08b]:

- *Anwendungsorientierte Modellierungskonzepte* orientieren sich in ihrer Begrifflichkeit an den abzubildenden Systemen der Anwendung. Im Gegensatz zu den übrigen Modellierungskonzepten implizieren diese Konzepte i.d.R. eine eingeschränkte Welt- bzw. Modellsicht und ggf. nur eine aufbau- oder ablauforientierte Sichtweise.
- Mit *Referenzmodellen* werden Anwendungen systematisch beschrieben. Es wird somit Erfahrungswissen dokumentiert, um den Aufwand für die Erstellung von Simulationsmodellen zu reduzieren.

Sprachorientierte Modellbeschreibung: Hier werden Simulationssprachen genutzt, mit deren Hilfe beliebige Systeme und Prozesse nachgebildet werden können. Sprachkonzepte zeichnen sich dadurch aus, dass Programmiersprachen oder Simulationssprachen zur Modellierung genutzt werden [Wen-08b]. Kennzeichnend für Sprachkonzepte ist, dass das Modell in Form einer solchen Sprache umgesetzt ist und damit eine hohe Flexibilität beim Modellaufbau aufweist.

In der Regel werden Mischformen der Modellbeschreibungsarten als Modellierungskonzept in Simulationssinstrumenten angeboten [VDI-3633a, S.12]. Im produktionstechnischen Kontext werden Simulationssinstrumente, die eine Modellbeschreibung nur mit Programmier- oder Simulationssprachen unterstützen, Ebenen null und eins, kaum noch eingesetzt [Küh-06, S.204].

In dieser Arbeit ist unter einer Programmiersprache zur Implementierung von Simulationsmodellen eine Simulationssprache zu verstehen, die in Kombination mit weiteren Modellbeschreibungsarten in Simulationssinstrumenten zum Aufbau und für die Verwaltung von Simulationsmodellen angeboten wird. Der Begriff Simulationssprache wird in der Arbeit auf die Bezeichnung von Simulationssinstrumenten der Ebene 1 begrenzt.

Um in der Praxis gängige Modellierungskonzepte in Simulationsinstrumenten zu identifizieren, sind im Folgenden marktübliche Simulationsinstrumente analysiert. Dabei ist der Fokus auf Simulationsinstrumente der Ebene 3 im Kontext der Anwendung zur Neu- und Umplanung von Materialflusssystemen gerichtet. In der wissenschaftlichen Literatur können hierzu maßgeblich Berichte von zwei Forschern bzw. Forschergruppen identifiziert werden. SWAIN (2015, 2017) führt jedes zweite Jahr Umfragen unter Herstellern von Simulationswerkzeugen durch [Swa-15] [Swa-17]. Er fordert die Hersteller auf, Informationen über ihre Produkte in einer Datenbank zu hinterlegen. Er identifiziert in seiner Umfrage unter 26 Herstellern 44 Produkte. Von diesen lassen sich elf als Monte-Carlo-Werkzeuge⁶ und 15 den Ebenen null, eins und vier, siehe Abbildung 2.3 auf Seite 21, zuordnen. Diese in Summe 26 Produkte, sowie vier Mehrfach-Einträge, sind im Folgenden von der Betrachtung ausgeschlossen. Die verbleibenden 14 Simulationsinstrumente sind als relevante Simulationsinstrumente bezeichnet. Ein Auszug der Fragen von SWAIN und die Antworten der Hersteller sind wörtlich im Anhang A, Tabelle A.1, abgedruckt. Diese Stichprobe sei wiederum mit dem von DIAS ET AL. (2016) veröffentlichten Ranking der Beliebtheit ereignisdiskreter Simulationsinstrumente abgeglichen [Dia-16]. Das Forscher-Team verfolgt den Ansatz, die Nutzungsintensität von Simulationsinstrumenten anhand unterschiedlicher Datenquellen wie der Anzahl der Treffer bei Google oder der Häufigkeit der Benennungen in sozialen Medien zu quantifizieren [Dia-07]. Die Analyse lässt keinen direkten Rückschluss auf die Qualität von Simulationsinstrumenten zu. DIAS ET AL. (2016) identifizierten 19 Simulationsinstrumente, von denen elf auch durch SWAIN benannt sind. Es deutet sich an, dass SWAIN die wichtigsten Simulationsinstrumente benannt hat. Diese Vermutung lässt sich darüber hinaus auch durch die Analyse von GRUNDIG (2015) stützen. Er benennt folgende gängige Simulationsinstrumente, die seinen Ausführungen nach bausteinorientiert sind, also einen grafisch-interaktiven Modellaufbau unter Verwendung von Bausteinen erlauben: Arena, AutoMod, Enterprise Dynamics, Factory CAD, Flexsim, ProModel, Quest, Tecnomatix Plant Simulation und Witness [Gru-15, S.248] [Gru-13]. Es ist ersichtlich, dass weitere marktgängige Simulationsinstrumente existieren, die nicht durch DIAS ET AL. (2016) benannt wurden. Diese erlauben auch einen grafisch-interaktiven Modellaufbau unter Verwendung von Bausteinen. Unter der somit erhärteten Annahme, dass es sich bei SWAIN um einen repräsentativen Überblick über die Eigenschaften marktgängiger Simulationsinstrumente handelt, lassen sich folgende Erkenntnisse ableiten:

- 14 von 14 Simulationsinstrumenten haben Schnittstellen, um automatisiert mit weiterer Software wie Datenbanken oder Tabellenkalkulationen zu kommunizieren.
- 14 von 14 Simulationsinstrumenten können durch eine andere Software gesteuert werden.
- 12 von 14 Simulationsinstrumenten unterstützen einen grafischen Modellaufbau durch Ziehen und Ablegen.
- 13 von 14 Simulationsinstrumenten unterstützen einen Modellaufbau unter Nutzung von Programmiersprache bzw. bieten Zugriff auf sogenannte programmierte Programmteile.

⁶Bei der Monte-Carlo-Methode handelt es sich um eine statische Betrachtung eines Systems [Ros-11, S.13]. Der Begriff wurde zu Beginn des Computerzeitalters im Jahr 1947 geprägt; die relevanten stochastischen Grundlagen lassen sich auf das Jahr 1777 zurückführen [Bie-13, S.4]. „Die Monte-Carlo-Methode kann als Methode zur Modellierung von Zufallsgrößen mit dem Ziel, spezielle Eigenschaften ihrer Verteilungen zu berechnen, definiert werden. In der Regel wird diese Modellierung auf Computern verwirklicht, obwohl in einigen Fällen die Methode auch „von Hand“ unter Zuhilfenahme einer Aufstellung von Zufallsziffern angewandt werden kann“ [VDI-4008b, S.2].

Es lässt sich im Kern festhalten, dass die Mehrheit untersuchter Simulationsinstrumente der Ebene drei eine grafisch-interaktive Modellbeschreibung unter Verwendung von Bausteinen/Objekten ermöglicht. Der Einsatz einer Programmiersprache zur Implementierung von Simulationsmodellen ist bei nahezu allen betrachteten Simulationsinstrumenten möglich. Die Steuerung der Simulationsinstrumente durch andere Software ist durch Schnittstellen immer realisierbar. Die Relevanz sowie das Potenzial dieser Erkenntnisse sind in den folgenden Abschnitten vertiefend diskutiert.

2.3.2.1 Bausteine und der grafisch-interaktive Modellaufbau

RUDEL (2016) definiert diese in Ihrer Dissertation Bausteinkästen als Sammlung von Bausteinen. Bausteine sind danach funktionell abgrenzbare Elemente mit definiertem Inhalt und Umfang, die sich unter Beachtung von Regeln kombinieren lassen [Rud-16]. Sie stützt sich dabei auf eine Literaturanalyse. KÜHN (2006) bezeichnet mit sogenannten Bausteinsimulatoren solche, die „mehr oder weniger spezialisierte Bausteine für verschiedene Systemkomponenten zur Verfügung“ stellen [Küh-06, S.204f.]. Bausteine müssen gemäß seiner Ausführung nur noch zusammengesetzt und parametrisiert werden. Der Einsatz von Bausteinkästen ist in der Industrie und Forschung sehr verbreitet. Es handelt es sich um ein anwendungsbezogenes Modellierungskonzept. Bausteinkästen sind meist so gestaltet, dass die Fachsprache auf ein Anwendungsgebiet bezogen ist.

Der VDA, Unterarbeitsgruppe Ablaufsimulation, entwickelte für das von den Mitgliedern meistgenutzte Simulationsinstrument Tecnomatix Plant Simulation einen Bausteinkasten sowie eine Modellierungsrichtlinie [May-10]. Ein Schulungskonzept gibt einen Überblick über den Bausteinkasten. Tutorials erlauben das selbstständige Einarbeiten in den Bausteinkasten. Darüber hinaus werden zur Einarbeitung in den Bausteinkasten Workshops an Universitäten angeboten. Kenntnisse in der Anwendung des Simulationsinstruments werden vorausgesetzt, um den Bausteinkasten nutzen zu können.

Ein weiteres Beispiel findet sich in RUDEL (2016 UND 2017). Sie beschreibt die Entwicklung des Baukastensystems *KMUSimMetall* [Rud-16] [Rud-17]. Es richtet sich an Kleine und Mittlere Unternehmen (KMU), die Metallerzeugnisse herstellen. Das dem System zugrunde liegende Simulationsinstrument basiert auf Petri-Netzen. Vorbereitete Bausteine umfassen CNC-Dreh- und Fräsmaschinen, CNC-Dreh-Fräszentren, manuelle Drehmaschinen, Fräsmaschinen, Bohrmaschinen und Schleifmaschinen, Montage-Arbeitsplätze, Gleitschleifmaschinen und Prüfplätze. Modelle lassen sich hinsichtlich der Art und Anzahl von Maschinen durch den Anwender individuell aufbauen. Das Ergebnis eines Simulationslaufs ist eine Ressourcenfein- und Reihenfolgeplanung. Die Software-Umgebung besteht aus dem Tabellenkalkulationsprogramm Microsoft Excel zur Dateneingabe und -ausgabe und dem auf Petri-Netzen basierenden Simulationsinstrument PACE.

Bausteinkästen kommen meist in Simulationsinstrumenten für spezifische Anwendungsbereiche, Ebene drei, zum Einsatz. Begrifflichkeiten eines Anwendungsbereichs werden in der Regel verwendet. Vordefinierte Funktionen eines Anwendungsbereichs werden dem Nutzer übersichtlich zur Verfügung gestellt. Bausteinkästen scheinen einen Beitrag dazu zu leisten, Anfängern die Einarbeitung in die Anwendung

von Simulationsinstrumenten zu erleichtern. Geringer Aufwand für die Modellerstellung und Standardisierung sind als Vorteile von Bausteinkästen zu benennen. Bausteinorientierte Modellierungskonzepte sind im Anwendungsbereich der Produktion und Logistik weit verbreitet [Gut-17, S.70f.].

2.3.2.2 Programmiersprachen zur Implementierung von Simulationsmodellen

„Ein Programm ist eine Abfolge von Daten und Befehlen an einen Prozessor, um diese Daten in andere Daten umzuwandeln. Es wird in einer formal definierten Sprache verfasst, der Programmiersprache“ [Hen-07, S.24]. Mehrdeutigkeiten einer Programmiersprache müssen vermieden werden. Dies erfordert, die *Syntax* und *Semantik* eindeutig zu definieren. Die Syntax beschreibt den formalen Aufbau von Zeichenketten nach gewissen Regeln [VDI-3633b, S.19] [Hen-07, S.24]. Die Semantik bezeichnet die Bedeutung einzelner Sprachelemente der Syntaxdefinition [Hen-07, S.24]. Sprachelemente können Wörter oder Symbole umfassen [Hit-08, S.13].

Bei sogenannten interpretierenden Programmiersprachen werden die Quelltexte zur Laufzeit des Programms eingelesen und Befehl für Befehl abgearbeitet [Hen-07, S.29]. Im Gegensatz dazu werden bei compilierenden Programmiersprachen die Quelltexte des Programms vor der ersten Ausführung zunächst durch einen Compiler in andere Sprachen, z.B. Maschinenprogramm, übersetzt [Hen-07, S.29] [VDI-3633b, S.5]. Ein Compiler bezeichnet ein „Übersetzungsprogramm, das eine Programmiersprache A [...] in eine andere Sprache B [...] übersetzt“ [VDI-3633b, S.5].

Simulationssprachen und Programmiersprachen zur Implementierung von Simulationsmodellen sind höhere und problemorientierte Programmiersprachen, die typische Funktionen in Simulationswerkzeugen und -instrumenten, wie die Ereignisverwaltung unterstützen [VDI-3633a, S.31f.]. Sie weisen spezifische Charakteristika auf [Noc-91, S.12f.]:

- Programmiersprachen zur Implementierung von Simulationsmodellen bieten Kontrollmechanismen zur Verwaltung
 - der Ereignisliste bei ereignisdiskreten Simulationswerkzeugen,
 - der Simulationszeit und der Festlegung von Nebenläufigkeiten.
- Es lassen sich Zufallszahlen generieren.
- Der Aufbau, die Verwaltung und Aufbereitung von Statistiken werden unterstützt.

Diese Sprachen können mithilfe von *Interpretern* ausgeführt oder durch einen *Compiler* in andere Sprachen übersetzt werden [VDI-3633b, S.15]. Hersteller von Simulationswerkzeugen bieten mit ihren Produkten sehr unterschiedliche Sprachen an. Zwei Anschauungen sind verbreitet [Noc-91, S.20]:

- Einfache Programmiersprachen zur Implementierung von Simulationsmodellen bieten möglichst wenige und einfache Programmkonstrukte an. Spezielle, häufig wiederkehrende Funktionen sind enthalten.

- Programmiersprachen zur Implementierung von Simulationsmodellen, die möglichst viele zusätzliche Funktionen enthalten, die besonders auf das Anwendungsgebiet des Simulationsinstruments zugeschnitten sind.

Kenntnisse in der Programmierung sind in beiden Fällen von Nöten [Noc-91, S.20]. Ein neuerer Ansatz ist in der Literatur hinsichtlich der Modellierung mit natürlicher Sprache verfügbar. Diese Entwicklung ist bislang aber lediglich im Labor nutzbar [May-15b]. Kurz zusammenfassend gilt es festzuhalten, dass die „Implementierung von Simulationsmodellen unter Verwendung von Programmiersprachen [...] in der Regel Simulationsexperten vorbehalten“ bleibt [VDI-3633a, S.32].

2.3.2.3 Potenzial der (teil-)automatischen Modellgenerierung

Unter der (teil-)automatischen Modellgenerierung sind Ansätze zu verstehen, bei denen Simulationsmodelle durch Algorithmen mit Daten einer externen Quelle generiert werden [Str-10]. Zur automatischen datengetriebenen Modellgenerierung ereignisdiskreter Simulationsmodelle werden drei relevante Ansätze unterschieden [Ber-14] [Eck-02] [Str-10]:

Parametrische Ansätze: Modelle werden auf Basis existierender Bausteine erstellt. Diese sind in Bibliotheken gespeichert und werden vom Modellgenerator instanziiert und mit Attributen parametrisiert.

Strukturbasierte Ansätze: Daten, die die Struktur des abzubildenden Systems beschreiben, bilden in diesem Ansatz den Ausgangspunkt. Beispiele sind Fabrikstrukturdaten im Sinne der Anlagentopologie wie Layout, Fertigungsmittel, Transportfunktionen, Verkehrswege, Flächen und Restriktionen [VDI-3633a, S.34].

Hybrid-wissensbasierte Ansätze: Hierunter werden Ansätze zusammengefasst, die Verfahren der künstlichen Intelligenz, wie Expertensysteme oder neuronale Netze, mit parametrischen und strukturbasierten Ansätzen kombinieren.

MAYER UND PÖGE (2010) beschreiben die Funktionalität der (teil-)automatischen Modellgenerierung auch im Zusammenhang mit dem VDA-Bausteinkasten [May-10]. In einem weiteren Ansatz zur Simulation von Supply Chains wurde die (teil-)automatische Modellgenerierung in dem Werkzeug ICON-SimChain umgesetzt [Gut-04]. Ein weiteres Beispiel findet sich bei SNEKERS (2013). Sie beschreibt ein generisches simulationsbasiertes Werkzeug zur taktischen Entscheidungsfindung im Hinblick auf operative Prozesse [Sni-13]. HARPER (2002) benennt einen solchen Ansatz, der es Führungskräften in Krankenhäusern ermöglicht, Handlungsalternativen im Betrieb von Krankenhäusern durch Simulation zu quantifizieren [Har-02]. In seiner Dissertation zur automatischen Generierung adaptiver Modelle zur Simulation von Produktionssystemen beschreibt BERGMANN (2014) einen Ansatz zur vollautomatischen Generierung von Modellstruktur und Modellverhalten [Ber-14]. Er wählt einen hybrid-wissensbasierten Ansatz

und zeigt auf, dass der Aufwand zur Durchführung einer Simulationsstudie in der Werkstattfertigung verringert werden kann.

In den benannten Ansätzen findet die Modellierung in einer dezidierten Modellierungsumgebung statt. Daten zur Modellstruktur und Parametrisierung werden über Schnittstellen in das Simulationsinstrument geladen. Dort generiert sich das Modell dann unter Nutzung vorbereiteter Bausteine (teil-)automatisch. Eine Benutzungsschnittstelle hat das Potenzial, Anfängern komplizierte/komplexe Simulationsinstrumente in einer vereinfachten Umgebung zugänglich zu machen.

2.3.2.4 Animation und Potenzial von Simulation in der Cloud

Bei einer Cloud handelt es sich um „abstrahierte virtualisierte IT-Ressourcen [...], die von Dienstleistern verwaltet werden“ [Kag-13, S.84]. Der physische Standort der IT-Ressourcen wie Rechenkapazitäten oder Anwendungen ist für den Nutzer oft nicht erkennbar. Benutzer erhalten durch das Internet Zugang zu den IT-Ressourcen. Bezogen auf die ereignisdiskrete Simulation ermöglicht eine Cloud die mobile Verwendung einer Benutzungsoberfläche. Die für die zentrale Ablaufsteuerung zur Erzeugung und Verarbeitung von Ereignissen erforderlichen Rechenkapazitäten, können in die Cloud ausgelagert sein. Mit besonderem Fokus auf der Animation als Bestandteil der Benutzungsoberfläche sind im Folgenden mögliche Cloud-IT-Architekturen differenziert. Die Animation steht in der Betrachtung im Vordergrund, da diese in einer Cloud-IT-Architektur schnell zum Engpass hinsichtlich des Datenaustauschs zwischen IT-Ressourcen werden kann.

Eine Animation ist sehr förderlich für das Verständnis des Lernenden über das Modellverhalten. Der Begriff Animation beschreibt die Erzeugung und Präsentation von Bildfolgen, in denen Änderungen einen visuellen Effekt erzeugen [VDI-3633b, S.3]. Beispiele für solch einen visuellen Effekt sind über die Zeit variierende Positionen sowie Änderungen von Formen und Farben [Gut-17, S.25] [VDI-3633b, S.21]. RABE ET AL. (2008) benennen die Animation als Technik der Validierung und Verifikation [Rab-08, S.95ff.]. Die Animation ist somit ein wichtiger Aspekt bei der Einarbeitung in ein Simulationsinstrument.

Das Web ermöglicht ständig aktuelle Daten in kürzester Übertragungszeit bereitzustellen und gewährleistet eine nahezu universelle Verfügbarkeit von Informationen [Hit-08, S.9]. Dies kann als Treiber für die Entwicklung von Ansätzen im Bereich der Simulation in der Cloud angesehen werden. Ausgehend von einer Client-Server-Architektur kann unterschieden werden [Byr-10]:

Lokale Simulation und Visualisierung / Animation: Das Simulationswerkzeug und die Visualisierungskomponenten befinden sich auf einem lokalen Computer. Im Kontext der Online-Simulation ist der Simulatorkern z.B. als Java-Anwendung im Browser integriert. Maßgeblicher Vorteil bei diesem Ansatz ist die geringe Latenz zwischen dem GUI und Simulationswerkzeug bzw. Simulatorkern. Im Vergleich zur Simulation auf in der Regel leistungsstarken Servern kann die auf einem handelsüblichen PC oder mobilen Endgerät wie Smartphone eingeschränkte Leistung im Hinblick auf die Rechenzeit problematisch sein.

Fern-Simulation und Visualisierung / Animation: Sowohl das Simulationswerkzeug als auch jegliche Animation werden auf einem Server ausgeführt und via Client ferngesteuert. Auf Anforderung werden entsprechende Rechen-Dienste durch den Server vollzogen. Dieser Ansatz kann auch als Serviceorientierte Architektur bezeichnet werden. Das sogenannte Serviceorientierte Architektur-Paradigma beschreibt das Zusammenfassen von technischen Elementen der Anwendungsentwicklung zu höherwertigen, fachlich orientierten Komponenten [Kag-13]. Konkret kann der Nutzer z.B. Daten über eine Webseite eingeben, wobei sich die Webseite (=GUI), der Webserver, die Datenbank und das Simulationswerkzeug auf dem gleichen Server befinden. Der Ansatz verspricht eine schnelle Simulation auf leistungsfähigen Computern. Handelsübliche Simulationswerkzeuge lassen sich nutzen, wenn diese über entsprechende Schnittstellen verfügen. Herausfordernd gestaltet sich die Animation und Steuerung der Simulation in Realzeit aufgrund von Latenzen in der Datenübermittlung.

Hybride Simulation und Visualisierung / Animation: Die Simulation läuft auf einem entfernten Server. Die Verbindung zwischen Browser als Client mit dem Server wird dabei mit einer Visualisierungs- und/oder Animationssoftware gestaltet. Es findet ständiger Datenaustausch zwischen Client und Server statt. Dieser Ansatz kombiniert die Vorteile der beiden zuvor benannten Ausprägungen. Jedoch setzt dieser eine ständige Datenverbindung zwischen dem Client und Server voraus. Dies kann insbesondere bei der Nutzung von mobilen Endgeräten als Client problematisch sein, da die Datenrate und -menge begrenzt sein kann. Darüber hinaus können Latenzen in der Datenübermittlung die Dauer der Berechnung von Simulationsergebnisdaten stark erhöhen.

In der Literatur existiert keine einheitliche Definition für den Begriff Cloud-Simulation. Die Begriffsvielfalt in der Literatur wie Online- oder Web-Simulation unterstreicht diese Aussage [Byr-10]. Das Verständnis in dieser Arbeit ist eine räumlich verteilte und skalierbare IT-Architektur zum Zweck der Simulation.

Abschließend sei erwähnt, dass in der Literatur keine Ansätze identifiziert werden konnten, in denen ein Simulationsinstrument inklusive Simulatorkern auf einem mobilen Endgerät betrieben wird. Es ist anzunehmen, dass die Speicherkapazität und Rechenleistung mobiler Endgeräte derzeit kaum ausreichend sind, um ereignisdiskrete Simulationen darauf auszuführen. Insofern eignen sich diese Geräte gegenwärtig lediglich als GUI zur Dateneingabe und Ergebnisvisualisierung.

2.4 Erhöhung der Lehr- und Lernproduktivität

In den folgenden Abschnitten sind Grundlagen, Begriffe und aktuelle Entwicklungen in den Themenbereichen des Designs von Instruktionen, der Erhöhung der Motivation durch Spiele, des Lernens und der Lernfabriken herausgearbeitet. Das Potenzial von blockbasierten Programmiersprachen ist schließlich benannt.

2.4.1 Instruktionsdesign

Bei dem Instruktionsdesign handelt es sich um eine Disziplin in der erforscht und gelehrt wird, wie „Lernangebote bzw. Lernumgebungen auf der Grundlage empirisch fundierter Theorien und Befunde systematisch konzipiert werden sollen, wenn bestimmte Bildungsziele zu erreichen sind“ [Nie-13]. Als Begründer gilt ROBERT MILLS GAGNÉ [Nie-13]. Dieser definiert die Instruktion als eine Art Mittel, das von Lehrern, Designern von Ausbildungsmaterialien, Curriculum-Spezialisten und anderen, die strukturierte Ausbildungspläne entwerfen, angewendet wird, um das Lernen zu fördern [Gag-79, S.19]. In der Literatur finden sich mehrere ähnliche Modelle des Instruktionsdesigns. Eingehende Diskussionen hierzu finden sich in [Nie-13], [Pal-17] oder [Sch-09]. Aufgrund der weiten Verbreitung des sogenannten ADDIE-Modells, steht dieses im Fokus dieser Arbeit.

Das Analyze, Design, Development, Implementation, Evaluation (ADDIE)-Modell wurde im Jahr 1975 vor dem Hintergrund militärischer Ausbildungszwecke entwickelt. Die fünf Phasen sind in der folgenden Tabelle 2.2 nach BRANSON ET AL. (1975) kurz spezifiziert [Bra-75].

Tabelle 2.2: ADDIE-Modell nach [Bra-75]

		Beschreibung
Phase des ADDIE-Modells	Analysis	Analysiere die Aufgabe, wähle Aufgaben/Funktionen, entwickle Performance-Messgrößen, analysiere bestehende Kurse, wähle eine Umgebung.
	Design	Entwickle Ziele und Tests, beschreibe das Eingangsverhalten, lege Sequenz und Struktur der Lehre fest.
	Development	Spezifiziere Lern-Ereignisse/-Aktivitäten, spezifiziere einen Instruktionsmanagementplan, begutachte und wähle existierendes Material aus, entwickle die Instruktion, validiere die Instruktion.
	Implementation	Implementiere den zuvor entwickelten Plan, führe die Lehre durch.
	Evaluation	Führe eine interne Begutachtung durch, führe eine externe Begutachtung durch, vollziehe Änderungen am System.

2.4.2 Gamification und Serious Games

Die Spieleentwickler DETERDING ET AL. (2011) unterscheiden die Ausprägungen des Spielens anhand der vier Kriterien *Playing*, *Gaming*, *Whole* und *Parts*, vgl. Abbildung 2.4 [Det-11]. Die Wissenschaftler beschreiben, dass die Unterscheidung zwischen *Playing* und *Gaming* auf das Konzept von *paidia* und *ludus* der Soziologen CAILLOIS UND BARASH (1958) zurückzuführen ist. Sie beschreiben *paidia* und *ludus* als Gegensätze [Cai-01, S.13]. Während *paidia* eine Art unkontrollierte Phantasie im Sinne von freier Improvisation und sorgloser Heiterkeit beschreibt, bezeichnen die Autoren *ludus* als Spiel, das durch Regeln und Streben nach Zielen bestimmt ist. Folglich steht beim *Playing* die Unterhaltung des Nutzers im Vordergrund. Eine eher als frei anzusehende Gestaltung des Spiels im Sinne eines begrenzten bzw. kaum vorhandenen Regelwerks wird zumeist damit verbunden. Beim *Gaming* steht ein ernsthafter Zweck im Sinne von klar definierten Zielen im Vordergrund. Ein strukturiertes Regelwerk stellt dies sicher. *Whole* bedeutet, dass das Spiel-System als geschlossen zu betrachten ist. Bei *Part* werden

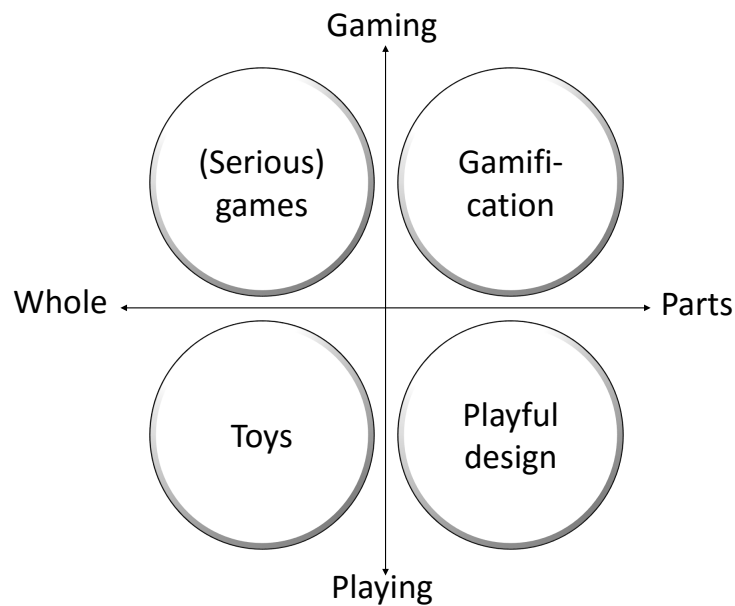


Abbildung 2.4: Abgrenzung „Gaming“ vs. „Playing“ nach [Det-11]

einzelne Spielelemente, die Spaß erzeugen sollen, in einem anderen Kontext angewendet. Während das Playing eher spontan stattfindet, ist der Start beim Gaming in der Regel geplant. Im Gegensatz zum Playing ist das Gaming durch Regeln, Ziele, nicht gemeint ist das Ziel der Unterhaltung, und einen eher strukturierten Ablauf geprägt. Die Ergebnisse beim Gaming können direkten Bezug zum realen Leben des Spielenden haben [Her-14, S.43]. Im Rahmen dieser Arbeit ist ausschließlich das Gaming relevant. Die deutschen Worte spielen und Spiel werden im Folgenden in diesem Sinn interpretiert.

Der Spiel-Experte ABT (1970) definiert den Begriff „Serious game“ als Spiel mit explizitem und sorgfältig durchdachtem Bildungsziel [Abt-70]. Diese Art der Spiele dient nicht in erster Linie dem Vergnügen. Dies bedeutet nicht, dass Serious games nicht unterhaltsam sind und sein sollen. Die wissenschaftliche Diskussion zeigt, dass in jüngerer Vergangenheit Videospiele mit Serious games verbunden werden [Dja-11]. Dies schließt die Nutzung auf mobilen Endgeräten im Sinne von Apps mit ein. Serious games finden breite Anwendung in unterschiedlichen Lebensbereichen, z.B. Gesundheit, Staat, Verteidigung, Marketing, Ausbildung und Erziehung [Zyd-05]. Ein online-Klassifizierungssystem von Video-Serious games unterscheidet existierende Spiele anhand erzieherischer und informativer Botschaften, dem Zweck des Marketings, des Trainings und des Handels [Alv-19]. Beispiele für Serious games sind Flug- oder medizinische Spiele mit dem Ziel, Piloten bzw. Ärzte zu trainieren [Boe-19].

Als eine Untergruppe von Serious games sind Alternate Reality Games zu benennen [Mül-15]. Das Ziel von Alternate Reality Games besteht darin, eine Geschichte zu erzeugen, die in das reale Leben eindringt [Kim-09]. Die Grenzen zwischen Realität und Geschichte können dabei miteinander verschmelzen, sodass Spieler vergessen können, dass sie spielen. Die vielfach zitierte Spiele-Forscherin MCGONIAL (2011) beschreibt ein Alternate Reality Game als ein im realen Leben - und nicht in einer virtuellen Umgebung - stattfindendes Spiel [McG-11, S.120]. Im Sinne eines erweiterten digitalen Spiels müssen beispielsweise Telefonanrufe getätigt oder an einen Ort gereist werden, um gemeinsam mit anderen Spielern kollektiv ein Problem zu lösen.

Das Wort „Gameification“ wurde das erste Mal schriftlich im Jahr 2008 in einem Blog genutzt. Weitere Blogger haben den Begriff daraufhin aufgegriffen und um das „e“ zu „Gamification“ gekürzt [Zac-18]. Der für die akademische Welt erste Aufsatz [Pan-17], mit der am häufigsten zitierten Definition des Begriffs „Gamification“ [Zac-15] stammt von DETERDING ET AL. (2011). Sie beschreiben Gamification als die Anwendung von Spiel-Design-Elementen in einem nicht-spielerischen Kontext [Det-11]. Solche Spiel-Design-Elemente, im Folgenden kurz als Spielelemente bezeichnet, können beispielsweise Level, Punkte, oder Abzeichen sein. Level repräsentieren kurze abgeschlossene Lerneinheiten. Ein Level umfasst eine durch den Lernenden zu lösende Aufgabe. Punkte dokumentieren sichtbar für eine Gruppe an Spielern den Fortschritt des einzelnen Spielers. Abzeichen machen Verdienste des Spielers für den Spieler und für Mitspieler sichtbar. Bei offenkundigen Abzeichen sind dem Spieler die Kriterien zur Verleihung bekannt. Im Gegensatz hierzu stehen die nicht offenkundigen Abzeichen. Als Beispiele für einen nicht-spielerischen Kontext benennen DETERDING ET AL. (2011) allgemein die Bereiche Ausbildung, Gesundheit und Nachrichten. Gamification zielt darauf ab, Handlungen von Menschen durch Motivation zu beeinflussen. Alle Spiele vereinen folgende Eigenschaften bzw. Bestandteile [McG-11, S.21]:

Ziel: Unter einem Ziel wird ein spezifisches Ergebnis verstanden, auf dessen Erreichung die Spieler hinarbeiten. Dies fordert deren Aufmerksamkeit und lenkt ihre Teilnahme während des Spiels. Das Ziel verschafft den Spielern ein Gefühl von Sinnhaftigkeit.

Regeln: Regeln limitieren, wie Spieler das Ziel erreichen können. Regeln fördern die Kreativität und das strategische Denken.

Feedback: Durch ein Feedback wissen Spieler, wie nahe sie dem Ziel sind. Feedback kann z.B. durch die Vergabe von Punkte oder Abzeichen an den Spieler gegeben werden. In der einfachsten Form kann das Feedback so einfach wie das Wissen des Spielers über ein objektives Ergebnis sein: „Das Spiel ist vorbei, wenn...“. Ein Live-Feedback empfindet der Spieler als ein Versprechen darüber, dass das Ziel definitiv erreichbar ist und motiviert ihn, weiter zu spielen.

Freiwillige Teilnahme: Die Freiheiten, am Spiel teilzunehmen und es zu beenden stellen sicher, dass stressige und herausfordernde Aktivitäten angenehm wahrgenommen werden.

In der Literatur besteht Konsens darüber, dass Serious games und Gamification die studentische Motivation erhöhen, Gruppenarbeiten positiv beeinflussen, Kommunikationsfähigkeiten trainieren und ein Experimentieren in einer sicheren Umgebung ermöglichen [Mül-15]. Ein Indikator für die weite Verbreitung sind die über 3.300 Spiele auf der Plattform <http://serious.gameclassification.com/> [Alv-19].

Studien von BARATA ET AL., in denen Gamification in Lehrveranstaltungen an Universitäten evaluiert wurde, zeigen eine erhöhte Präsenz Studierender in Lehrveranstaltungen sowie ein proaktives und engagiertes Verhalten [Bar-13b] [Bar-13a] [Mül-15]. Die Autoren nutzen Punkte und Level dazu, Feedback an die Lernenden zu geben. Leaderboards, also Ranglisten in denen Studierende gemäß ihren individuellen Punkten absteigend sortiert sind und Abzeichen werden als Spielelemente verwendet. Die Autoren deuten an, dass die Anzahl regelmäßig teilnehmender Studierender, die Häufigkeit des Downloads der Lehrunterlagen und die Anzahl der Beiträge im digitalen Kurs-Forum anstiegen. Studierende

reagierten häufiger auf Beiträge im Kurs-Forum und initiierten achtmal so viele Diskussionsgruppen. Studierende geben an, höhere Motivation und größeres Interesse als in anderen Lehrveranstaltungen zu empfinden.

COLLER UND SHERNOFF (2009) bewerteten in ihrer Studie das Engagement von Studierenden, nachdem sie das Curriculum eines Bachelorstudiengangs zum Maschinenbau umstrukturierten [Col-09] [Mül-15]. Aufgaben und Lern-Erfahrungen wurden mit Video-/bzw. Computerspielen gestaltet. Studierende sollten z.B. Computerprogramme schreiben, in welchen ein simuliertes Auto auf einer Fahrstrecke zu bewegen ist. Die Autoren stellten fest, dass die Studierenden mehr Konzentration, Genuss und Interesse an der Lösung von Aufgaben zeigten. Dies ist auf klare Regeln und unmittelbares Feedback zurückzuführen.

Es kann festgehalten werden, dass Spiele die Lehr- und Lernproduktivität von Studierenden durch Motivation positiv beeinflussen können. Die in der Literatur beschriebenen Ansätze zeigen die Anwendbarkeit von Spielen in Lehrveranstaltungen an Hochschulen und beziehen sich auf Studentengruppen. Im Folgenden ist relevante psychologische und soziologische Literatur mit Blick auf die Lernförderlichkeit und Motivation dokumentiert. Es lassen sich sogenannte Spieler-Typen differenzieren, die unterschiedlich auf Spielelemente reagieren. Diese sind im Folgenden neben einem beispielhaften Spielentstehungsprozess diskutiert.

2.4.2.1 Spieler-Typen

Aufgrund individueller menschlicher Persönlichkeitsmerkmale und Werte besteht eine besondere Herausforderung in der Gestaltung von Spielen. Spiele sollen vom Spieler als motivierend und vergnüglich wahrgenommen werden. Vor diesem Hintergrund sind in der Literatur sogenannte Spieler-Typen unterschieden [Ste-11]. Umfangreich zitiert, diskutiert und zum Teil um weitere Spieler-Typen erweitert ist die Taxonomie von BARTLE (1996). Diese soll für diese Arbeit genutzt werden. Er teilt Menschen in die vier Spieler-Typen Sammler, Entdecker, Socializer und Zerstörer ein [Bar-96]:

Sammler: Bei den Sammlern ordnet sich alles dem Punktesammeln und Level-Aufstieg unter. Entdeckungen dienen dazu, neue Wege zum Erlangen von mehr Punkten zu beschreiten, oder dazu, irgendwo ein paar verbleibende Punkte zu ergattern. Socializen wird als entspannende Möglichkeit wahrgenommen, Mitspieler hinsichtlich ihres Wissens über das Punktesammeln auszuspähen. Zerstören ist nur erforderlich, um Spieler zu eliminieren, die im Weg stehen.

Entdecker: Die Entdecker genießen es, abgelegene Orte oder interessante Eigenschaften wie Bugs zu entdecken. Punktesammeln kann notwendig sein, um die nächste Phase der Entdeckung zu ermöglichen. Dies wird allerdings als ermüdend empfunden. Sie denken, dass jeder mit etwas Verstand dies tun könnte. Zerstören kann in sich selbst eine konstruktive Übung sein, wird allerdings als langweilig und bedrängend empfunden, sobald der Eliminierte langfristig nach Vergeltung strebt. Socializen kann als informativ wahrgenommen werden, allerdings wird oft das von Mitspielern wiedergegebene als irrelevant oder bereits veraltet angesehen.

Socializer: Bei Socializern dient das Spiel eher als Hintergrund. Im Vordergrund steht die gemeinsame Plattform, auf der etwas mit Spielern passiert. Die Beziehungen zwischen Spielern sind wichtig: Empathie, Sympathie, Witzelei, Unterhaltung, Zuhören sind von Bedeutung. So steht das Kennenlernen von Menschen, das Verständnis und die Generierung langanhaltender Beziehungen zu ihnen im Vordergrund. Ein Entdecken kann im Spiel notwendig sein, um zu verstehen, worüber die Mitspieler sprechen. Punktesammeln kann von Bedeutung sein, um auf den spezifischen Sprachgebrauch höherer Level Zugriff zu haben, oder um einen gewissen Status in einer Gemeinschaft zu erlangen. Das Zerstören ist nur zu entschuldigen, wenn es sich um ein Versehen oder Rache für eine nicht zu tolerierende Tat einem Freund gegenüber handelt.

Zerstörer: Die Zerstörer werden durch gegenseitiges Überbieten motiviert. Dazu zählt auch das Attakieren der anderen Spieler, um diese zu vernichten. Je massiver ein Spieler andere Spieler bedrängt, desto größere Freude empfindet er. Punktesammeln ist in der Regel erforderlich, um so mächtig zu sein, Verwüstung anzurichten. Entdeckungen können erforderlich sein, um neue Möglichkeiten aufzutun, die Mitspieler ausgeklügelt zu vernichten. Socializen kann sinnvoll sein, um beispielsweise Gewohnheiten von Mitspielern ausfindig zu machen oder Strategien mit Zerstörer-Gefährten abzusprechen.

Nach ZICHERMANN UND CUNNINGHAM (2011) ist ein Mensch nicht ausschließlich einem Spieler-Typen zuzuordnen. Vielmehr weist jeder Mensch mehr oder weniger stark ausgeprägte Eigenschaften jedes Spieler-Typs auf [Zic-11, S.23f.]. Folglich sollten Spielelemente durch den Spieleentwickler so gestaltet werden, dass alle Spieler-Typen sich angesprochen fühlen. Diese Erkenntnis soll auf die zu entwickelnde Lernumgebung für ereignisdiskrete Simulation übertragen werden, deren Nutzung durch Studierendengruppen erfolgt.

2.4.2.2 Motivation und Flow

Spiele haben auf Menschen eine motivierende Wirkung. Motivation kann grundsätzlich in Amotivation, intrinsisch und extrinsisch unterteilt werden. Amotivation beschreibt einen Zustand, bei dem eine Tat absichtlich unterlassen wird. Beispielsweise kann dies auftreten, wenn sich eine Person als nicht kompetent genug zur Durchführung einer Aktivität einschätzt. Intrinsische Motivation ist definiert als das Durchführen einer Aktivität zur inhärenten Befriedigung aus Interesse oder Genuss.

Nach der von DECI UND RYAN geprägten Selbstbestimmungstheorie begründet sich die intrinsische Motivation durch die Befriedigung der psychologischen Grundbedürfnisse nach Kompetenz, Autonomie und Verbundenheit. Extrinsisch motiviertes Verhalten basiert auf einem trennbaren Ergebnis wie einer Belohnung oder der Vermeidung einer Strafe [Rya-00].

Die extrinsische Motivation lässt sich unterteilen. Abgestuft nach dem Grad der äußeren Beeinflussung kann in den extremen Ausprägungen die sogenannte externe Regulierung von der Integration abgegrenzt werden [Rya-00]. Die externe Regulierung beschreibt eine Ausprägung der extrinsischen Motivation, bei

der Menschen aufgrund von Belohnungen oder Strafen von außen handeln. Dies tritt beispielsweise auf, wenn ein Studierender Hausarbeiten erledigt, weil dieser Sanktionen wie eine schlechte Note durch den Lehrenden fürchtet. Die Integration beschreibt die unabhängigste Form der extrinsischen Motivation. Regeln, Werte und/oder Normen wurden verinnerlicht und motivieren zum Handeln. Ein Beispiel für diese Art der extrinsischen Motivation ist ein Studierender, der die Hausarbeiten bearbeitet, da er daran glaubt, dass diese Tätigkeit einen wichtigen Beitrag für die spätere Karriere darstellt.

Insofern ein Mensch intrinsische Motivation zur Durchführung einer Aktivität verspürt, ist es nicht erforderlich, extrinsische Anreize zu setzen. Übertragen auf die Einarbeitung in ereignisdiskrete Simulation ist klar, dass die Menschen, die intrinsische Motivation verspüren, sich mit der Simulation zu beschäftigen, dies auch ohne äußere Anreize tun werden.

Unterstützt durch Spielelemente kann ein sogenannter Flow erlebbar sein. Im Jahr 1975 veröffentlichte CSIKSZENTMIHALY erstmals seine Flow-Theorie [Csi-14b]. „Flow“ beschreibt das menschliche Empfinden von Konzentration, geringer Selbsterfahrung und Genuss. Dieser Zustand tritt typischerweise während herausfordernden Aktivitäten ein, die im Hinblick auf den Schwierigkeitsgrad auf die Fähigkeiten der individuellen Person angepasst sind [Ull-12]. METZ UND THEIS (2011) beschreiben den Flow, also das *Fließen* oder *Strömen*, als ein Gefühl des völligen Aufgehens in einer Tätigkeit oder gar als Tätigkeitsrausch bzw. Funktionslust [Met-11, S.65]. Als konkretes Beispiel benennt RHEINBERG (2004) den „Computerfreak, der erst am schmerzenden Rücken mitten in der Nacht merkt, daß er schon wieder viele Stunden am Rechner zugebracht und dabei das Essen sowie einen Termin vergessen hat“ [Rhe-04]. Der Zusammenhang zwischen Herausforderung an den Spieler und Fähigkeiten des Spielers im Hinblick auf den Flow ist schematisch in Abbildung 2.5 dargestellt [Csi-14a, S.247ff.].

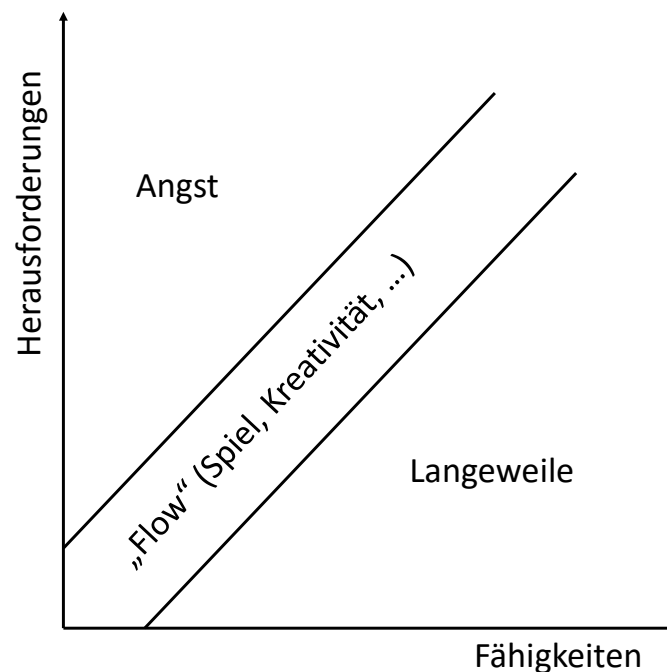


Abbildung 2.5: Flow-Theorie nach [Csi-14a, S.247ff.]: Flow wird empfunden, wenn Schwierigkeitsgrad und Fähigkeiten der individuellen Person im Einklang sind

RHEINBERG (2004) kommt nach einer Untersuchung mehrerer Studien zum Ergebnis, dass eine positive Korrelation zwischen dem Flow-Erlebnis und der sogenannten Lernleistung besteht [Rhe-04, S.154ff.]. METZ UND THEIS (2011) benennen konkret eine positive Korrelation zwischen dem Flow-Erlebnis und dem Lernerfolg mit Serious games [Met-11, S.65].

2.4.2.3 Spielentstehung

Der Gamification-Assistenzprofessor MANRIQUE (2019) benennt sechs Schritte, um im Sinne des Spielentstehungsprozesses gamifizierte Erfahrungen zu generieren [Man-19] [Mül-15]:

- Zusammenstellung des Teams der Spieleentwickler. Das Problem bzw. die Zielstellung ganz klar formulieren.
- Visualisierung des Warum, Was und Wer: Zielgruppe analysieren und definieren, wie die gamifizierte Erfahrung diese beeinflussen soll.
- Entwurf des Themas und einer Geschichte.
- Spielelemente wählen, z.B. Level, Geschenke, Avatare, Teams und Spielmechaniken⁷ gestalten.
- Grafik verbessern, da diese Aufmerksamkeit erregen soll.
- Wiederholte Spiel-Tests.

Nach MANRIQUE (2019) gilt es vier Phasen im Spielentstehungsprozess aus Sicht des Spielers zu unterscheiden [Man-19]:

Entdeckung: Das Ziel besteht zunächst darin, die Aufmerksamkeit von potenziellen Nutzern zu gewinnen. Dies kann beispielsweise durch eine ansprechend gestaltete Benutzeroberfläche realisiert sein. Menschen sollen in sozialen Medien oder in Gesellschaften darüber sprechen. Derjenige, der das Spiel nicht kennt, soll Neugier verspüren.

Onboarding: Das Onboarding dient dazu, den Spieler mit dem Spiel vertraut zu machen. Durch einen klar definierten Lernpfad soll der Spieler die Grundlagen zur Bedienung des Spiels erlernen. Dies kann z.B. durch Tutorials oder Erklär-Videos passieren. MANRIQUE schlägt vor, für jede korrekte Aktion eine Belohnung, z.B. in Form von Punkten, zu vergeben.

Midgame: Im Midgame findet das eigentliche Spiel statt. Die Herausforderungen an den Spieler steigen mit seinen Fähigkeiten. Neue Spielmechaniken werden eingeführt.

Endgame: Das Endgame stellt den letzten Teil eines Spiels dar. Besonderes Augenmerk liegt auf der verstärkten Integration sozialer Spielelemente wie die Bildung von Gruppen. Auch ist vorstellbar, dass erfahrene Nutzer in der Rolle als Lehrer Anfänger unterstützen. Der Spieler soll eine sogenannte unvergängliche Erfahrung machen.

⁷Eine einheitliche Definition für den Begriff *Spielmechanik* existiert nicht, vgl. die Diskussion in [Poj-18]. In der vorliegenden Arbeit ist unter einer Spielmechanik das Zusammenspiel von Spielelementen bezeichnet.

Digitale Spiele werden mit sogenannten Game-Engines erstellt. Es handelt sich um „lizenzierbare Softwarepakete, die die Infrastruktur des Spiels bereitstellen und der Echtzeitberechnung von digitalen 3-D-Umgebungen dienen“ [Sch-18, S.119] [Gre-09, S.3ff.]. Game-Engines sind erweiterbar und dienen der Entwicklung unterschiedlicher Spiele. Sie sind hinsichtlich ihrer grafischen Eigenschaften als sehr leistungsfähige Programme zur Erstellung digitaler Spiele einzuschätzen.

2.4.2.4 Gamification im Fabrikbetrieb

In jüngster Zeit werden vermehrt Aufsätze veröffentlicht, in denen konkrete Beispiele zum Einsatz von Serious Games und Gamification im Kontext der Ausbildung zum Fabrikbetrieb beschrieben sind.

Das *Business Game for Total Life Cycle Management* wurde an der Technischen Universität Braunschweig entwickelt [Böh-11]. In Teams unterteilte Studierende repräsentieren konkurrierende Firmen der Automobilindustrie. Die Teams bewerben sich im Spiel um Fördermittel. Mit sogenannten Strategiekarten können Spieler ihre Verbesserungen vor dem Hintergrund des erwarteten ökologischen Einflusses beschreiben. Wichtige Key Performance Indicators (KPIs)⁸ hinsichtlich der Erfolgsmessung sind z.B. die Öko-Effizienz, der Gewinn und die Kundenattraktivität. Microsoft Excel kommt zum Einsatz, um die jeweiligen Werte zu errechnen. Das Spiel kann in ca. sieben bis acht Stunden abgeschlossen werden.

Ein ähnlicher Ansatz wird mit dem *Resource Efficiency Learning Game - Electric Scooter Game* der TU Berlin verfolgt [Rei-14]. Dies beachtet verstärkt eine internationale Sichtweise. Die Spieler verfolgen das Ziel, eine hypothetische Elektroroller-Fabrik hinsichtlich ökonomischer, ökologischer und sozialer Kriterien zu verbessern. Studierende erarbeiten in Kleingruppen sogenannte Investment-Proposals. Hier benennen sie, welche Anpassungen beabsichtigt sind und wie die an den Dimensionen der Nachhaltigkeit orientierten KPIs hierdurch voraussichtlich beeinflusst werden. Der Schwerpunkt der Maßnahmen liegt in der Netzwerkgestaltung. Produktions- und Vertriebsstandorte können in Entwicklungs- und Schwellenländern als auch in industrialisierten Ländern liegen.

KEIL UND SCHMIDT (2007) beschreiben den Einsatz eines sogenannten Simulationsspiels [Kei-07]. Ausgangspunkt ist ein Simulationsmodell, welches Material- als auch Energieflüsse abbildet. Die Parameter werden nicht von den Studierenden geändert. Sie wählen vielmehr aus einem Katalog möglicher Handlungsalternativen. Die Ergebnisse der Simulationsläufe erhalten sie in Papierform. Dem Simulationsspiel liegt die Geschichte des fiktiven mittelständischen Automobilzulieferers „PMG Schwäbele“ zugrunde, dessen Effizienz es zu steigern gilt.

Im Kontext sogenannter Learnstruments zeigen MÜLLER, REISE, DUC UND SELIGER (2016) den Einsatz eines Simulationsspiels an einem manuellen Montagearbeitsplatz [Mül-16b]. Learnstruments sind tangible als auch intangible Artefakte, die dem Lernenden die Funktionalität automatisch vermitteln können [Pos-11] [Men-18]. Simulationsspiele kommen dabei an dem sogenannten Smart-Assembly-Workplace

⁸Bei einem KPI handelt es sich um eine wichtige Kennzahl, anhand derer die Erfüllung der Aufgabe gemessen wird.

zum Einsatz, der eine markerlose Bewegungserkennung ermöglicht [Krü-15] [McF-13] [Mül-16a]. Das Spiel soll insbesondere Studierenden ein initiales Verständnis hinsichtlich der manuellen Montage geben. Am Arbeitsplatz wird dem Lernenden eine Geschichte erzählt, um ihn unter motivierenden Aspekten zu befähigen, einen Elektromotor für ein Fahrrad zu montieren. Es handelt sich um einen Ansatz unter Nutzung von Augmented Reality (AR).

POTENTE ET AL. (2013) beschreiben den Einsatz von Gamification im Kontext ihrer Software *OptiWo* [Pot-13]. Die Software ist dazu bestimmt, dass Manager unterschiedliche Produktionsnetz-Konfigurationen studieren und bewerten können. Die Autoren haben in der Software Spielelemente gestaltet. Sie kommen zum Ergebnis, dass die Software im industriellen Einsatz von Experten als „wertvoll“ eingeschätzt wird.

GÜNTNER ET AL. (2015) berichten nach Abschluss des von der Industriellen Gemeinschaftsforschung geförderten Projekts *GameLog - Gamification in der Intralogistik* über die Erprobung des Einsatzes von Gamification zur Motivations- und Leistungssteigerung in der manuellen Mann-zu-Ware Kommissionierung [Gün-15]. Sie entwickelten ein Vorgehen zur Anwendung von Gamification bei manuellen Handhabungsprozessen. In einer Studie stellten sie fest, dass die Motivation und Leistung der Probanden signifikant höher waren.

Gamification im Fabrikbetrieb erfreut sich steigender Beliebtheit. Die Ausführungen in diesem Kapitel zeigen, dass das lern- und motivationsförderliche Potenzial von Gamification für die Industrie und Hochschullehre groß ist. Mit Blick auf die ereignisdiskrete Simulation werden im Kapitel 2.5 weitere Ansätze aufgegriffen und später auf die Entwicklungen in dieser Arbeit bezogen.

2.4.3 Lernfabriken und erfahrungsbasiertes Lernen

Mit dem Keynote Paper der Internationalen Akademie für Produktionstechnik (CIRP) *Learning Factories for future oriented research and education in manufacturing* wurde im Jahr 2017 eine anerkannte Definition des Begriffs *Lernfabrik* geschaffen [Abe-17]. Dieser Aufsatz, ergänzt um zahlreiche Beiträge der jährlich stattfindenden *Conference on Learning Factories*, formt zunehmend das Forschungsgebiet. Die Definition des Begriffs *Lernfabrik* nach der CIRP Enzyklopädie umfasst im engeren Sinne eine Lernumgebung, [Abe-14] [Abe-15]

- die authentische *Prozesse*, mehrere Stationen, technische als auch organisatorische Aspekte vereint,
- deren *Aufbau* veränderlich und einer realen Wertschöpfungskette ähnlich ist,
- in welcher ein physisches *Produkt* hergestellt wird und
- in der ein *didaktischer Ansatz* verfolgt wird, in dem formelles, informelles und non-formelles Lernen ermöglicht wird, indem Lernende Entscheidungen vor Ort treffen können.

In Abhängigkeit des *Zwecks* einer Lernfabrik findet das Lernen durch Lehren, Training und/oder Forschung statt. Folglich können als Lernergebnis Kompetenzentwicklung und/oder Innovation vorliegen. Ein langfristiger Plan für den Betrieb einer Lernfabrik ist wünschenswert.

Lernfabriken kommen in der Industrie, Lehre und Forschung zum Einsatz. Lernfabriken unterscheiden sich hinsichtlich der Aufbau- und Ablauforganisation und ihren Lernzielen zum Teil erheblich. Die Morphologie von TISCH ET AL. (2015) spannt das Feld möglicher Ausprägungen von Lernfabriken mit 59 Eigenschaften auf [Tis-15]. Die sich hieraus ergebende Kombinatorik erfordert eine sinnvolle Eingrenzung. Im Fokus der vorliegenden Arbeit stehen Lernfabriken, in denen die (De-)Montage und/oder Logistik durch die Lernenden gestaltet werden kann. Produkte werden in Massen-, Serien- oder Kleinserienproduktion durch die Lernenden hergestellt. In der Lernfabrik stehen die Gestaltungsprinzipien Ganzheitlicher Produktionssysteme als Lernziel im Vordergrund [Abe-17]. Die Lernfabrik ist hinsichtlich der Methoden des Fließprinzips und Pull-Prinzips veränderlich.

Lernfabriken sind im Kontext der Ausbildung von Ingenieuren ein Ort, an dem das *erfahrungsbasierte Lernen* stattfindet [Abe-17] [Mül-17b]. Nach KOLB (1984) wird unter *Lernen* ein Prozess verstanden, bei dem Wissen durch Transformation von Erfahrungen generiert wird [Kol-84, S.38]. „Wissen wiederum basiert auf Informationen als Daten“, wobei Daten „in einem Bedeutungskontext stehen und als Grundlage von Entscheidungen und Handlungen dienen“ [Nor-18, S.41]. Kompetenzen basieren auf Wissen, das angewendet wird [Nor-18, S.42]. Erfahrung ist „eine durch Praxis erworbene Fähigkeit sicherer Orientierung in Handlungs- und Sachzusammenhängen“ [Zwa-03].

Ein weitverbreitetes Modell, mit dem der Lern- und Problemlösungsprozess beschrieben ist, wurde von den Psychologen KOLB ET AL. (1971) entwickelt [Kol-71] [Mül-17b]. Dieses Modell ist heute als Kolbs erfahrungsbasierter Lernzyklus bekannt. Der Zyklus ist in Abbildung 2.6 abgedruckt. Durch aktives Experimentieren erlangen Lernende eine konkrete Erfahrung. Dieser Erfahrung folgen eine Beobachtung und Reflexion, die zur Bildung eines abstrakten Konzepts führt, welches wiederum in zukünftigen Experimenten zu einer Erfahrung führt. Der erfahrungsbasierte Lernzyklus wiederholt sich ständig. In

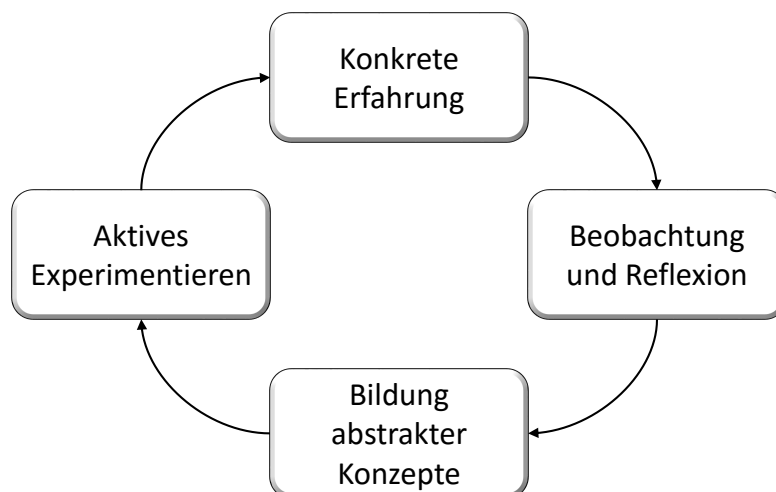


Abbildung 2.6: Erfahrungsbasierter Lernzyklus nach [Kol-71]

diesem Sinne kann Lernen als Wiedererlernen verstanden werden. Der generische Zyklus lässt sich mit Blick auf individuelle Bedürfnisse, Ziele oder Lernstile hoch individuell gestalten. KOLB ET AL. (1971) führen als Beispiel den Mathematiker an, bei dem der Schwerpunkt auf der Bildung abstrakter Konzepte liegen mag. Im Vergleich dazu benennen sie ein Beispiel eines Poeten, der womöglich mehr auf seine individuellen Erfahrungen aus der Vergangenheit baut.

Im Sinne von Kolbs Lernzyklus können Lernende durch mehrmaliges „Versuchen“ z.B. den Durchsatz in einer Lernfabrik erhöhen. Meist wird in Lernfabriken ein besonderer Fokus auf die individuellen Handlungen des Lernenden gelegt. Die Entwicklung von Kompetenzen des Lernenden steht im Vordergrund [Abe-12]. Im Sinne der Taxonomie der Lernziele nach BLOOM werden in Lernfabriken kognitive [And-01], affektive [Blo-84] und psychomotorische Lernziele verfolgt [Abe-17]. Dies ist für Ingenieure insbesondere geeignet, da sie sich in einer ihr vertrauten Umgebung befinden und ohne Sicherheits- oder Beschädigungsrisiken experimentieren können. In Lernfabriken können physische und/oder virtuelle Bestandteile integriert sein. Physische Bestandteile können Arbeitsplätze oder Rollenförderer sein. Virtuelle Bestandteile können E-Learning oder digitale Assistenzsysteme am Arbeitsplatz umfassen. Lernfabriken erfordern ein didaktisches Konzept und sollen das experimentelle und problembasierte Lernen herausstellen [Abe-15]. Der handlungsorientierte Ansatz wirkt im Vergleich zur traditionellen Instruktion z.B. mit textuellen Aufgaben oder Vorlesungsskripten intensiver auf die Lernenden.

In der Literatur sind Ansätze zu finden, die die ereignisdiskrete Simulation mit Lernfabriken in Verbindung bringen [Mül-17b]. SCHOLZ ET AL. (2016) nutzen eine sogenannte vorprogrammierte Simulationsumgebung, um den Lernenden Unterschiede im Ressourcenverbrauch zwischen unterschiedlichen, kaum näher spezifizierten, Versorgungskonzepten in der Lernfabrik aufzuzeigen [Sch-16]. BÄCK (2008) beschreibt ein simulationsbasiertes Planspiel, das zur spielerischen Verbesserung der Logistik eines fiktiven Unternehmens genutzt wird [Bäc-08]. Erprobt mit Studierenden und industriellen Teilnehmern fokussiert das Spiel die Ausbildung im SCM. PRINZ ET AL. (2016) beschreiben den Einsatz eines Simulationsmodells zur Produktionssteuerung in einer Lernfabrik [Pri-16]. Im Kontext von Industrie 4.0 soll Lernenden nahegebracht werden, wie eine sogenannte simulationsbasierte Produktionssteuerung funktioniert. Die Teilnehmer werden mit einem Simulationsinstrument, einem MES (=Manufacturing Execution System) und Agentensystemen vertraut gemacht. Lernende sollen erkennen, dass Simulation für kaum näher spezifizierte Aspekte der Produktionssteuerung genutzt werden kann [Pri-16].

In den benannten Forschungsberichten wird die Simulation als Hilfsmittel angeführt, um Lernziele in den Bereichen Produktionssteuerung und Materialfluss zu erreichen. In keinem der benannten Ansätze steht die Ausbildung von Lernenden im Hinblick auf die Durchführung von Simulationsstudien oder die Anwendung von Simulationsinstrumenten im Fokus.

2.4.4 Visuelle blockbasierte Programmiersprache

Als eine Alternative zur traditionellen textbasierten Programmierung setzen sich bei Anfängern in der Programmierung vermehrt visuelle blockbasierte Programmierwerkzeuge durch. Programmieranfänger

kombinieren vorbereitete Blöcke durch Ziehen und Ablegen, sogenanntes Instanzieren, und erhalten somit ausführbare Programme. Die vorbereiteten Blöcke sind für den Lernenden in einer Block-Bibliothek hinterlegt. Instanziiert werden die Blöcke durch den Nutzer im Block-Editor. Blöcke stellen Programm-elemente wie Funktionen oder Variablen dar [Kur-15]. Die Besonderheit besteht darin, dass die Blöcke visuelle Merkmale aufweisen. So weiß der Benutzer intuitiv anhand von Farben und Formen, wo ein gewisser Block verwendet werden kann. In der Regel umfassen die existierenden visuellen blockbasierten Programmiersprachen Blöcke, die Bedingungen, Schleifen, die Deklaration von Variablen und besondere vordefinierte Funktionen abbilden können.

Ein Beispiel für ein visuelles blockbasiertes Programmierwerkzeug ist Scratch, ein Projekt der Lifelong-Kindergarten-Group am Media-Lab des Massachusetts Institute of Technology [Scr-19]. Es handelt sich um eine visuelle blockbasierte Programmiersprache, die insbesondere für acht- bis 16-Jährige konzipiert ist. Es lassen sich Blöcke durch Ziehen und Ablegen aneinanderfügen, um einfache Spiele oder Animationen zu erstellen. Ein Beispiel für diese Programmiersprache ist in Abbildung 2.7 abgedruckt [Wei-15].



Abbildung 2.7: Beispiel der visuellen blockbasierten Programmiersprache Scratch nach [Wei-15]

Bei Scratch handelt es sich neben der Programmierumgebung auch um eine Community [Res-09] [Mül-17a]. Sogenannte „Scratcher“ können untereinander Projekte austauschen. Das Ziel von Scratch besteht nach eigenen Angaben in der Erziehung einer neuen Generation von kreativen und systematischen Denkern, die in der Lage sind, die Programmierung zu nutzen, um ihre Ideen auszudrücken. Die Ausbildung professioneller Programmierer ist ausdrücklich nicht das Ziel. Scratch ist die in der Literatur meistdiskutierte visuelle blockbasierte Programmiersprache. Neben Scratch existieren inzwischen weitere wie Google Blockly [Goo-19b]. Mit Blockly liefert Google ein plattformübergreifendes Werkzeug mit vorbereiteten Blöcken. Nutzer können den Programmcode durch Ziehen und Ablegen von Blöcken aufbauen. Als Ergebnis erhält der Nutzer nach einer Kompilierung ein syntaktisch korrektes Programm in einer gewünschten textbasierten Programmiersprache. Diese sind auf JavaScript, Python, PHP, LUA und Dart beschränkt. Nutzer können auch individuelle Blöcke erstellen.

KURIHARA ET AL. (2015) stellen fest, dass die Nutzung von visuellen blockbasierten Programmiersprachen maßgeblich für Programmieranfänger ein großer Vorteil sein kann, da Nutzer bei der Syntax und Semantik unterstützt werden [Kur-15]. Kennt ein Lernender die Syntax einer Programmiersprache nicht, kann sich schnell Frustration einstellen, wenn beispielsweise Punkte oder Klammern vergessen werden. Versteht er die Semantik nicht, kann er das Verhalten von Funktionen nicht nachvollziehen und kaum einschätzen, wann er welche Funktion nutzen kann. Blöcke können natürliche Sprache als ein Beschreibungsmittel und Code-Schnipsel enthalten. Beispielsweise folgt auf ein *if* eine Bedingung und darauf das Schlüsselwort *then*.

Ein Einsatzgebiet von visueller blockbasierter Programmiersprache sind Schulen, um Schüler in die Programmierung einzuführen. WEINTROP UND WILENSKY (2015) kommen in einer Studie über die Unterschiede zwischen textbasierter Programmierung und visueller blockbasierter Programmierung zum Ergebnis, dass die visuelle blockbasierte Programmierung einfacher als die textbasierte Programmierung anzuwenden ist. Sie beschreiben, dass diese Erkenntnis jedoch vorwiegend für den Lernprozess im Anfangsstadium zutreffend ist. Im Vergleich mit der textbasierten Programmierung müssen ein eingeschränkter Funktionsumfang, mehr Zeit für den Aufbau von Programmen und fehlende Authentizität im Sinne begrenzter Anwendbarkeit für gewisse fachliche Fragestellungen in Kauf genommen werden [Wei-15]. Die Lernförderlichkeit der visuellen blockbasierten Programmiersprache und die gesteigerte Motivation von Schülern ist auch in weiteren Studien als Ergebnis benannt [Oua-15] [São-16] [Wil-10]. KALELIOĞLU (2015) kommt zu der Erkenntnis, dass Grundschüler durch die Nutzung solcher Programmierwerkzeuge eine positive Einstellung im Hinblick auf die Programmierung entwickeln [Kal-15].

Kurz zusammengefasst tragen visuelle blockbasierte Programmiersprachen dazu bei, Anfänger in Programmiersprachen einzuarbeiten. Übertragen auf die ereignisdiskrete Simulation ergibt sich das Potenzial zur Einarbeitung in Programmiersprache zur Implementierung von Simulationsmodellen. Mit einer visuellen blockbasierten Programmiersprache zur Implementierung von Simulationsmodellen scheinen zügig korrekte Programme oder Programmteile von Lernenden mit geringem Vorwissen aufgebaut werden zu können. Eine solche Programmiersprache kann durch vorgegebene Schlüsselwörter, die in der richtigen Reihenfolge stehen, einen Beitrag zu einem verifizierten Simulationsmodell leisten [VDI-3633b, S.19ff.].

2.5 Vorarbeiten zur Einarbeitung in ereignisdiskrete Simulation

Die Fachgruppe Edukation und Simulation der Arbeitsgemeinschaft Simulation (ASIM) unterscheidet zwischen der Simulation in/für Ausbildung (SIM in EDU) und der Ausbildung im Modellaufbau und der Simulation (EDU in SIM) [ASI-19] [Mül-17b]. SIM in EDU ist als die Nutzung von Simulation zur Ausbildung in einem anderen Bereich, z.B. via E-Learning und/oder Blended Learning, zu verstehen. EDU in SIM fokussiert die Ausbildung im Modellaufbau und der Simulation.

Ansätze der *SIM in EDU* umfassen beispielsweise Planspiele unter Nutzung von Simulation. ZÜLCH UND STOHWASSER (2006) berichten in diesem Kontext über die Reorganisation von Produktionssystemen [Zül-06]. MUSTAFEE UND KATSALIAKI (2010) benennen ein konkretes Beispiel zur Ausbildung im SCM [Mus-10]. ROEDER UND MIYAOKA (2015) verwenden an der Universität San Francisco in Management-Kursen das Simulationsinstrument *Littlefield Technologies* [Roe-15]. Veranstaltungsbegleitend kommen zwei Spiele zum Einsatz: *Capacity Management Game* und *Managing Customer Responsiveness*. Die Studierenden verbessern in einer ereignisdiskreten Simulation vor dem Hintergrund eines gegebenen Zielsystems die Profitabilität einer virtuellen Firma [Roe-15]. WOOD (2007) benennt im Rahmen seiner Analyse von Modulbeschreibungen management-orientierter Kurse online-Versionen des vom Massachusetts Institute of Technology entwickelten Bierspiels (Beer Game) zur Simulation des Peitscheneffekts (Bullwhip-Effekt). Dieses Spiel wird oft begleitend in Lehrveranstaltungen zum SCM eingesetzt. Auch das *Automobile Supply Chain Management Game* des Dublin Institute of Technology kann zur Vermittlung von Wechselwirkungen bei der Gestaltung von Produktionsnetzen durch Studierende genutzt werden [Tob-11]. REMI-OMOSOWON ET AL. (2016) beschreiben die Implementierung einer gamifizierten simulationsbasierten Lernumgebung zur Ausbildung von Lagerarbeitern [Rem-16]. Bevor Lagerarbeiter schwere und sperrige Packstücke kommissionieren, können sie mögliche Anordnungen vorab auf einem Tablet spielerisch simulieren. DELAGO ET AL. (2016) berichten über ein von ihnen entwickeltes Werkzeug, das auf einem Spezialsimulator für Lean Management basiert. Lernende sollen Lean-Prinzipien verstehen [Del-16].

In allen benannten Beispielen der *SIM in EDU*-Perspektive sollen Lernende durch Experimentieren in einer vorbereiteten Simulationsumgebung Zusammenhänge eines vordefinierten Anwendungsbereichs anhand eines konkreten Szenarios verstehen. Die Simulation ist ausschließlich als Hilfsmittel zu betrachten, um Zusammenhänge in einem Anwendungsbereich verständlich aufzuzeigen. In keinem der Ansätze steht die Simulationstechnik im Sinne eines Lernziels im Vordergrund. Lernende bedienen ein GUI, Simulationsinstrument oder einen Spezialsimulator, welche lediglich die im Hinblick auf das Lernziel spezifischen Funktionsumfänge umfassen. Beispielsweise lassen sich zumeist die Werte ausgewählter Attribute durch einfach gestaltete, anwendungsfreundliche GUIs intuitiv manipulieren. Es sind lediglich vordefinierte Parametervariationen möglich.

Zusammenfassend lässt sich feststellen, dass die Werkzeuge, die im Kontext der *SIM in EDU* zum Einsatz kommen, durch Reduktion des bereitgestellten Funktionsumfangs stark vereinfacht und mit Blick auf ein spezifisches Teilgebiet eines Anwendungsbereichs eingeschränkt sind.

Vor diesem Hintergrund sind die vorgestellten Ansätze kaum dazu geeignet, Studierende in ereignisdiskrete Simulation einzuarbeiten. Der Fokus dieser Arbeit liegt auf der *EDU in SIM* mit Simulationsinstrumenten der Ebene drei nach Abbildung 2.3 auf Seite 21. Im Folgenden werden ausschließlich Ansätze, die der Sichtweise *EDU in SIM* zuzuordnen sind, betrachtet. Ansätze, die einen Beitrag zur Einarbeitung in ereignisdiskrete Simulationsinstrumente zur Durchführung von Simulationsstudien versprechen, sind aufgeführt. Die strukturierte Bewertung der im Folgenden angeführten Ansätze ist im Kapitel 3.2 vorgenommen.

Serious Games: Als interessante Ansätze im Kontext von Materialflusssystemen lassen sich die Serious Games *PlantVille* von *Siemens*, *Production line* von *Positech Games* und *Learn2Work* von *Korion* benennen.

Bei *PlantVille* steht nach Angaben des Magazins *COMPUTERWOCHE* (2011) die Steigerung von Produktivität, Effizienz und Nachhaltigkeit einer digital simulierten Fabrik im Vordergrund [Com-11]. Interessensgruppen wie Studierende oder Kunden erhalten einen Einblick in den Alltag eines sogenannten Anlagenmanagers. In einer sehr aufwändig gestalteten dreidimensionalen Web-Oberfläche stehen dem Nutzer vorbereitete Szenarien zur Wahl. Diese umfassen eine Flaschenabfüllanlage, eine Vitaminfabrik und eine Eisenbahnfertigung. Eine virtuelle Person namens *Pete* gibt dem Spieler Tipps zur Anlagenplanung. Darüber hinaus ist ein Chat mit realen Experten möglich. KPIs zur Bewertung von Sicherheit, Lieferzeit, Qualität, das Energiemanagement und die Mitarbeiterzufriedenheit dienen dem Vergleich von Spielern [Ada-11]. Der animierte Charakter *Pete* leitet den Spieler im Sinne einer Geschichte an. Je nach gewähltem Szenario löst der Spieler unterschiedliche Aufgaben und verbessert Level für Level die KPIs. Das *PlantVille Café* ist als virtueller sozialer Treffpunkt gestaltet, in dem Fachthemen besprochen werden. Inzwischen ist *PlantVille* nicht mehr über die von Siemens angegebene Internetadresse erreichbar. Hinweise, warum die Software nicht mehr öffentlich zugänglich ist, gibt der Hersteller nicht.

Der Hersteller *Positech games* bietet ein dreidimensional aufwändig gestaltetes Serious Game mit dem Namen *Production line* zum Verkauf an [Pos-19]. Um im Serious Game erfolgreich zu sein, muss der Nutzer die Produktionskosten einer digital simulierten Autofabrik so weit wie möglich verringern. Hierzu kann der Spieler das Layout einer Automobilproduktion, z.B. Produktionsbereiche, Ressourcen, Wege und Energieversorgung, gestalten.

Bei *Learn2Work* handelt es sich um eine dreidimensionale sogenannte Unternehmenssimulation [Kor-11] [lea-15]. Die Entwickler zielen auf einen Einsatz in der betrieblichen Weiterbildung ab. *Learn2Work* umfasst die Arbeitsbereiche Produktion, Personalwesen, Logistik, Marketing und Vertrieb sowie Controlling. Unter anderem können Lernende in die Produktionsplanung, Verbesserung des Layouts, Investitionsentscheidungen und ergonomische Arbeitsplatzgestaltung eingearbeitet werden. Die Idee für *Learn2Work* wurde 2004 am Fraunhofer Institut für Arbeitswirtschaft und Organisation entwickelt. Das Serious Game wurde von der Firma *Korion* realisiert. Der Spieler übernimmt mit weiteren Spielern als Manager die Verantwortung für einen virtuellen Produktionsbetrieb. Das Ziel besteht darin, dass Spieler die Wirkzusammenhänge in der Fabrik systematisch erfassen. Um in *Learn2Work* erfolgreich zu sein, gilt es, richtige Entscheidungen bspw. hinsichtlich der Priorisierung profitabler Aufträge, des Personaleinsatzes und der -qualifikation sowie Marketingkampagnen zu treffen [Goe-11]. Das Serious Game wird im Sinne eines blended learnings durch Fachseminare begleitet.

Projekt eQual 2.0: Das Projekt *eQual 2.0 e-Qualifizierung für effiziente Logistikprozesse* hatte zum Ziel, „jedem Angestellten eines Unternehmens die Durchführung von Simulationen und damit die Evaluierung neuer Ideen zur Umgestaltung des jeweiligen Arbeitsplatzes zu ermöglichen“ [Mos-13]. MOSBLECH ET AL. (2013) entwickelten vor dem Hintergrund der kostenintensiven Schulung

„aller“ Mitarbeiter eines Unternehmens eine Benutzungsschnittstelle, die einen interaktiven Umgang mit eQual 2.0 ermöglicht. Bei eQual 2.0 handelt es sich um eine IT-Architektur, die maßgeblich aus einer Benutzungsschnittstelle, einem Webserver und einem Lernmanagement System besteht. Diese drei Komponenten wirken als Serviceorientierte Architektur. Die Autoren verstehen unter der Serviceorientierten Architektur das hierarchische Zusammenspiel der benannten Komponenten, wobei der Benutzer durch die Benutzungsschnittstelle nach der Dateneingabe eine Service-Anforderung (=Auftrag) an ein Simulationsinstrument sendet. Das Simulationsinstrument arbeitet Aufträge gemäß dem FIFO-Prinzip ab. Die Ergebnisse werden daraufhin an die Benutzungsschnittstelle zur Ansicht für den Nutzer übertragen. Die Benutzungsschnittstelle ist als Website umgesetzt. Voraussetzung zur Nutzung durch unerfahrene Simulationsanwender ist, dass ein Experte bereits ein Simulationsmodell erstellt hat. Der Mitarbeiter kann einzelne Werte von Attributen des Experten-Modells ändern und somit die Auswirkungen dieser anhand der resultierenden Ergebnisse abschätzen.

Webbasierte Supply Chain Simulation: Aufgrund des hohen Aufwands zur Durchführung von Simulationsstudien und der dafür erforderlichen Qualifikation erarbeitete SCHMUCK (2014) in seiner Dissertation ein Referenzmodell zur Modellierung von Supply Chain Netzwerken sowie eine webbasierte Simulation von Produktions- und Logistiksystemen [Sch-14b]. Er möchte Unternehmen dazu befähigen, die Simulation stärker als bisher in den betrieblichen Alltag zu integrieren. Durch das Referenzmodell sollen Anwender schnell komplexe Supply Chain Netzwerke abbilden, auch wenn diese keine Simulationsexperten sind. Das Referenzmodell ist an dem SCOR (=Supply Chain Operations Reference)-Modell orientiert [Kuh-08, S.227f.]. Das Referenzmodell umfasst die Bausteine *demand, plan, source, store, make, deliver* und *transport* [Sch-14b]. Jeder der vom Anwender eingesetzten Bausteine kann individuell parametrisiert werden. Die Bausteine werden durch Eingangs- und Ausgangsbeziehungen miteinander verbunden, um so den Materialfluss sowie den entgegengesetzten Informationsfluss abzubilden. Unter einer webbasierten Simulation versteht der Autor eine Client-Server-Architektur, bei der „eine Datenbank das zentrale Element zwischen Modellierung und Simulation ist“ [Sch-14b]. Der Ansatz lässt sich der *Fern-Simulation und Visualisierung / Animation* zuordnen. Diese Architektur begünstigt gemäß den Ausführungen des Autors u.a. die schnelle und kostengünstige Verteilung der Simulationsanwendung, Simulatorunabhängigkeit und Erstellung neuer Geschäftsmodelle.

Einen ähnlichen Ansatz verfolgt auch FECHTELER (2019) im Rahmen des EU-geförderten Projekts *e-save* mit dem Werkzeug *SimChain* [Fec-19]. Es handelt sich um eine Weiterentwicklung von *ICON-SimChain* [Gut-04]. Bausteine wie *Customer, Sites, Hubs, Plain Suppliers* sowie *Transport Relations* sind für den Benutzer vorbereitet. Dieser kann die Bausteine in einem Web-GUI instanziierten und parametrisieren.

Simulationsbasierte Wertstromanalyse: Im Kontext der Fabrikplanung kann das Wertstromdesign als Methode zur Erfassung und übersichtlichen Darstellung von Wertströmen in einer Fabrik genutzt werden [Tsc-16]. Unter einer Wertstromanalyse ist ein Hilfsmittel zur Abbildung der gesamten Prozesskette einer Produktfamilie mit den dazugehörigen Informations- und Materialflüssen zu verstehen [Hei-08, S.312]. Bei der Wertstrommethode wird der Wertstrom durch Symbole

für Produktions- und Geschäftsprozesse, den Material- und Informationsfluss, Kunden und Lieferanten dargestellt [Erl-10, S.32f.]. Das Ziel besteht darin, Verschwendungen im Wertstrom zu identifizieren [Hei-08, S.312]. Die Durchführung einer Wertstromanalyse wird an vielen technisch und wirtschaftlich orientierten Hochschulen im Kontext des Lean Managements gelehrt.

Die simulationsbasierte Wertstromanalyse stellt einen Ansatz für den vereinfachten Aufbau von Simulationsmodellen dar. LIAN UND VAN LANDEGHEM (2002) entwickelten ein erstes Vorgehen, das die Wertstrommethode und ereignisdiskrete Simulation miteinander in Verbindung bringt [Lia-02]. Zuerst wird eine Wertstromanalyse traditionell mit „Bleistift und Papier“ durchgeführt. Diese wird im nächsten Schritt in einem Simulationsmodell formalisiert. Die statisch ermittelten Ergebnisse werden zur Validierung des Simulationsmodells genutzt. Schließlich sind vor dem Hintergrund des Einflusses weiterer Produkte auf das Wertstromdesign Parameterstudien durchzuführen. Abschließend ist das Simulationsmodell zur Weiterbildung von Mitarbeitern aufzubereiten und zu nutzen. Im Jahr 2007 berichteten die Autoren in einem weiteren Aufsatz über die Entwicklung von Bausteinen/Objekten für das ereignisdiskrete und objektorientierte Simulationsinstrument Enterprise Dynamics [Lia-07].

Mit Blick auf das grafische Erscheinungsbild besteht zwischen der traditionellen Wertstromanalyse und dem simulationsbasierten Ansatz eine hohe Ähnlichkeit, was es geübten Anwendern der Wertstromanalyse erleichtert, dynamische Simulationsstudien durchzuführen. Darüber hinaus sind Lean-geschulte Mitarbeiter in der Lage, die Simulationsergebnisse in der ihnen bekannten grafischen Form zu „lesen“ [Sol-09].

Die simulationsbasierte Wertstromanalyse ist in der Zwischenzeit weit verbreitet. Beispielsweise bietet Siemens standardmäßig mit dem Simulationsinstrument *Tecnomatix Plant Simulation* eine solche Klassenbibliothek zur Modellierung und Simulation an [Sie-19]. Der *Process Simulator* von ProModel ist ein Add-In der Software Microsoft Visio und ermöglicht dem Anwender durch Instanziierung vordefinierter sogenannter Shapes den Aufbau von Simulationsmodellen [Pro-19]. Die auf Simulation spezialisierte Unternehmensberatung SIMPLAN AG (2019) berichtet auf ihrer Homepage über das aktuelle Forschungsprojekt *MobiSim* [Sim-19a]. In diesem wird ein sogenannter Wertstromsimulator entwickelt, der zur Nutzung auf mobilen Endgeräten zur Verfügung stehen soll. Als potenzielle Anwender sind produktionsnahe Mitarbeiter benannt. Nach Angaben der Unternehmensberatung verspricht der mobile Ansatz Transparenz und Akzeptanz von Veränderungen bei Mitarbeitern, „sorgt für eine Einbeziehung von Mitarbeitern und erlaubt eine Vorab-Quantifizierung von Veränderungen ohne Medienbruch direkt am Ort des Geschehens“ [Sim-19a]. Des Weiteren benennt die SIMPLAN AG (2019) auf deren Unternehmenswebsite das Produkt *SimVSM* [Sim-19b]. Auf einem mobilen Endgerät lassen sich damit Wertströme modellieren. Per sogenanntem Webservice werden die Modelle an einen Simulationsserver übertragen. Innerhalb kurzer Zeit werden dann die Simulationsergebnisdaten an das mobile Gerät zurück übertragen. Zur Nutzung von *SimVSM* sind nach Herstellerangaben Kenntnisse in der Modellierung von Wertströmen erforderlich.

Manufacturing in Python: OLAITANA ET AL. (2014) stellen fest, dass insbesondere KMU, die in kommerzielle Simulationsinstrumente investiert haben, eher zurückhaltend bei weiteren Investitionen sind. Dies ist vor allem mit Blick auf die Ausbildung von Mitarbeiter/innen für die Modellierung und Simulation der Fall [Ola-14]. Sie erklären, dass die Qualifikation von Simulationsanwendern für den Projekterfolg maßgeblich ist. Die Autoren beschreiben in ihrem Aufsatz die Entwicklung des ereignisdiskreten Simulationsinstruments *Manufacturing in Python*. Es steht open-source zur Verfügung. Python ist eine objektorientierte Programmiersprache. Das Simulationsinstrument basiert auf dem open-source ereignisdiskreten Simulationswerkzeug *Simulation in Python* [Sch-14a]. Simulation in Python beinhaltet einen Zeitfortschrittsmechanismus sowie Funktionalitäten zur Ereignisverwaltung. Gemäß den Ausführungen von OLAITANA ET AL. (2014) ist Manufacturing in Python so gestaltet, dass Nutzer mit unterschiedlichem Vorwissen das Simulationsinstrument zur Entscheidungsunterstützung nutzen können. Anwender mit geringen Vorkenntnissen können auf Bausteine zurückgreifen, die auf einer open-source Plattform durch sogenannte hochqualifizierte Simulationsanwender, Statistiker und Akademiker bereitgestellt werden. Darüber hinaus sollen KMU bei der Nutzung des Simulationsinstruments davon profitieren, dass Anbindungen an Enterprise-Resource-Planning-Werkzeuge verfügbar sind, die einen schnellen Modellaufbau zulassen. Die Bausteine der Software Manufacturing in Python sind als produktionsspezifische Klassenobjekte und -bausteine zu verstehen, die jeweils vordefinierte Funktionalitäten abbilden [Ola-14]. Diese Bausteine lassen sich in sogenannte *Basis-Objekte* wie Arbeitsstationen oder Puffer, *Unterbrecher* wie Fehler oder Schichtmodelle, *nicht-stationäre Objekte* wie Produkte oder der Kunde an der Verkaufsstelle, *Ressourcen-Objekte* wie Werker oder Elektrizität und *unterstützende Objekte* wie Programme zur Abbildung individueller Funktionalitäten unterteilen [Ola-14].

Projekt simKMU: Das zwischen 2008 und 2011 geförderte Projekt „simKMU - Entwicklung unternehmensübergreifender, prozessintegrierter und internetbasierter Simulationsdienstleistungen für KMU“ hatte zum Ziel, „die Wettbewerbsfähigkeit von KMU zu stärken, indem über eine webbasierte Infrastruktur neue, internetbasierte Simulationsbausteine für unterschiedliche Anwendungen und Geschäftsmodelle in den Bereichen Logistik, Produktionsplanung und Geschäftsprozesse angeboten werden“ [Wim-11, S.2ff.]. Konkret wurde eine IT-Architektur erarbeitet, die das Erstellen von Simulationsmodellen über einen Web-Browser zulässt. Die Benutzungsoberfläche, im Folgenden als *SimWebApp* bezeichnet, ist dabei auf die Anforderungen von KMU zugeschnitten. Zu diesen Anforderungen zählen, dass Anwender Simulationsstudien „ohne Vorkenntnisse“ durchführen können sowie eine benutzerfreundliche Bedienung und ein einfaches Datenmanagement [Wim-11, S.186ff.]. Die vom Benutzer eingegebenen Daten werden in eine Datenbank übertragen, die wiederum einen Datenaustausch mit einem Simulationsinstrument ermöglicht. Das Simulationsmodell generiert sich automatisch basierend auf den vom Benutzer eingegebenen Daten. Die Autoren beschreiben, dass das Simulationsinstrument ohne größere Änderungen an SimWebApp austauschbar sei. Sie schränken dies allerdings auf eine sogenannte ähnliche Datenbasis, bestehend aus Ressourcen, Produkten und Prozessen, ein [Wim-11, S.263ff.].

Die Autoren argumentieren, dass die grafische Animation nur für repräsentative Zwecke sinnvoll sei und bieten aus diesem Grund dem Benutzer die Auswertungen von Simulationsläufen mithilfe von Verlaufsdiagrammen und logistischen Kennzahlen an [Wim-11, S.283ff.]. An die-

ser Stelle sei betont, dass die Animation sich dazu nutzen lässt, „Funktionalitäten und Abläufe zu verstehen oder spezifische Modellsituationen zu erklären“ [Wen-08c, S.148]. Zwar beschreiben sie im Projektabschlussbericht die Integration einer 3D Umgebung zur Animation, jedoch scheint dies nur als ein Use-Case implementiert zu sein [Wim-11, S.398].

Der im Projekt simKMU erarbeitete Vorschlag hinsichtlich des E-Learnings sei im Folgenden kurz ausgeführt. Die Autoren konzipierten einen Leitfaden. Dieser richtet sich sowohl an Anfänger als auch an erfahrene Mitarbeiter aus den Themenbereichen Logistik und der sogenannten Simulationsmethoden. Sie benennen relevante Themenfelder aus dem Bereich der Modellierung und Simulation, wie die Beurteilung der Simulationswürdigkeit oder Methoden der Verifikation und Modellierung und ordnen diesen ihrer Auffassung nach passende Vermittlungsmethoden zu. E-Learning-Inhalte wie Wikis, das Aufzeigen von sogenannten Best Practices anhand konkreter Beispiele, textuell beschriebene Aufgaben, eine Sammlung häufig gestellter Fragen inklusive Antworten, Video-Seminare und ein Glossar sind beispielhaft benannt. Gemäß den Schilderungen von WIMPF ET AL. (2011) erzeugten sie umfangreiches Lehrmaterial zu den Themenfeldern Grundlagen der Simulationstechnik, Grundlagen von Simulationsinstrumenten, praktische Grundlagen der Logistiksimulation, Grundlagen der Modellierung, Grundlagen der Datenerhebung und -strukturierung, Grundlagen zur Durchführung einer Simulation, Grundlagen der Produktionslogistik und Grundlagen der Statistik und Wahrscheinlichkeitstheorie. SimWebApp beinhaltet kurze Hilfetexte und textuelle Schritt für Schritt Anleitungen, um Benutzer hinsichtlich der Bedienung der Benutzeroberfläche zu unterstützen [Wim-11, S.301ff.].

Simulationsassistenzsysteme: Simulationsassistenzsysteme unterstützen die Durchführung einer Simulationsstudie von der anfänglichen Datensammlung und Bewertung über das Experiment-Design bis hin zur Auswertung der Ergebnisse [May-12] [May-15a]. Assistenzsysteme unterstützen den Modellierer durch (teil-)automatische Modellgenerierung und durch Bereitstellung von im Hinblick auf das Anwendungsgebiet relevanten Bausteinkästen. Konkret beschreiben die Autoren die Software *SimAssist*. Diese besteht aus einer Benutzeroberfläche, mehreren Datenbanken sowie einer Client-Server-Anwendung. Simulationsstudien können definiert und Experimente durchgeführt werden. Das Werkzeug richtet sich an Experten im Umgang mit ereignisdiskreter Simulation.

Textuelle Aufgaben: Derzeit basiert die Ausbildung im Kontext der ereignisdiskreten Simulation oft auf textuellen Aufgaben. Im Buchhandel erhältliche Sammelwerke sind beispielsweise die Folgenden: [Ban-15], [Ban-11], [Ban-10a], [Ban-08], [Ele-12], [Kel-15] oder [Kel-10]. Hypothetische Fälle und Probleme sind textuell beschrieben, die dem Lernenden als Grundlage für die Durchführung von Simulationsstudien dienen. Zumeist ist in den Publikationen Schritt für Schritt im Sinne eines Tutorials eine exemplarische Lösung der Aufgabe in einem spezifischen Simulationsinstrument abgedruckt und diskutiert. Dies soll es Lernenden ermöglichen, im Eigenstudium die Durchführung von Simulationsstudien inklusive der Anwendung eines exemplarischen Simulationsinstruments zu erlernen. Zumeist werden Lernende in den Büchern dazu angeleitet, Aufgaben „linear“ durchzuarbeiten. Die Bücher beziehen sich in der Regel auf ein spezifisches Simulationsinstrument.

Projekt ClouDES: PADILLA ET AL. (2015 UND 2016) beschreiben die Nutzung von Simulationsspielen zum Lehren und Lernen von ereignisdiskreter Simulation [Pad-15] [Pad-16]. Die Autoren schlagen vor, zunächst Simulationsspiele als unterhaltsame Lernaktivität, einzusetzen. Bei dieser sollen Schüler kaum näher spezifizierte ereignisdiskrete Simulationskonzepte durch reduzierte Fachsprache in Form von Bausteinen erlernen, um darauffolgend in einen sogenannten noch ernstere Serious-Game-Modus zu wechseln. Konkret sind von den Autoren die zwei Spiele *Medieval Wars* und *Dystopian City* benannt, die in dem Simulationsinstrument *ClouDES* implementiert sind [Pad-14]. Das Kunstwort ClouDES setzt sich aus dem cloudbasierten Ansatz und der in der US-amerikanischen Literatur weitverbreiteten Abkürzung Discrete-Event Simulation (DES) (=ereignisdiskrete Simulation) zusammen. Bei ClouDES handelt es sich um einen Ansatz der Simulation in der Cloud. Vom Benutzer eingegebene Daten werden an einen Server inklusive Simulationsinstrument übermittelt, der die Ergebnisse eines Simulationslaufs wiederum zum Benutzer zurückschickt [San-16]. Das GUI ist plattformunabhängig gestaltet und lässt sich im Web-Browser sowie als App darstellen. Die Autoren beschreiben, dass das GUI vom Benutzer eingegebene Daten an ein Simulationsinstrument über das Web sendet. Nach erfolgten Simulationsläufen werden wiederum die Ergebnisse an das GUI übertragen und dem Benutzer angezeigt. ClouDES zielt darauf ab, Schüler für die Fächer Mathematik, Informatik, Naturwissenschaft und Technik zu begeistern. Das Ziel beider Spielansätze besteht darin, Simulationsmodelle durch Instanziierung vorbereiteter Bausteine und Objekte gemäß einer Spielbeschreibung aufzubauen. Mit Blick auf das Ziel des jeweiligen Spiels müssen Nutzer die korrekten Bausteine und Objekte wählen und parametrisieren. Der Ablauf in ereignisdiskreten Simulationsinstrumenten soll dem Spieler durch Warteschlangen aufgezeigt werden. Diese ergeben sich im Simulationsmodell. Das Modell wiederum wird von Lernenden gemäß einer ihnen erzählten Geschichte aufgebaut. Die Idee für *Medieval Wars* basiert auf einem Schieß- und Strategiespiel, in welchem die Spieler eine simulierte Menschengruppe aus einer gefährlichen Stadt leiten sollen. Ziel ist, möglichst viele Menschen lebendig aus der Stadt zu leiten. Dazu müssen Lernende ein partiell verfügbares digitales Modell gemäß einer Beschreibung um sogenannte Entscheidungsblöcke ergänzen. Danach ist eine Parametervariation durch den Spieler im Sinne einer Sensitivitätsanalyse erforderlich. *Medieval Wars* ist ähnlich aufgebaut, jedoch beschreiben die Autoren ein militärisches Szenario, das im Mittelalter spielt. Das Lernziel besteht in der Anwendung und Parametrisierung weiterer Objekte und Bausteine.

Im folgenden Kapitel 3 sind die neun identifizierten relevanten Vorarbeiten zur Einarbeitung in ereignisdiskrete Simulation anhand von 17 Kriterien charakterisiert. Im Kapitel 3.1 sind zunächst die Kriterien zur Abgrenzung benannt. Die Abgrenzung ist im Kapitel 3.2 vorgenommen.

3 Handlungsbedarf

Die vorliegende Arbeit soll einen Beitrag dazu leisten, ereignisdiskrete Simulation zur Entscheidungsunterstützung bei der Neu- und Umplanung von Materialflusssystemen zu verbreiten. Eine mobile und spielerische Lernumgebung soll konzipiert und prototypisch umgesetzt werden. Sie soll es Anfängern ermöglichen, sich in ein Simulationsinstrument zur Durchführung von Simulationsstudien einzuarbeiten. Zunächst sind im Kapitel 3.1 Anforderungen an das Konzept beschrieben, die sich aus den Erkenntnissen der Kapitel Planung von Materialflusssystemen, Simulation von Materialflusssystemen, Qualifizierung von Simulations-Anwendern und Erhöhung der Lehr- und Lernproduktivität ableiten. Im darauffolgenden Kapitel 3.2 sind die zuvor benannten Ansätze aus Kapitel 2.5 von den Anforderungen abgegrenzt, woraus sich der Handlungsbedarf ergibt.

3.1 Anforderungen

Die zielgerichtete Entwicklung einer Lernumgebung für ereignisdiskrete Simulation erfordert klar definierte *Lernziele* im Kontext eines abgegrenzten *Anwendungsbereichs*. Etablierte Vorgehensweisen zum *Instruktionsdesign* sowie zur *Spielentstehung* sind dabei zu beachten. Eine skalierbare *IT-Architektur* ermöglicht eine orts- und zeitunabhängige Nutzung der Lernumgebung. In der Tabelle 3.1 auf Seite 53 sind zusammenfassend die Beziehungen zwischen den Anforderungen und den Erkenntnissen aus den vorangegangenen Kapiteln dargestellt. Die Tabelle ermöglicht das schnelle Nachschlagen. Im Folgenden sind die soeben kurz beschriebenen Felder ausgeführt.

- ① - **Lernziel:** Eine Lernumgebung sollte hinsichtlich des Lernziels so gestaltet sein, dass Lernende in die Bedienung eines Simulationsinstruments eingearbeitet werden (Anforderung 1). Die Einarbeitung kann zunächst in einer dezidierten Benutzungsoberfläche stattfinden. Bei der Entwicklung der Oberfläche gilt es zu beachten, dass diese hinsichtlich des grafisch-interaktiven Modellaufbaus und des Erscheinungsbilds dem jeweiligen Simulationsinstrument nachempfunden ist. Lernende sollen in relevante vordefinierte physische und logische Bausteine sowie in Objekte des Simulationsinstruments eingearbeitet werden (Anforderung 2). Logische Bausteine umfassen auch Sprachkonzepte wie Programmiersprachen. Diese Sprachen ermöglichen es, individuelle Simulationsmodelle aufzubauen. Simulationsinstrumente für spezifische Anwendungsbereiche, Ebene drei, vgl. Abbildung 2.3, bieten dem Benutzer in der Regel eine entsprechende Schnittstelle zur Programmierung an. Die Nutzung von Programmiersprachen zur Implementierung von Simulationsmodellen setzt voraus, dass Anwender in die Syntax und Semantik eingearbeitet wurden

(Anforderung 3). Die Beherrschung von Syntax und Semantik ist Voraussetzung für den Aufbau verifizierter Simulationsmodelle unter Verwendung von Programmiersprache zur Implementierung von Simulationsmodellen [VDI-3633b, S.21]. Schließlich sollen die Lernenden Erfahrung in der Durchführung von Simulationsstudien sammeln (Anforderung 4). Dies ist erforderlich, da die Abstraktion sowie die Ergebnisanalyse und -interpretation Erfahrung erfordern [Ban-10b, S.23f.] [Sha-98].

- ② - **Anwendungsbereich:** Hinsichtlich des Anwendungsbereichs der Materialflusssysteme soll ein Schwerpunkt auf die Einarbeitung in Simulationsinstrumente gelegt werden, die die flexible Abbildung der Gestaltungsprinzipien Fließprinzip und Pull-Prinzip unterstützen (Anforderung 5). Die Gestaltungsprinzipien umfassen bewährte Methoden zur Verbesserung von Materialflusssystemen, die häufig in der Praxis Anwendung finden und meist in relevanten Lehrveranstaltungen für die beschriebene Zielgruppe behandelt werden. Mit Anforderung 6 ist die Erweiterbarkeit der Modellwelt um anwenderdefinierte Bausteine sicherzustellen. Diese Flexibilität soll vor dem Hintergrund jüngster Entwicklungen um CPPS am Beispiel der Multiagentensimulation ermöglicht werden.
- ③ - **Motivation und Instruktionsdesign:** Es gilt mit Blick auf die Motivation und das Instruktionsdesign, eine zeitgemäße Lernumgebung zu konzipieren, die die jungen Lernenden zur Nutzung motiviert. Der Zusammenhang zwischen Lernerfolg und Motivation durch Gamification ist in mehreren Erfahrungsberichten in der wissenschaftlichen Literatur dokumentiert [Bar-13a] [Bar-13b] [Col-09] [Mül-15] [Rhe-04]. Vor diesem Hintergrund sollen Spielelemente in der Lernumgebung angewendet werden (Anforderung 7), die die Zusammenarbeit und Konkurrenz von Studierenden in der Gruppe fördern. In der Fachliteratur ist zwischen unterschiedlichen Spieler-Typen unterschieden. Für jeden Spieler-Typ müssen in der Lernumgebung eine Auswahl passender Spielelemente gestaltet sein (Anforderung 8). Die Beachtung der Flow-Theorie sei als Anforderung 9 benannt [Csi-14a, S.247ff.]. Hierdurch soll der Einklang von Herausforderungen in Form von zu lösenden Aufgaben mit den individuellen Fähigkeiten des Spielers sichergestellt sein und der Lernerfolg erhöht werden [Met-11, S.65]. Eine Lernumgebung ist gemäß einer etablierten Vorgehensweise wie dem Instruktionsdesign zu entwickeln (Anforderung 10). Dies stellt sicher, dass die Lernumgebung strukturiert und zielorientiert entsteht.
- ④ - **Umsetzung und Betrieb der Lernumgebung:** Mit Blick auf die Umsetzung und den Betrieb der Lernumgebung ist eine Cloud-IT-Architektur vorteilhaft (Anforderung 11). Diese erlaubt eine räumlich verteilte und skalierbare Anordnung von Simulationsinstrumenten, Datenbank sowie von Ein- und Ausgabegeräten. Eventuell vorhandene Lizenzen für Simulationsinstrumente können effektiv genutzt werden. Eine solche Architektur ist Voraussetzung für eine orts- und zeitunabhängige Interaktion mit einem mobilen Endgerät (Anforderung 12). Die Modellierung im Sinne einer Dateneingabe und die Ergebnisanzeige sind als Basisfunktionalitäten eines solchen GUI zu benennen. Weiterhin ist ein mobiler Ansatz zur Erhöhung der Lehr- und Lernproduktivität geeignet. Eine mehrmalige kurzzeitige Nutzung „zwischendurch“ kann dadurch ermöglicht werden. Durch die Interaktion mit einem ähnlich dem Simulationsinstrument gestalteten GUI sollen Lernende schließlich in ein Simulationsinstrument eingearbeitet werden. Das Simulationsinstrument soll in den Lernprozess mit einbezogen werden (Anforderung 13). Online-Funktionalitäten sind als Anforderung 14 zu benennen. Hierunter sind Mechanismen zu verstehen, die die Gruppenarbeit stärken. Beispi-

le sind Schnittstellen zur Interaktion mit anderen Nutzern. Die Umsetzung einer ansprechenden Animation ist als Anforderung 15 zu benennen. Diese ermöglicht es dem Lernenden, Funktionalitäten und Abläufe im Simulationsmodell zu verstehen oder spezifische Modellsituationen zu erklären [Wen-08a, S.148]. Die mobile Lernumgebung zur Einarbeitung in das Simulationsinstrument und insbesondere der grafisch-interaktive Modellaufbau sollen so gestaltet sein, dass diese Ähnlichkeiten zum Simulationsinstrument aufweisen, in das Lernende eingearbeitet werden sollen (Anforderung 16). So ist sichergestellt, dass Lernende sich bei einem Wechsel von der dezidierten Lernumgebung hin zum Simulationsinstrument schnell zurechtfinden und z.B. die Symbole wiedererkennen. Abschließend gilt es festzuhalten, dass es attraktiv ist, physische Produktionsumgebungen wie Lernfabriken in den Lernprozess einzubeziehen (Anforderung 17), da sich diese insbesondere zum Aufbau von Erfahrung bei der in der Arbeit benannten Zielgruppe eignen.

Tabelle 3.1: Kapitelverweise für Anforderungen

Feld	Anforderung	Kapitel
①	1 - Einarbeitung in Simulationsinstrument	1; 2.3.1; 2.3.2
	2 - Einarbeitung in Bausteine/Objekte, Ebene 3	2.3.2.1
	3 - Einarbeitung in Programmiersp. zur Implement. v. Sim.-modellen	2.3.2.2
	4 - Durchführung von Simulationsstudien	1; 2.2.4
②	5 - Materialflusssystem/Fließprinzip und Pull-Prinzip	1; 2.1.1
	6 - Multiagentensystem	2.1.2; 2.2.3
③	7 - Nutzung von Spielelementen	2.4.2
	8 - Beachtung unterschiedlicher Spieler-Typen	2.4.2.1
	9 - Beachtung des „Flows“	2.4.2.2
	10 - Beachtung des Instruktionsdesigns	2.4.1; 2.4.2.3
④	11 - Cloud-IT-Architektur	2.2.2; 2.3.2.3; 2.3.2.4
	12 - Interaktion mit GUI auf mobilem Gerät	1; 2.3.2.4
	13 - Interaktion mit Simulationsinstrument	1; 2.3.1
	14 - Online-Funktionalitäten	1; 2.4.2.3
	15 - Animation	2.3.2.4
	16 - Gestaltung/Interaktion ähnlich zu Simulationsinstrument	2.3.2; 2.4.4
	17 - Einbeziehung von Lernfabrik	2.4.3

Legende:

- ① – Lernziel
- ② – Anwendungsbereich
- ③ – Motivation und Instruktionsdesign
- ④ – Umsetzung und Betrieb der Lernumgebung

3.2 Bewertung und Zwischenfazit

Zur Ermittlung des Handlungsbedarfs sind in der Tabelle 3.2 auf Seite 55 die im Kapitel 2.5 dokumentierten Vorarbeiten aufgegriffen und hinsichtlich der im Kapitel 3.1 benannten Anforderungen charakterisiert. Im unteren Teil der Tabelle 3.2 sind die Vorarbeiten aus dem Kapitel 2.5 darüber hinaus angesichts de-

ren Eignung für Lernende mit unterschiedlichem *Vorwissen* unterschieden. Auch findet sich im unteren Teil der Tabelle eine vergleichende Einordnung des jeweils fokussierten *Simulationsinstruments* wieder.

Das *Vorwissen* ist wie folgt unterschieden:

- Anfänger: Lernender, der bislang keine Simulationsinstrumente nutzte.
- Fortgeschrittener Anfänger: Der Lernende hatte bereits ersten Kontakt mit ereignisdiskreten Simulationsinstrumenten. Er baute bereits erste kleine Modelle auf.
- Fortgeschrittener: Der Lernende führte bereits eine Simulationsstudie auf Basis einer textuellen Aufgabe durch.
- Erfahrener Fortgeschrittener: Der Lernende hat bereits eine Simulationsstudie im Sinne eines Industrieprojekts abgeschlossen.
- Experte: Der Lernende führte bereits mehrere Simulationsstudien durch.

Die Einordnung bzgl. des *Simulationsinstruments* richtet sich nach der in Abbildung 2.3 auf Seite 21 gezeigten Taxonomie. Im Folgenden sind die relevanten Vorarbeiten aus Kapitel 2.5 kurz erneut aufgegriffen und mit Blick auf das Ziel der Arbeit diskutiert.

Serious Games: Die drei Serious Games *PlantVille*, *Production line* und *Learn2Work* fokussieren meist vorgegebene sehr spezifische Vorgänge wie das Abfüllen von Flaschen. Oft können nur einzelne Attribute durch den Spieler verändert werden. Die Modellwelt ist kaum erweiterbar. Sie eignen sich kaum zur Lösung realer Probleme. Eine Zuordnung von Serious Games zu der Klasse der Simulationsinstrumente für einen Teilbereich von Materialflusssystemen scheint zulässig. Sehr spezifische Bausteine, die sich kaum parametrisieren lassen und fehlende Schnittstellen sind die maßgeblichen Schwachpunkte für einen Einsatz im Sinne industrieller Fragestellungen. Sehr wohl vorstellbar ist, dass Serious Games sich dazu eignen, Spielern die grundlegenden Zusammenhänge in Materialflusssystemen zu vermitteln.

Projekt eQual 2.0: Konkrete Lernziele für eQual 2.0 sind im Aufsatz von MOSBLECH ET AL. (2013) nicht benannt [Mos-13]. Der beschriebene Ansatz scheint eher darauf abzielen, Nutzern anschaulich produktionslogistische Zusammenhänge aufzuzeigen, als Lerninhalte zum Modellaufbau und zur Simulation zu vermitteln. Eine Besonderheit des Ansatzes ist, dass eQual 2.0 in das Lernmanagement System Moodle (=Modular Object-Oriented Dynamic Learning Environment) eingebunden ist. Bei Moodle handelt es sich um ein weitverbreitetes open-source Lernmanagement System, das weitreichende Online-Funktionalitäten aufweist, quelloffen und kostenfrei verfügbar ist [moo-19] [Mos-13]. Konkret ist die von den Autoren erstellte Website als Benutzungsschnittstelle in Moodle integriert. Dies ermöglicht dem Nutzer die direkte Verwendung von Online-Funktionalitäten wie Glossaren oder Wikis. Es ist durchaus vorstellbar, dass Lernende durch Verwendung von eQual 2.0 implizit ein Verständnis über die Durchführung von Simulationsstudien aufbauen können.

Tabelle 3.2: Handlungsbedarf

Anforderungen:			Vorarbeiten aus Kapitel 2.5:								
			Serious Games	eQual 2.0	Web-SCM	Sim-Wertstrom	Manufacturing in Python	Projekt simKMU	Simulationsassistenzsysteme	Textuelle Aufgaben	ClouDES
①	1	Einarb. in Simulationsinstrument	○	○	○	●	○	○	○	●	●
	2	Einarb. in Bausteine/Objekte, Ebene 3	○	○	●	●	○	●	○	●	○
	3	Einarb. in Programmiersp. z.I.v.Sim.-mod.	○	○	○	○	○	○	○	●	○
	4	Durchführung von Simulationsstudien	○	●	●	○	○	●	●	●	●
②	5	Materialflusssystem/Fließpri. und Pull-Pri.	●	●	●	●	○	●	●	●	○
	6	Multiagentensystem	●	○	●	○	○	○	○	●	○
③	7	Nutzung von Spielelementen	●	○	○	○	○	○	○	○	●
	8	Beachtung unterschiedlicher Spieler-Typen	●	○	○	○	○	○	○	○	○
	9	Beachtung des „Flows“	●	○	○	○	○	○	○	○	○
	10	Beachtung des Instruktionsdesigns	●	●	○	○	○	●	○	●	●
④	11	Cloud-IT-Architektur	●	●	●	●	○	●			●
	12	Interaktion mit GUI auf mobilem Gerät	○	○	○	●	○	○	○		●
	13	Interaktion mit Simulationsinstrument	●	○	○	●	●	○	○	●	●
	14	Online-Funktionalitäten	●	●	●	○	●	●	○		●
	15	Animation	●					●			●
	16	Gestaltung/Interaktion ähnlich Sim.-Instru.		○	○	○		○	○		○
	17	Einbeziehung von Lernfabrik	○	○	○	○	○	○	○	○	○
Vergleichende Einordnungen:											
Vorwissen	Anfänger		●	●	●	●	○	●	○	●	●
	Fortgeschrittener Anfänger		●	●	●	●	●	●	○	●	●
	Fortgeschrittener		○	○	●	●	●	●	●	●	●
	Erfahrener Fortgeschrittener		○	○	○	○	●	○	●	●	○
	Experte		○	○	○	○	●	○	●	○	○
Klasse	Allgemeines Simulationsinstrument		○			○	○	○	●	●	●
	Simulationsinstrument für MFS		○			●	●	●	●	○	●
	Simulationsinstrument für Teilber. in MFS		●			●	●	●	●	○	●

Legende:

- – voll erfüllt/richtet sich an
- ◐ – teilweise erfüllt/kann sich richten an
- – nicht erfüllt/richtet sich nicht an
- keine Angabe/nicht zutreffend
- ID – Instruktionsdesign
- MFS – Materialflusssystem

- ① – Lernziel
- ② – Anwendungsbereich
- ③ – Motivation und Instruktionsdesign
- ④ – Umsetzung und Betrieb der Lernumgebung

Webbasierte Supply Chain Simulation: Zielgruppe für die Nutzung von Simulationsinstrumenten aus dem Bereich des SCM sind meist Menschen mit geringem Vorwissen in der Simulationstechnik. Spezialisten aus dem Anwendungsbereich des SCM modellieren in einem vom Simulationsinstrument abgekoppelten GUI Produktionsnetze. Die im Simulationsinstrument verwendeten Begriffe orientieren sich am Anwendungsbereich des SCM. Eine gewisse Nutzerführung im GUI trägt sicherlich dazu bei, dass der Anwender ein Verständnis für die Durchführung von Simulationsstudien entwickelt. Diese Ansätze eignen sich jedoch kaum dafür, Anfänger in die Nutzung flexibler Simulationsinstrumente einzuarbeiten.

Simulationsbasierte Wertstromanalyse: Bausteinkästen für die Wertstromanalyse sind oft hoch abstrahiert. Der Anwender, der Vorwissen im Bereich der Ganzheitlichen Produktionssysteme hat, findet sich in einer ihm vertrauten begrifflichen Welt wieder. Auch das Erscheinungsbild vordefinierter Bausteine entspricht zumeist den in der gängigen Literatur verwendeten Symbolen, vgl. [VDI-2870b, S.76ff.]. Da die Modellierung meist im Simulationsinstrument stattfindet, können diese Ansätze dazu beitragen, Nutzer zu einem gewissen Grad an ein solches Simulationsinstrument heranzuführen. Voraussetzung für die Nutzung solcher Bausteinkästen ist, dass der Nutzer die Durchführung einer Wertstromanalyse beherrscht.

Manufacturing in Python: Es bleibt offen, warum sich das im Entwicklungsstadium befindliche Simulationsinstrument Manufacturing in Python gerade zur Nutzung durch Anwender mit geringem Vorwissen eignen soll. Sicherlich ist eine Online-Community hilfreich, die unerfahrene Anwender durch die Bereitstellung von Bausteinen unterstützt. Findet sich jedoch kein Experte, der bereit ist, den Lernenden zu unterstützen, so müsste dieser - wie bei vielen anderen objektorientierten Simulationsinstrumenten - selbst auf die Programmierung zurückgreifen. Besonders positiv hervorzuheben ist bei diesem Ansatz die freie Zugänglichkeit und Erweiterbarkeit des Simulationsinstruments sowie von Objekten und Bausteinen durch eine Online-Gemeinschaft.

Projekt simKMU: Die grafische Oberfläche SimWebApp zur Datenein- und -ausgabe weist kaum Ähnlichkeiten zu bekannten Simulationsinstrumenten auf. Vielmehr scheint diese durch strukturierte Benutzerführung und KMU-spezifische Bausteine die Komplexität marktgängiger Simulationsinstrumente zu verringern. Dies schränkt jedoch auch die Breite möglicher Anwendungsbereiche ein. Die Möglichkeit zur Implementierung individueller Abläufe durch eine Programmiersprache ist in der Oberfläche für den Anwender nicht vorgesehen. Die Verwendung von Spielelementen ist im Projektbericht nicht dokumentiert [Wim-11]. Die Nutzung mobiler Apps ist in diesem Projekt nicht Gegenstand.

Simulationsassistenzsysteme: Simulationsassistenzsysteme tragen durch klare Benutzerführung dazu bei, dass Simulationsstudien strukturiert durchgeführt werden. Die Benutzerführung ist gemäß etablierter Vorgehensmodelle zur Durchführung von Simulationsstudien strukturiert. Ein Beispiel ist in der VDA RICHTLINIE 4811 dokumentiert [VDA-4811]. Assistenzsysteme fokussieren ein Teilgebiet eines Anwendungsbereichs, im Beispiel von MAYER UND MIESCHNER (2015) Fragestellungen in der Automobilindustrie [May-15a]. Vor dem Hintergrund der Standardisierung der Durchführung von Simulationsstudien ist ein Einsatz dieser Assistenzsysteme sicherlich sehr

sinnvoll. Hinsichtlich der Lernförderlichkeit für Simulationsinstrumente können sie kaum einen Beitrag leisten, da Kenntnisse in der Bedienung von Simulationsinstrumenten vorausgesetzt werden.

Textuelle Aufgaben: Textuelle Aufgaben scheinen dazu geeignet zu sein, Lernenden die Anwendung eines exemplarischen Simulationsinstruments nahe zu bringen. Im Handel verfügbare Bücher sind in der Regel als gut strukturierte Sammelwerke aufeinander aufbauender Aufgaben inklusive Erklärungen einzuschätzen. Allerdings sind Bücher im Zeitalter der „Digital Natives“ kaum als ein attraktives Medium anzusehen, das insbesondere junge Lernende dazu motiviert, sich in Software einzuarbeiten [Pre-01]. Weiterhin können die Lernenden auch kaum auf die praktischen Aufgaben vorbereitet werden, die bei der Durchführung von Simulationsstudien zu lösen sind. Beispielsweise kann die Wahl eines angemessenen Abstraktionslevels kaum mit diesen textuellen Aufgaben trainiert werden. Meist sind alle relevanten Daten im Hinblick auf die Problemstellung gegeben und eben nur relevante Daten in der Aufgabenstellung benannt. Aus diesen Gründen müssen durch die Lernenden selten Annahmen im Sinne einer Idealisierung oder Reduktion getroffen werden. Darüber hinaus können den Studierenden durch textuelle Aufgaben kaum die Auswirkungen mangelhafter Validierung des Simulationsmodells aufgezeigt werden, da die simulativ ermittelten Verbesserungen nicht in die Realität implementiert werden. Spielelemente wie Punkte oder Level kommen in den analysierten Fachbüchern nicht zum Einsatz.

ClouDES: Die Autoren führen nicht aus, inwiefern das Spielkonzept auf andere Simulationsinstrumente übertragbar ist. Die Möglichkeit zur Nutzung von Programmiersprachen zur Implementierung von Simulationsmodellen ist in ClouDES nicht gegeben. Aufgrund der fehlenden Flexibilität eignet sich das Simulationsinstrument ClouDES kaum zur Analyse komplizierter oder komplexer Materialflusssysteme. In dem dargelegten Ansatz kommt lediglich ein Spielelement zum Einsatz: das Erzählen einer „Geschichte“. Eine Differenzierung von Spielelementen nach Spielertypen ist nicht erfolgt. Des Weiteren fokussieren die Entwickler von ClouDES mit ihrem allgemeinen Simulationsinstrument ausschließlich die Zielgruppe Schüler.

4 Lernumgebung für ereignisdiskrete Simulation

Als Beitrag zur Einarbeitung in ereignisdiskrete Simulation ist im Folgenden eine gamifizierte Lernumgebung konzipiert. Der Fokus ist auf die Einarbeitung in ein Simulationsinstrument zur Durchführung von Simulationsstudien im Anwendungsbereich der Materialflusssysteme gerichtet. Abbildung 4.1 gibt einen Überblick über die innovativen Bestandteile des Konzepts. Diese umfassen inkrementell komplizierter werdende Simulationsmodelle (IKS), visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen (VBPIS) sowie das Vorgehen zum erfahrungsbasierten Erlernen der Durchführung von Simulationsstudien (EEDS). IKS und VBPIS sind in eine skalierbare Cloud-IT-Architektur eingebettet, die auch eine Animation ermöglicht.

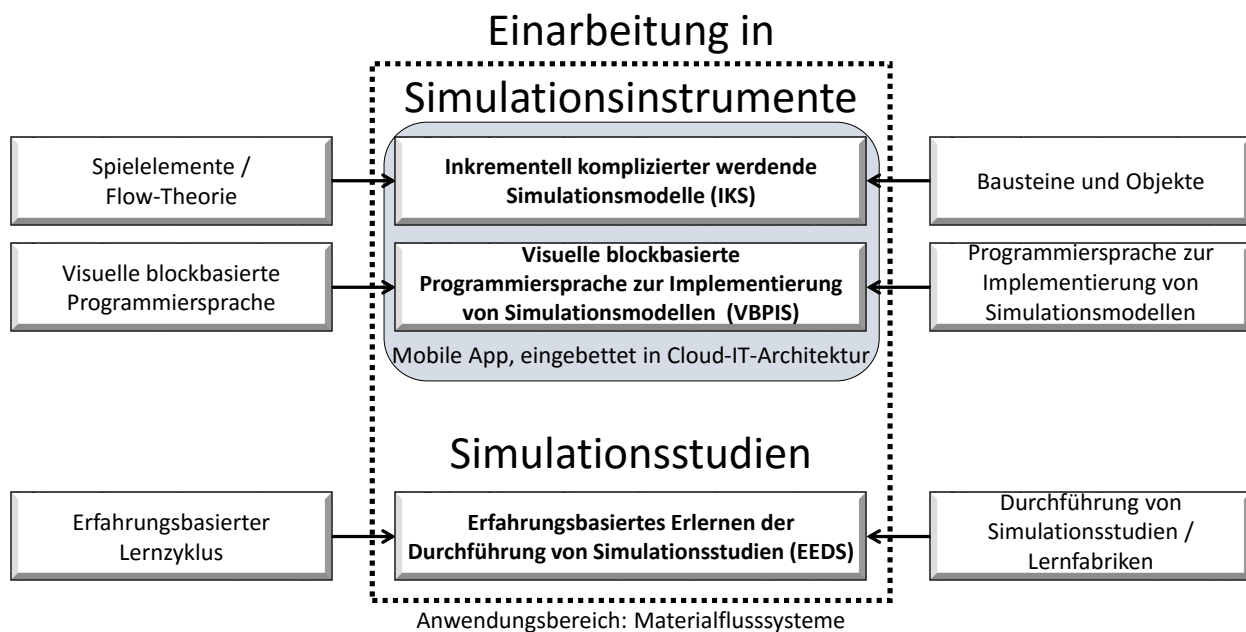


Abbildung 4.1: Überblick über die Bestandteile des Konzepts

Im Folgenden ergibt sich die Struktur der Kapitel anhand der Lerninhalte zur Einarbeitung in das Simulationsinstrument (Kapitel 4.1) zur Durchführung von Simulationsstudien (Kapitel 4.2). Im Kapitel 4.3 sind die vorgestellten Bestandteile schließlich im Sinne eines Spiel- und Lernverlaufs zusammengeführt.

Das im Abschnitt 2.4.1 beschriebene ADDIE-Modell ist dieser Arbeit zugrunde gelegt worden (Anforderung 10). Die *Analyse* ist in den Kapiteln 1 und 2, der *Entwurf* in diesem Kapitel 4, die *Umsetzung*, *Implementierung* und *Evaluierung* im Kapitel 5 dokumentiert.

4.1 Einarbeitung in Simulationsinstrument

Lernende sollen in ereignisdiskrete Simulationsinstrumente orts- und zeitunabhängig eingearbeitet werden (Anforderung 1). Vor diesem Hintergrund wird eine Cloud-IT-Architektur konzipiert (Kapitel 4.1.1), die diese Anforderung erfüllen soll. Diese Architektur dient als Basis für den Betrieb der IKS und VBPIs (Kapitel 4.1.2).

4.1.1 Cloud-IT-Architektur

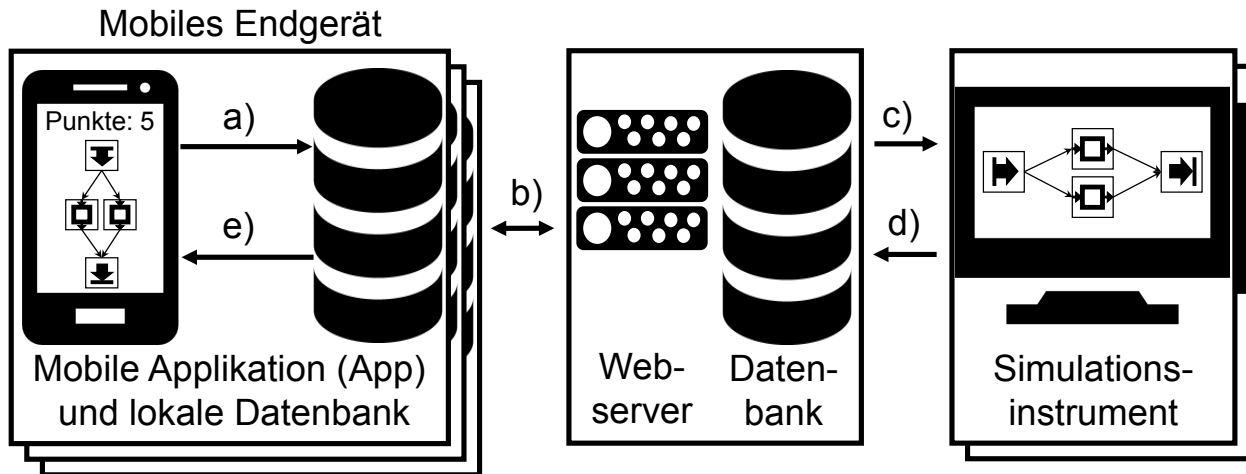
Aus Entwicklersicht besteht die vorgeschlagene Cloud-IT-Architektur aus einer *lernförderlich gestalteten App*, einem *Webserver*, einer *Datenbank* und einem *Simulationsinstrument* (Anforderung 11). Schematisch ist das Zusammenspiel in Abbildung 4.2 dargestellt⁹. Die mobile App wird durch Lernende bedient (Anforderung 12). Die Lernenden bauen spielerisch Simulationsmodelle auf und stellen so Eingabe- und Experimentdaten bereit. Die Modellierungsumgebung der App ist hinsichtlich der grafischen Gestaltung an das Simulationsinstrument angelehnt (Anforderung 16). Attribute der Objekte und Bausteine wie Typ, Position in x und y Richtung und Bearbeitungszeit werden in eine lokale Datenbank übernommen, siehe Abbildung 4.2a. Mit Attributen des Simulationsmodells, wie dem Simulationszeitraum, wird ebenso verfahren. Die *lokale Datenbank* wird in einem definierten Zeitintervall unter Nutzung von Skripten, die auf einem *Webserver* hinterlegt sind, mit einer somit global erreichbaren Datenbank über das Internet synchronisiert, vgl. Abbildung 4.2b. Ein Algorithmus in dem Simulationsinstrument generiert basierend auf den in der Datenbank hinterlegten Daten automatisch ein Modell und startet einen Simulationslauf, vgl. Abbildung 4.2c. Es handelt sich um einen parametrischen Ansatz der (teil-)automatischen Modellgenerierung, vgl. Kapitel 2.3.2.3. Zum Ende werden Simulationsergebnisdaten in die Datenbank übertragen, siehe Abbildung 4.2d. Lernende können auf diese Daten zugreifen, um sie auszuwerten, vgl. Abbildung 4.2e.

Lernende müssen ein gängiges mobiles Endgerät wie ein Smartphone oder ein Tablet mit aktuellem Betriebssystem besitzen. Eine Internetverbindung ist darüber hinaus erforderlich. Diese Internetverbindung muss nicht dauerhaft aktiv sein, sondern nur, wenn der Nutzer eine Synchronisation wünscht. Somit eignet sich die Architektur auch zur Nutzung bei instabiler Internetverbindung. Optional können Eingabegeräte wie ein Bluetooth Game-Pad oder eine Brille zur stereoskopischen/immersiven Projektion genutzt werden.

Die in Abbildung 4.2 skizzierte Cloud-IT-Architektur ist aus mehreren Gründen vorteilhaft:

Mobilität: Der Webserver ermöglicht die Nutzung mobiler Endgeräte unterschiedlicher Hersteller und Betriebssysteme. Beispielsweise können Lernende ihnen bereits vertraute Endgeräte wie das persönliche Smartphone oder Tablet verwenden, welches sie auch für weitere Zwecke wie zum Surfen oder

⁹Das in Abbildung 4.2 schematisch dargestellte Zusammenspiel wird im Verlauf der Arbeit um die in der prototypischen Implementierung gewählten Softwarepakete sowie um den Ablauf zur Übersetzung der VBPIs zwischen App und Simulationsinstrument vertieft.



Legende:

- a) – Attribute von Objekten/Bausteinen und Attribute des Simulationsmodells werden in die Datenbank geschrieben.
- b) – Synchronisation der Datenbanken über das Internet.
- c) – Relevante Daten werden von der Datenbank in das Simulationsinstrument übertragen. Das Simulationsinstrument generiert automatisch das Simulationsmodell und startet den Simulationslauf.
- d) – Simulationsergebnisdaten werden in die Datenbank übertragen.
- e) – Lernende werten die Simulationsergebnisdaten aus.



– App, die auf einem Smartphone betrieben wird

– Bausteine/Objekte

– Datenbank

– Webserver

– Computer/Server mit Simulationsinstrument

Abbildung 4.2: Cloud-IT-Architektur: Zusammenspiel zwischen App, Datenbank und Simulationsinstrument nach [Mül-17a]

Messaging nutzen. Ereignisdiskrete Simulationsinstrumente unterschiedlicher Hersteller, die eine (teil-)automatische Modellgenerierung unterstützen und geeignete Schnittstellen haben, können genutzt werden. Vor dem Hintergrund des Lernziels der Einarbeitung in ein Simulationsinstrument ist an dieser Architektur vorteilhaft, dass Lernende das automatisch generierte Simulationsmodell im Simulationsinstrument nachvollziehen können (Anforderung 13).

Leistungsfähigkeit/Erweiterbarkeit: Aufgrund beschränkter Speicherkapazität und Rechenleistung handelsüblicher mobiler Endgeräte wie Smartphones oder Tablets wäre eine Ausführung der Rechenoperationen hierauf mit Risiken verbunden. Aus diesen Gründen sind die Simulationsläufe an ein externes Simulationsinstrument auszulagern, in welchem sich das Simulationsmodell auf Basis der von der App übergebenen Daten automatisch generiert. Das Simulationsinstrument wird auf einer geeigneten IT Hardware betrieben. Die Cloud-IT-Architektur ist skalierbar. Die Anzahl mobiler Apps sowie die Anzahl der Simulationsinstrumente sind nahezu beliebig veränderbar. Auf Anwenderseite kann sich durch eine Vielzahl gleichzeitig mit der App interagierenden Personen eine Warteschlange nicht begonnener Simulationsaufträge vor dem Simulationsinstrument bilden. Um einen Engpass bei der Abarbeitung der Simulationsaufträge beim Simulationsinstrument zu vermeiden, lassen sich weitere Simulationsinstrumente zuschalten. Jedem Simulationsinstrument wird dabei gemäß dem FIFO-Prinzip ein Auftrag zur Simulation zugewiesen. Die Simulationsinstrumente werden auf Computern betrieben, die räumlich verteilt angeordnet sein können.

Wartung und Weiterentwicklung: Durch Veröffentlichung von App-Updates, Aktualisierung der Datenbasis in der Datenbank oder von Skripten auf dem Webserver, lassen sich Erweiterungen der Software schnell und unkompliziert durch einen Entwickler vornehmen. Software-Bugs können ebenso leicht behoben werden.

Lizenz Simulationsinstrument: Die Anzahl erforderlicher Lizenzen für Simulationsinstrumente ist bei dieser Architektur von der Anzahl der genutzten Simulationsinstrumente abhängig. Mindestens ein Simulationsinstrument ist erforderlich. Die Anzahl der mobilen Endgeräte ist nahezu unbegrenzt.

Allerdings ergeben sich auch *Herausforderungen* bei der Umsetzung und dem Betrieb der Architektur:

Animation: Eine lokale Simulation und Animation, wie sie im Kapitel 2.3.2.4 beschrieben wurde, ist durch die verteilte Cloud-IT-Architektur nicht möglich, da die App als GUI dient und nicht mit einem Simulatorkern ausgestattet ist. Folglich ist die Animation in Echtzeit kaum umsetzbar. Zwar wäre ein Ansatz der Fern-Simulation bzw. der hybriden Simulation und Animation vorstellbar, jedoch sind diese mit Blick auf die Implementierung als sehr aufwändig und im Betrieb als fehleranfällig einzuschätzen. Beispielsweise sind instabile Datenverbindungen zu berücksichtigen.

Eine Animation des Simulationsfortschritts in der App stellt sicher, dass Lernende die Funktionalitäten und Abläufe im Simulationsmodell verstehen können [Wen-08a, S.148]. Anforderung 15 muss vor dem Hintergrund der beschriebenen Herausforderung so umgesetzt sein, dass der Lernende in der App die Animation des von ihm gestalteten Modells möglichst zeitnah nachvollziehen kann.

Datennutzung: Durch mobile Datennutzung können dem Anwender Kosten entstehen.

Update Simulationsinstrument: Durch die Architektur ist gewährleistet, dass stets auf ein aktuelles Simulationsinstrument zugegriffen wird. Im Falle von Hersteller-Updates für das Simulationsinstrument kann es erforderlich sein, die Struktur der Datenbank, die Skripte auf dem Webserver und/oder die mobile Anwendung zu aktualisieren.

Kern der Architektur ist die *gamifizierte App*. Lernende bauen in ihr spielerisch (Anforderungen 7-10) *Inkrementell komplizierter werdende Simulationsmodelle (IKS)* auf, um sie in die Nutzung von Bausteinen und Objekten einzuarbeiten (Anforderung 2). *Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen (VBPIS)* kommt in der App zum Einsatz, um den Nutzern die Syntax und Semantik einer beispielhaft gewählten Programmiersprache zur Implementierung von Simulationsmodellen zu vermitteln (Anforderung 3). Es wird ein *implizites Erlernen* eines objektorientierten Simulationsinstruments angestrebt. Der grafisch-interaktive Modellaufbau ist in der App so zu gestalten, dass die Interaktion unter Nutzung von Bausteinen/Objekten ähnlich zur Modellierungsumgebung des exemplarischen Simulationsinstruments gestaltet ist (Anforderung 16).

4.1.2 Lernförderliche Elemente

Unter Gamification versteht man die Anwendung von Elementen des Spielens in einem nicht-spielerischen Kontext [Det-11]. Ein Spielelement können Punkte sein. Punkte dokumentieren sichtbar für eine Gruppe an Spielern den Fortschritt des einzelnen Spielers. Gamification lässt sich dazu einsetzen, die Motivation zu steigern, um Menschen z.B. zu einer häufigen Nutzung der Lernumgebung zu motivieren. Punkte, die der Nutzer erwirbt, indem er Level abschließt, sind zur Belohnung zu vergeben. Punkte werden zunächst automatisch dem Punktekonto des Spielers gutgeschrieben, im späteren Verlauf des Spiels durch Mitspieler. Diese Punkte sind in einem sogenannten Leaderboard öffentlich sichtbar. Abzeichen machen über den Abschluss eines Levels hinaus Verdienste des Spielers für den Spieler und Mitspieler sichtbar. Abzeichen werden nach offenkundigen und nicht offenkundigen Kriterien an Spieler verliehen. Offenkundig bezeichnet Kriterien zur Verleihung eines Abzeichens, die dem Lernenden bekannt sind. Kennt der Lernende das Kriterium nicht, so ist es nicht offenkundig. Der Erwerb einer gewissen Punktzahl kann ein Kriterium für die Verleihung eines Abzeichens sein. Im Gegensatz zu den Punkten können mit Abzeichen Spieler belohnt werden, die ein Kriterium zuerst oder innerhalb eines festgelegten Zeitraums erfüllen. Die Lernenden werden unter spielerischen Gesichtspunkten dazu motiviert, IKS aufzubauen. VBPIS kommt zum Einsatz, um sie in die Programmierung einzuarbeiten.

4.1.2.1 Inkrementell komplizierter werdende Simulationsmodelle

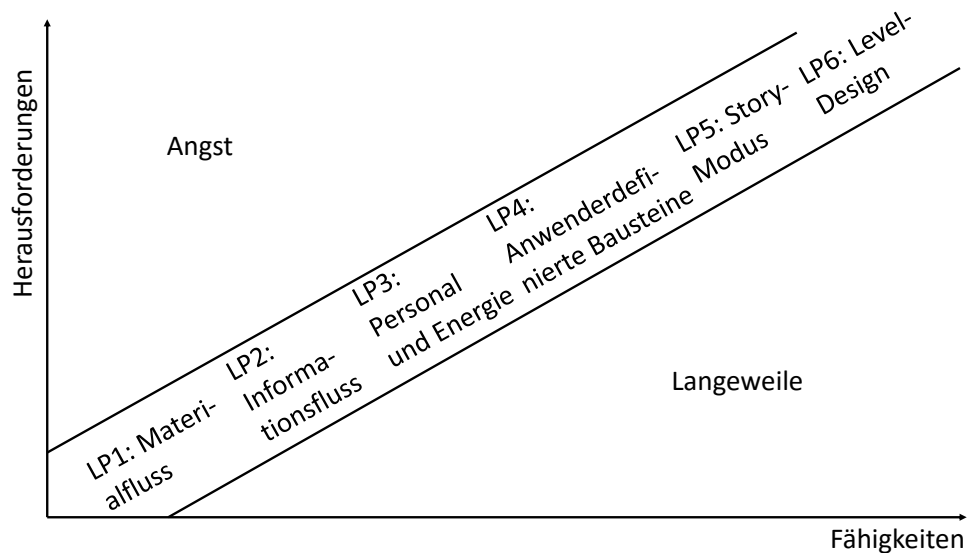
IKS dienen der schrittweisen Einarbeitung in die grafisch-interaktiven Modellierung unter Nutzung von Bausteinen/Objekten. Im Einklang von Schwierigkeitsgrad der Aufgabenstellung mit den Fähigkeiten des individuellen Spielers (Anforderung 9) sind zunächst Simulationsmodelle zu erstellen, die eine geringe An-

zahl an bislang unbekannten Bausteinen/Objekten umfassen [Mül-17a]. Diese bestehen beispielsweise aus einer Quelle, einer Senke und einer Bearbeitungsstation. Levelbasiert wird der Schwierigkeitsgrad erhöht, indem neue Bausteine und Objekte eingeführt werden, z.B. eine Montagestation. Je Level ist eine Aufgabe durch den Lernenden zu lösen. Die Einarbeitung in die vom zu erlernenden Simulationsinstrument unterstützte Programmiersprache zur Implementierung von Simulationsmodellen ist ein Teil des Lernprozesses. Zur Lösung schwierigerer Aufgaben müssen Lernende Programme mit einer solchen Sprache formalisieren.

Mit den IKS ist ein *induktives Vorgehen* verbunden. Während bei einem deduktiven Vorgehen zuerst die Vermittlung theoretischer Inhalte im Vordergrund steht, richtet sich der Fokus bei einem induktiven Vorgehen zunächst auf die praktische Arbeit im Sinne des Aufbaus von Simulationsmodellen und der Simulation. Bei einem induktiven Vorgehen entscheidet der Lernende, ob und welche theoretischen Inhalte, die ihm für den Aufbau des Simulationsmodells und zur Simulation fehlen, er nachschlägt. Er eignet sich in Eigeninitiative das Wissen an. Das induktive Vorgehen hat drei maßgebliche Vorteile. Das Spiel kann direkt mit Beispielen begonnen werden, ohne die häufig als trocken empfundene Theorie zu studieren. Dem Lernenden bereits bekannte Theorie muss nicht erneut studiert werden, um mit dem Spiel fortfahren zu können. Ein dezidiert Theorie-Bereich birgt den Charakter, dass es in der Lernumgebung etwas zu entdecken gibt. Der Spieler-Typ „Entdecker“ fühlt sich somit angesprochen.

Die IKS sind in Level strukturiert. Level repräsentieren kurze abgeschlossene Lerneinheiten. Jedes Level beinhaltet eine durch den Lernenden zu lösende Aufgabe. Lernende bauen in jedem Level ein Simulationsmodell auf. So lösen sie eine ihnen gestellte Aufgabe. Durch den erfolgreichen Abschluss von Leveln werden Lernende belohnt. Ähnliche Level sind thematisch in Levelpacks gruppiert. Durch speicherbare Arbeitsstände und aufgrund des geringen Zeitaufwands für die Lösung eines Levels soll die Lernumgebung zu einer Nutzung „zwischen durch“ einladen.

Die Levelpacks sind thematisch so gegliedert, dass Lernende in zunehmendem Maße die Breite des Anwendungsbereichs des Simulationsinstruments kennenlernen. Der Anwendungsbereich wird in Form von zu lösenden Aufgaben exemplarisch vertieft. Die Lernenden sind in die Abbildung von Gegenständen des Materialflusses wie Personal, Informationen, Produktionsmittel, Materialflussmittel, Informationsflussmittel, Energie und Infrastruktur einzuarbeiten [Sch-08a]. Die Levelpacks sind mit thematischen Schwerpunkten hinsichtlich der benannten Gegenstände des Materialflusses gestaltet. Konkret sind diese Schwerpunkte der Materialfluss, Informationsfluss, das Personal und die Energie, die anwenderdefinierten Bausteine, der Story-Modus und das Level-Design. Abbildung 4.3 verdeutlicht den Aufbau der überwiegend später in der Arbeit exemplarisch umgesetzten Levelpacks im Kontext der Flow-Theorie. Während der ersten drei Levelpacks wird vorwiegend eine Wissensvermittlung fokussiert. Im vierten, fünften und sechsten Levelpack steht ein Kompetenzerwerb im Fokus, indem die Lernenden maßgeblich mit Simulationsmodellen *experimentieren*. Lernende werden so dazu befähigt, Bausteine/Objekte im Bezug auf unterschiedliche Problemstellungen im Bereich der Materialflusssysteme anzuwenden. Hier werden zuvor erstellte Simulationsmodelle durch Lernende hinsichtlich gegebener KPIs verbessert. Lernende sollen dazu befähigt werden, Sensitivitätsanalysen durchzuführen. Voraussetzung zum Experimentieren mit Simulationsmodellen ist, dass diese vom Nutzer selbst aufgebaut wurden. Voraussetzung dafür ist wiederum die Kenntnis hinsichtlich des Funktionsumfangs von Bausteinen und Objekten. Hieraus resultiert der in Ab-

**Legende:**

LP – Levelpack

Abbildung 4.3: Thematische Untergliederung in sechs Levelpacks im Spannungsfeld von Herausforderungen und Fähigkeiten

bildung 4.3 dargestellte Aufbau. Im Folgenden sind die Levelpacks kurz ausgeführt und die beabsichtigte Wirkung von Spielelementen und -mechaniken auf die Spieler beschrieben.

Levelpack 1 Materialfluss: Im Levelpack 1 werden die Objekte und Bausteine zur Modellierung des Materialflusses eingeführt (Anforderung 2). Die Abbildung von Produktions- und Materialflussmitteln steht im Fokus. Die Zeit zur Bearbeitung eines Levels durch den Lernenden beträgt nur wenige Minuten. Vorausgesetzt sei ein konzentrierter Spieler. Dies ermöglicht schnelle Erfolgserlebnisse für den Spieler. Ein Video über die Bedienung der Modellierungsumgebung ist dem Nutzer zu Beginn zur Verfügung zu stellen. Dies soll dazu beitragen, dass Nutzer schnell die Gesten zur Bedienung der App kennenlernen und zügig ihr erstes Erfolgserlebnis durch einen erfolgreichen Levelabschluss generieren können. Benutzer werden mit Punkten und Abzeichen belohnt. Nach dem Abschluss des Levelpacks können die Lernenden Simulationsmodelle mit vorkonfigurierten Bausteinen und Objekten des Materialflusses aufbauen.

Levelpack 2 Informationsfluss: Levelpack 2 ist ähnlich zu Levelpack 1 konzipiert. Die Abbildung des Informationsflusses und von Informationsflussmitteln stehen im Mittelpunkt. Hier richtet sich der Fokus verstärkt auf die Einarbeitung des Lernenden in die Programmierung mit der VBPI (Anforderung 5). Die Verwendung von Tabellen zur strukturierten Verwaltung von Eingabe- und Modelldaten soll trainiert werden. Nach dem Abschluss des Levelpacks sind die Lernenden dazu befähigt, den Materialfluss unter Verwendung selbst erstellter Programme zu beeinflussen.

Levelpack 3 Personal und Energie: Analog zu der Gestaltung der Levelpacks 1 und 2 zielt das Levelpack 3 auf die Abbildung von Mitarbeitern und Energie ab. Im Sinne einer personalintegrierten

Modellierung wird erlernt, Mitarbeiter im Sinne von Ressourcen abzubilden. Die Betrachtung des Energieverbrauchs sei auf die elektrische Energie von Produktions- und Materialflussmitteln begrenzt.

Levelpack 4 Anwenderdefinierte Bausteine: Im Levelpack 4 erlernen die Anwender die Modellierung und Simulation mit anwenderdefinierten Bausteinen. Ein anwenderdefinierter Baustein wurde von einem Simulationsexperten erstellt. Beispiele sind Bausteine zur Modellierung von Multiagentensystemen oder der Bausteine des VDA-Bausteinkastens (Anforderung 6). Nach dem Abschluss des Levelpacks sind die Lernenden dazu befähigt, mit anwenderdefinierten Bausteinen zu modellieren.

Levelpack 5 Story-Modus: Im Levelpack 5 werden dem Nutzer Modelle bereitgestellt, die bereits teilweise aufgebaut sind. Dies soll den Lernenden von als lästig oder langweilig wahrgenommenen repetitiven Aufgaben der Modellierung befreien. Beispiele sind die Instanziierung von Bausteinen/Objekten. Weiterhin lassen sich dadurch Missverständnisse bei der Interpretation der Aufgabenstellungen oder Fehleingaben, z.B. bei der Parametrisierung von Attributen, vermindern. Das Ziel des Lernenden besteht darin, Änderungen am Simulationsmodell hinsichtlich des Materialflusses vorzunehmen und Sensitivitätsanalysen durchzuführen. KPIs sind zu verbessern. Die möglichen Änderungen des Materialflusses werden durch eine Geschichte eingegrenzt. Beispielsweise ergibt sich aus der Geschichte das Ziel, den Durchsatz einer simulierten Montage unter Beachtung von Rahmenbedingungen wie Schichtmodellen zu erhöhen. Hierzu muss der Lernende das gegebene Modell so verändern, dass eine Verbesserung der jeweiligen KPIs erreicht wird. Der Schwerpunkt liegt in der Erstellung und Anpassung von individuellen Abläufen unter Nutzung von Programmiersprache zur Implementierung von Simulationsmodellen. Im Levelpack 5 können die Ergebnisse kaum noch automatisiert überprüft werden. Grund ist, dass mehr als eine Lösung valide ist. Demzufolge gilt ab diesem Levelpack ein sogenannter Peer-Review (Anforderung 14). Das bedeutet, dass Level von Lernenden eingereicht und von mindestens zwei weiteren Lernenden evaluiert werden. Die Voraussetzung, dass ein Lernender einen Peer-Review durchführen darf, besteht darin, dass er das entsprechende Level bereits erfolgreich abgeschlossen hat. Nach dem Abschluss des Levelpacks wissen die Lernenden, dass ein Simulationsinstrument in unterschiedlichen Anwendungsbereichen angewendet werden kann. Sie sind dazu befähigt, den Materialfluss mit selbst erstellten Programmen zu beeinflussen. Die Lernenden können über selbst erstellte Programme Attribute von Bausteinen/Objekten manipulieren.

Levelpack 6 Level-Design: Nach Abschluss von Levelpack 5 können im Rahmen des sechsten Levelpacks von Nutzern individuelle Level erstellt werden. Genauer ausgedrückt bedeutet dies, dass Nutzer Aufgabenbeschreibungen in die App implementieren und weiteren Anwendern zur Bearbeitung zur Verfügung stellen (Anforderung 14). Um sicherzustellen, dass die Aufgaben sinnvoll und deren Lösung herausfordernd sind, ist ein System zur Punktevergabe durch die Community sinnvoll. Gute können so von schlechten Levels unterschieden werden. Nach dem Abschluss des Levelpacks haben die Lernenden die in den zuvor gespielten Levelpacks erworbenen Kompetenzen gefestigt.

Spielelemente und -mechaniken sind gestaltet, um den Benutzer für seine Aktivität und seinen Erfolg zu belohnen. Die Belohnung wird durch Spielelemente erreicht, die die Zusammenarbeit und Konkurrenz von Studierenden fördern (Anforderung 7). Spielelemente richten sich an unterschiedliche Spieler-Typen.

Die Spielmechaniken sind so gestaltet, dass Lernende motiviert die Lerninhalte konsumieren. Sie sollen häufig mit der App interagieren. Bei erfolgreichem Abschluss eines Levels oder durch erfolgten Peer-Review ist ein *Punkt* zu vergeben. Punkte wiederum erhöhen den Rang des Spielers im global verfügbaren *Leaderboard*. Mitspieler können dieses Leaderboard einsehen. Durch das Erreichen einer bestimmten Punktzahl wird dem Spieler ein *Abzeichen* verliehen. So machen Abzeichen über den Abschluss eines Levels hinaus Verdienste des Spielers für ihn und Mitspieler sichtbar. Neben der Punktevergabe sollen Abzeichen dazu motivieren, auf Ziele hin zu arbeiten. Diese Ziele ergeben sich aus den für den Spieler bekannten Kriterien zur Verleihung von Abzeichen. Bei offenkundigen Abzeichen handelt es sich im einfachsten Fall um eine Grafik, die für den Spieler sichtbar wird oder den Benutzernamen des Spielers, der für Mitspieler im Zusammenhang mit dem Abzeichen angezeigt wird. Beispielsweise kann ein Abzeichen dafür verliehen werden, wenn Levelpack 1 erfolgreich abgeschlossen wurde. Abzeichen werden für offenkundige und nicht offenkundige Kriterien verliehen. Benutzer sollen wissen, durch welche Aktionen sie *offenkundige Abzeichen* erhalten können, z.B. „Beende Levelpack 1“. So wissen die Spieler immer, welche Schritte sie unternehmen müssen, um im Spiel voranzukommen. Auch sollen sie sehen, welchem Mitspieler bereits welches Abzeichen verliehen wurde. So soll der eigene Fortschritt im Vergleich zu den Mitspielern bewertbar sein. Ein Streben nach Fortschritt des Spielers soll erzeugt werden. So soll der Spieler als Anreiz, Abzeichen zu erwerben, sehen, welchem Mitspieler diese bereits verliehen wurden. Hiervon ausgenommen sind die *nicht offenkundigen Abzeichen*, deren Kriterien zur Verleihung dem Spieler zunächst unbekannt sind. Nicht offenkundige Abzeichen motivieren Spieler zum Entdecken. Der Spieler versucht, die vermuteten Kriterien zur Verleihung des Abzeichens zu erfüllen. Die Voraussetzung dafür, dass ein Spieler einen *Peer-Review* durchführen darf, besteht in der vorangegangenen erfolgreichen Lösung der Aufgabe, die zum Review bereitsteht. Bei einem Peer-Review wird die Aufgabenlösung eines Lernenden durch einen zweiten Lernenden evaluiert, der diese Voraussetzung erfüllt. Für jeden erfolgten Peer-Review erhält der Mitspieler, dessen Lösung begutachtet wurde, durch den Reviewenden Punkte. Der Reviewer erhält je absolviertem Review Punkte.

Die Spielelemente sind so zu wählen, dass unterschiedliche Spieler-Typen angesprochen werden (Anforderung 8). BARTLE (1996) unterscheidet die Spieler-Typen *Sammler*, *Entdecker*, *Socializer* und *Zerstörer*, vgl. Kapitel 2.4.2.1. Durch Beachtung dieser vier Spieler-Typen bei der Wahl der Spielelemente kann die Anwendbarkeit der Entwicklungen für eine Gruppe von Studierenden sichergestellt werden. Im Folgenden sind je Spieler-Typ zwei Beispiele für die beabsichtigte Wirkung der Spielelemente Punkte, Leaderboard, offenkundiges Abzeichen, nicht offenkundiges Abzeichen und Peer-Review auf den jeweiligen Spieler-Typen benannt.

Der Sammler möchte möglichst viele Level spielen und Reviews vollziehen, um viele *Punkte* zu sammeln. So kann der Sammler einen hohen Rang im Leaderboard erreichen. Der Entdecker nimmt das Punktesammeln als erforderlich wahr, um neue Funktionen in der Spiel-Umgebung, wie den Peer-Review freizuschalten und dessen Funktionen entdecken zu können. Das *Leaderboard* gibt dem Socializer Anlass dazu, Gespräche mit anderen Spielern hinsichtlich der Erhöhung des Rangs zu führen. Für den

Zerstörer gibt das Leaderboard wertvolle Hinweise darauf, inwiefern Mitspieler zu einer Bedrohung werden. Der Sammler möchte möglichst alle *offenkundigen Abzeichen* erhalten. Er liest sich die Kriterien für die Verleihung aufmerksam durch und versucht, diese schnell zu erfüllen. Von Abzeichen, die nur einer begrenzten Anzahl an Spielern verliehen werden, z.B. den ersten fünf, die Levelpack 1 abschließen, fühlt sich der Zerstörer sehr angespornt. Durch die Verleihung solch eines „knappen Gutes“ erhöht er seinen Rang für alle ersichtlich. *Nicht offenkundige Abzeichen*, also solche, bei denen die Kriterien zur Verleihung den Spielern zunächst unbekannt sind, wirken insbesondere auf den Entdecker und Zerstörer motivierend. Der Entdecker sucht in der App versteckte Menüs oder versucht so schnell wie möglich ein Levelpack abzuschließen, wenn er vermutet, dafür ein Abzeichen zu erhalten. Der Socializer nimmt das Freischalten von Abzeichen als ein notwendiges Übel wahr, um nicht den Anschluss an die Mitspieler zu verlieren und das Repertoire an Themen für eine Interaktion mit ihnen zu füllen. Der *Peer-Review* gibt dem Socializer Anlass dazu, mit anderen über womöglich gute oder schlechte Ansätze anderer Spieler zu sprechen. Der Zerstörer kann seine erlangte Macht im Peer-Review dazu nutzen, anderen Spielern zu zeigen, was diese gut oder schlecht machen.

4.1.2.2 Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen

Die VBPIs vereint die grafischen Eigenschaften der visuellen blockbasierten Programmiersprache mit Funktionen von objektorientierter Programmiersprache zur Implementierung von Simulationsmodellen. Programmiersprachen zur Implementierung von Simulationsmodellen unterstützen in Simulationsinstrumenten typische Funktionen wie die Ereignisverwaltung oder Datensammlung [VDI-3633a, S.31f.]. In der Regel kommen diese Sprachen zum Einsatz, wenn individuelle Abläufe in einem Simulationsmodell abgebildet werden sollen, die sich nicht oder nur aufwändig durch vorkonfigurierte Bausteine und Objekte beschreiben lassen. Visuelle blockbasierte Programmiersprache ermöglicht es Anfängern in der Programmierung, sich in die Nutzung von Programmiersprachen einzuarbeiten.

Die VBPIs ist im Hinblick auf die visuellen und blockbasierten Eigenschaften von Scratch inspiriert [Scr-19]. Das Ziel besteht darin, den zukünftigen Simulations-Nutzer in die Programmierung am Beispiel der Syntax einer Programmiersprache zur Implementierung von Simulationsmodellen einzuarbeiten. Es ist ein Compiler erforderlich, der das vom Lernenden im Block-Editor der App entwickelte Programm in die vom Simulationsinstrument unterstützte Programmiersprache zur Implementierung von Simulationsmodellen übersetzt.

Im Vergleich zu den im Stand der Wissenschaft und Technik benannten visuellen blockbasierten Programmiersprachen wie Scratch oder Google Blockly weist die VBPIs Eigenschaften objektorientierter Programmiersprachen zur Implementierung von Simulationsmodellen auf [Goo-19b] [Scr-19]. So sind neben Blöcken, mit denen sich Bedingungen, Schleifen, Variablendeklarationen, Objektpfade und Operatoren abbilden lassen, simulationsspezifische Erweiterungen dieser erforderlich:

Zugriff auf Simulationsobjekte und -bausteine: Objekte, sowohl temporär als auch permanent, und Bausteine, sowohl stationär als auch mobil, müssen durch den Programmierer adressierbar sein.

Zugriff auf Attribute von Objekten und Bausteinen sowie simulationsspezifische Funktionen:

Das Lesen und Manipulieren von Attributen von Objekten und Bausteinen ist eine wichtige Voraussetzung, um das Verhalten von Objekten und Bausteinen durch Programmcode zu ändern. Auch das Ausführen vorkonfigurierter Abläufe wie das Transferieren eines temporären Objekts muss unterstützt werden.

Die VBPIs ist spezifisch für jedes Simulationsinstrument und die darin unterstützte Programmiersprache zur Implementierung von Simulationsmodellen zu gestalten. Die VBPIs, die in der App zum Einsatz kommt, weist also syntaktische Ähnlichkeiten zur Programmiersprache zur Implementierung von Simulationsmodellen auf, die erlernt werden soll (Anforderung 16). Es gilt zu bedenken, dass die Syntax der jeweiligen Programmiersprache zur Implementierung von Simulationsmodellen abhängig vom genutzten Simulationsinstrument unterschiedlich definiert ist. An dieser Stelle der Arbeit ist folglich ein allgemein formuliertes Konzept beschrieben. Dieses ist im Verlauf der Arbeit durch ein ganz konkretes Umsetzungsbeispiel vertieft.

Der angestrebte Verlauf des Lernens ist in der Tabelle 4.1 im Hinblick auf die Unterteilung in Levelpacks beschrieben. Während in den ersten beiden Levelpacks primär das Erlernen der Syntax im Vordergrund steht, richtet sich der Fokus in den folgenden Levelpacks verstärkt auf die Semantik. In den ersten Levelpacks sind Aufgaben an den Lernenden zu stellen, für deren Lösung er in der Aufgabenstellung abgedruckten Quellcode in der VBPIs repliziert. In den höheren Levelpacks sind dem Lernenden Pseudocode oder nur noch textuelle Beschreibungen der gewünschten Funktionalitäten zur Verfügung zu stellen. Während zu Beginn Wissen über die korrekte Syntax aufgebaut wird, sind die späteren Levelpacks auf einen Kompetenzaufbau hinsichtlich der Syntax und Semantik gerichtet. In diesem Kontext bezeichnet Kompetenz die Anwendung des Wissens über die Syntax der Programmiersprache zur Implementierung von Simulationsmodellen zur Lösung von Aufgaben im Sinne der IKS. An den Simulationsergebnisdaten sind

Tabelle 4.1: Lernziele mit VBPIs im Kontext der Levelpacks

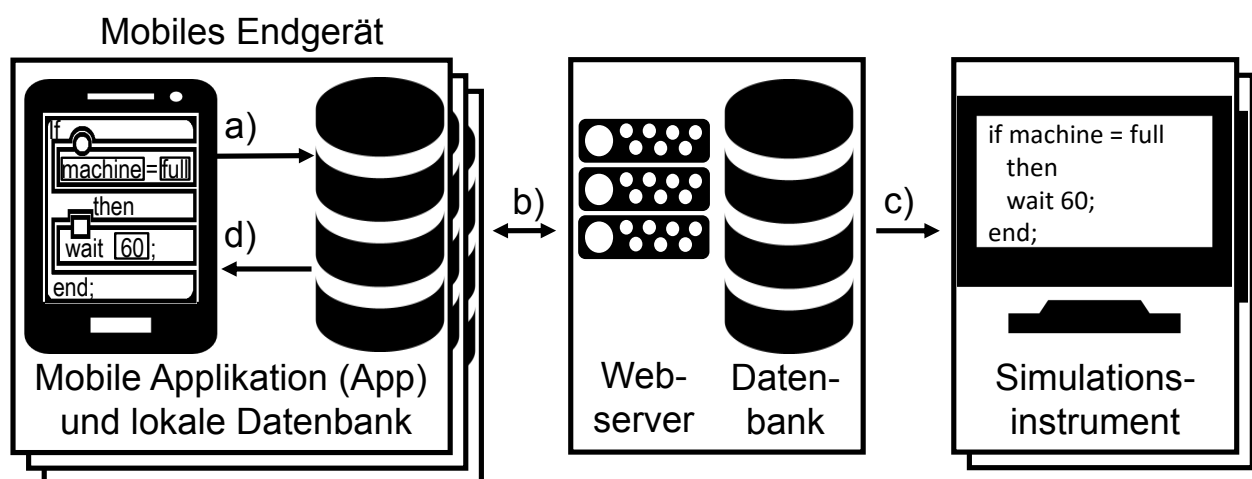
	Programmierung		Lernziel	
	Syntax	Semantik	Wissen	Kompetenz
Levelpack 1	●	◐	●	○
Levelpack 2	●	●	●	○
Levelpack 3	◐	◐	●	○
Levelpack 4	◐	◐	◐	●
Levelpack 5	◐	●	○	●
Levelpack 6	◐	●	○	●

Legende:

- – fokussiert
- ◐ – teilweise fokussiert
- – nicht fokussiert

die durch den Lernenden programmierten Verbesserungen des Materialflusses messbar. Diese Simulationsergebnisdaten geben dem Lernenden ein Feedback hinsichtlich der Korrektheit und Wirksamkeit des durch ihn implementierten Ablaufs.

Generisch lässt sich auch die Cloud-IT-Architektur der VBPIs formulieren. Abbildung 4.4 zeigt - angelehnt an die in Abbildung 4.2 gezeigte Cloud-IT-Architektur für den Betrieb der Lernumgebung - die Architektur zum Betrieb der VBPIs. Lernende bauen in dem Block-Editor der App durch Instanziierung von Blöcken, die in der Block-Bibliothek bereitgestellt sind, ein Programm auf. Dieser Block-Editor kann als ein Modellelement verstanden werden, das der Benutzer in der grafischen Modellierungsumgebung der App bei Bedarf instanziiert. Es ist ein Compiler erforderlich. Dieser übersetzt den Code so, dass er in der Datenbank strukturiert gespeichert werden kann, vgl. Abbildung 4.4a. Wie oben beschrieben wird die lokale mit der global erreichbaren Datenbank synchronisiert, vgl. Abbildung 4.4b. Das Simulationsinstrument lädt die Daten in Tabellenform, vgl. Abbildung 4.4c. Während der (teil-)automatischen Modellgenerierung findet im Simulationsinstrument die Übersetzung in die Programmiersprache zur Implementierung von Simulationsmodellen statt. Die in der Abbildung 4.4d aufgezeigte Übersetzung zwischen der lokalen Datenbank und App ist erforderlich, damit der Benutzer auch bereits in der App erstellte Programme wieder im Block-Editor öffnen und manipulieren kann.



Legende:

- a) – Das im Block-Editor der App aufgebaute Programm wird übersetzt und in der Datenbank strukturiert gespeichert.
- b) – Synchronisation der Datenbanken über das Internet.
- c) – Relevante Daten werden von der Datenbank in das Simulationsinstrument übertragen. Das Simulationsinstrument übersetzt die Daten in die vom Simulationsinstrument unterstützte Programmiersprache zur Implementierung von Simulationsmodellen und startet den Simulationslauf.
- d) – Lernende werten die Simulationsergebnisdaten aus.



- Block
- für bereits eingeführte Grafiken siehe Legende der Abbildung 4.2

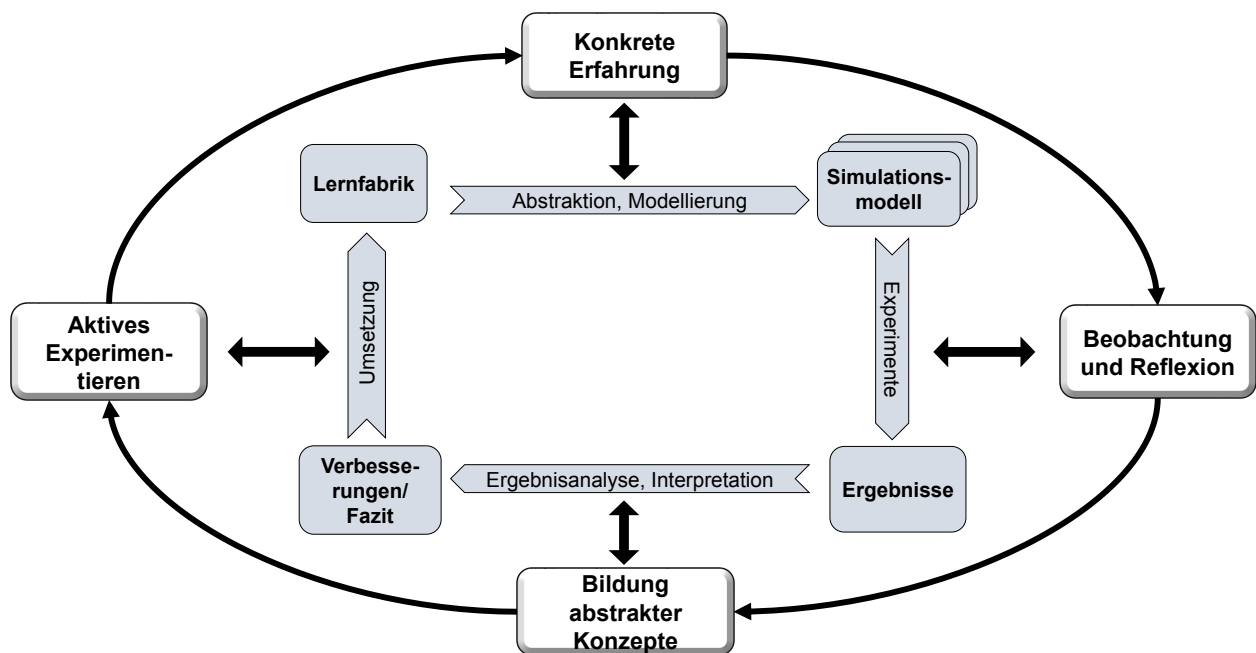
Abbildung 4.4: VBPIs im Kontext der in Abbildung 4.2 vorgestellten Cloud-IT-Architektur

4.2 Einarbeitung in die Durchführung von Simulationsstudien

Das Vorgehen zum erfahrungsbasierten Erlernen der Durchführung von Simulationsstudien (EEDS) orientiert sich an KÜHNS Simulationszyklus, der konkret unter Nutzung einer Lernfabrik im engeren Sinne interpretiert wird (Anforderung 17) [Küh-06]. Das Vorgehen wird durch Lernende angewendet. Durch Anwendung des Vorgehens in einer Lernfabrik soll der Aufbau von Erfahrungen beim Lernenden in der Durchführung von Simulationsstudien unterstützt werden. Die Organisation der Lernfabrik ist mit Blick auf die Methoden des Fließprinzips und Pull-Prinzips veränderlich. Die Gestaltungsprinzipien umfassen bewährte Methoden zur Verbesserung von Materialflusssystemen, die häufig in der Praxis Anwendung finden und meist in relevanten Lehrveranstaltungen der benannten Zielgruppe behandelt werden. Beispiel für Methoden sind der One Piece Flow oder das U-Layout (Fließprinzip), Kanban oder der Supermarkt (Pull-Prinzip) [VDI-2870a]. Lernende sollen somit in die Durchführung von Simulationsstudien im Kontext der Materialflusssysteme eingearbeitet werden (Anforderung 4). Zumal in Lernfabriken maßgeblich ein erfahrungsbasiertes Lernen stattfindet, ist im Folgenden der Lernzyklus nach KOLB ET AL. (1971) zur Illustration verwendet [Kol-71]. Das Vorgehen zum EEDS ist in Abbildung 4.5 abgedruckt. Als das Reale Materialflusssystem ist die *Lernfabrik* zu interpretieren. *Simulationsmodelle* sind als abstrahierte digitale Abbildungen der Gegenstände des Materialflusses wie Informationen, Personal oder Produktionsmittel zu verstehen. Die *Ergebnisse* beziehen sich auf Daten, die durch Experimente mit dem digital nachgebildeten Materialflusssystem gewonnen wurden. *Verbesserungen* sind *Ergebnisse*, die durch den Lernenden im Hinblick auf deren Umsetzbarkeit in der Lernfabrik interpretiert wurden.

Das Lernziel besteht darin, Simulationsstudien durchzuführen. Der Lernende bildet zunächst die Lernfabrik im Simulationsinstrument nach. Voraussetzung hierfür ist, dass der Lernende zunächst durch den Betrieb der Lernfabrik die Problemstellung wie einen zu geringen Durchsatz erkennt, gemeinsam mit einem Lehrenden die Ziele spezifiziert, die Daten beschafft, z.B. durch Zählen oder Messen, und diese aufbereitet. Die Formalisierung und Implementierung eines ersten sogenannten Ist-Modells schließen sich an [Küh-06] [VDI-3633a, S.18ff.]. Der Lernende gewinnt während des Aufbaus des Modells Erfahrungen in der Abstraktion. Der Lernende muss entscheiden, auf welche unwichtigen Details er beim Modellaufbau verzichtet und welche realen Gegebenheiten er vereinfacht [VDI-3633b, S.3].

Im nächsten Schritt sind Engpässe durch den Lernenden am digitalen Modell zu identifizieren [Mül-17b]. Sensitivitätsanalysen sind durchzuführen und Verbesserungen zu ermitteln. Bei einer Sensitivitätsanalyse beobachtet der Lernende das Systemverhalten bei Änderung einer einzelnen Einflussgröße [VDI-3633b, S.16]. Er beobachtet und reflektiert das Systemverhalten im Hinblick auf das Untersuchungsziel. Durch Sensitivitätsanalysen kann der Lernende auf Basis der vereinfachten digitalen Nachbildung der Lernfabrik ein Systemverständnis entwickeln. Es lassen sich Methoden des Fließprinzips und Pull-Prinzips wie der One Piece Flow und Kanban untersuchen. Der Lernende führt Experimente mit dem Simulationsmodell durch, bis Simulationsergebnisdaten vorliegen, die das vorab festgelegte Ziel der Untersuchung erfüllen. Die Maßnahmen, die den Experimenten zugrunde liegen, sind nun mit Blick auf die Umsetzbarkeit in der Lernfabrik zu interpretieren und analysieren. Da die Ergebnisse und Maßnahmen mit dem Simulationsmodell ermittelt und festgelegt wurden, das lediglich eine abstrahierte Darstellung der Lernfabrik ist, muss der Nutzer diese auf Anwendbarkeit in der Realität überprüfen. Nachdem die sogenannte



Legende:

-  – Simulationszyklus nach KÜHN, vgl. Kapitel 2.2.4 [Küh-06]
-  – Erfahrungsbasierter Lernzyklus nach KOLB, vgl. Kapitel 2.4.3 [Kol-71]
-  – Beziehung zwischen beiden Zyklen

Abbildung 4.5: Vorgehen zum EEDS nach [Mül-17b]

Verbesserung spezifiziert wurde, kann diese in die Lernfabrik übertragen werden. Durch das aktive Experimentieren im Sinne des Betriebs der verbesserten Lernfabrik erhält der Lernende eine Rückmeldung bezüglich der Wirksamkeit der durch ihn ermittelten Maßnahmen.

Der zentrale Vorteil, der sich aus der Anwendung des Vorgehens zum EEDS ergibt, besteht darin, dass Lernende eine Simulationsstudie mit all den zugehörigen praktischen Herausforderungen erleben können. Daten müssen bestimmt und gesammelt werden. Lernende müssen die Fabrik dazu in begrenzter Zeit selbst betreiben und bspw. Durchlaufzeiten von Produkten mit Stoppuhren erfassen. Diese Daten müssen entsprechend aufbereitet werden, damit sie durch die Simulation nutzbar sind. Beispielsweise müssen durch die Lernenden Datenformate, Einheiten oder Datentypen formatiert und bestimmt werden. Die Lernenden können die individuellen Verbesserungen, die sie am digitalen Modell experimentell ermittelten, in die Lernfabrik übertragen. Lernende erleben folglich, ob deren Verbesserungen den gewünschten Erfolg in der Realität zur Folge haben. Spätestens wenn die Lernenden die Lernfabrik nach der Umsetzung der ermittelten Verbesserungen betreiben, erfahren sie, ob die durch die Lernenden getroffenen Annahmen bei der Abstraktion angemessen waren und ob sie diese bei der Ergebnisanalyse angemessen berücksichtigt haben.

Das Vorgehen ist in einer Lernfabrik im Sinne eines Alternate Reality Games anzuwenden. Die Aufgabenstellung soll unter Zuhilfenahme der Lernfabrik die Grenze zwischen Realität und einem Spiel aufweichen. Lernende sind in freiwillig gebildete Kleingruppen aufzuteilen. Innerhalb der Gruppe wird ko-

operiert und zwischen Gruppen herrscht Konkurrenz. Dies wird durch KPIs erreicht, die die Leistung der Gruppe transparent für die anderen Gruppen darstellen. Beispiele können der Durchsatz an Fertigerzeugnissen oder die mittlere Auslastung von Werkern sein. Eine Begrenzung der Zeit ist geboten. Auf den Einsatz weiterer Spielelemente wie Punkte oder Level kann verzichtet werden, da Alternate Reality Games in der Lernfabrik und nicht in einer virtuellen Umgebung stattfinden [McG-11, S.120].

4.3 Verlauf des Spielens und Lernens

Das Zusammenspiel von App mit den Bestandteilen IKS und VBPIs und dem Vorgehen zum EEDS ist als Spiel- und Lernverlauf in Tabelle 4.2 strukturiert. Dieser Ablauf stellt die Lernumgebung zur Einarbeitung in ereignisdiskrete Simulation im Anwendungsbereich der Materialflusssysteme dar. Die Lernumgebung ist gemäß der Phasen *Onboarding*, *Midgame* und *Endgame* strukturiert [Man-19]. Lernende sollen nach Beendigung des Spiels dazu befähigt sein, Simulationsmodelle in einem Simulationsinstrument aufzubauen und Simulationsstudien durchzuführen. Sie sollen eigenständig weitere Bausteine und Objekte im Simulationsinstrument nutzen können.

Das Ziel des Onboardings besteht darin, den Nutzern einen leichten Einstieg in die Lernumgebung zu ermöglichen. Hierzu gehört auch, die initiale Aufmerksamkeit des Nutzers für die Lernumgebung zu gewinnen. Sie werden darüber hinaus in die Bedienung der App eingeführt. Dies umfasst beispielsweise Gesten wie das Wischen. Die wichtigsten Bausteine und Objekte des Simulationsinstruments werden in Form von Videos kurz vorgestellt.

Der wesentliche Teil des Spielens und Lernens findet im Midgame statt. Um weiterhin Punkte sammeln und Abzeichen freischalten zu können, müssen Nutzer individuelle Abläufe durch eine Programmiersprache implementieren. Hierzu nutzen sie die VBPIs. Zunächst replizieren die Lernenden den in der Aufgabenbeschreibung gezeigten Code, später erstellen sie ihn basierend auf Pseudocode und schließlich auf einer textuellen Beschreibung. Vor dem Hintergrund einer weiteren Intensivierung des Trainings und um die Gruppeninteraktion zu stärken, kann der Lernende im nächsten Schritt Aufgabenstellungen für andere Lernende konzipieren. Die Animation der von Nutzern aufgebauten Modelle im Sinne einer stereoskopischen/immersiven Projektion dient in dieser Spiel-Phase zur Motivationssteigerung und Validierung/Verifikation.

In der Phase des Early Endgames nutzt der Lernende in der App oder dem Simulationsinstrument anwenderdefinierte Bausteine. Solche Bausteine sind in der Praxis oft in Bibliotheken zusammengefasst und beziehen sich häufig auf einen konkreten Anwendungsbereich. Beispiele hierfür sind der VDA Bausteinkasten oder Bausteine zur Simulation von Multiagentensystemen [May-10]. Im Endgame werden die Lernenden praxisnah in die Durchführung von Simulationsstudien im Sinne des Simulationszyklus eingearbeitet. Das Vorgehen zum EEDS ist in einer Lernfabrik anzuwenden. Lernende, die Zugang zu einer Lernfabrik haben, können in dieser die Durchführung von Simulationsstudien trainieren. Lernende ohne Zugang zu solch einer Fabrik erlernen dies mithilfe eines miniaturisierten Modells einer Fabrik.

Tabelle 4.2: Lernumgebung zur Einarbeitung in Simulation: Überblick über den Spiel- und Lernverlauf

		Software	Elemente				Inhalt	(Qualifikations-)Ziel
		Mobile App auf Smartphone Simulationsinstrument auf PC	Tutorial/Video	IKS	VBPIs	Vorg. zum EEDS und Lernfabr.		
Spiel-Ebene	Early Onboarding	● ○	● ○ ○ ○				Der Lernende schaut zunächst kurze Video-Sequenzen.	Nutzer können die App bedienen.
	Late Onboarding	● ○	○ ● ● ○				Levelpack 1	Lernende sind mit wichtigen Bausteinen und Objekten vertraut. Bsp.: Quelle, Senke und Förderband.
	Early Midgame	● ○	○ ● ● ○				Levelpacks 2 und 3	Nutzer kennen die Syntax einer Programmiersprache zur Implementierung von Simulationsmodellen.
	Late Midgame	● ○	○ ● ● ○				Levelpack 4	Nutzer können mit anwenderdefinierten Bausteinen wie zur Multiagentensimulation Modelle aufbauen.
	Early Endgame	● ●	○ ● ● ●				Levelpacks 5 und 6	Nutzer haben Routine im Modellaufbau und der Programmierung entwickelt. Sie können Sensitivitätsanalysen durchführen.
	Endgame	● ●	○ ○ ● ●				Lernende ermitteln Engpässe in einer Lernfabrik anhand eines digitalen Modells. Sie erweitern diese simulativ und in der Lernfabrik.	Lernende kennen die Schritte zur Durchführung von Simulationsstudien und die Bedeutung der Abstraktion, Verifikation und Validierung.

Legende:

- – Kommt zur Anwendung
- – Kann zur Anwendung kommen
- – Kommt nicht zur Anwendung

5 Prototypische Implementierung und Erprobung

Das Kapitel zur prototypischen Implementierung und Erprobung ist in zwei Abschnitte gegliedert. Zunächst ist die Erstellung und Erprobung der mobilen App Production Simulation Application (PSIMA) beschrieben. PSIMA dient als Benutzungsschnittstelle für Lernende. PSIMA integriert die IKS und VBPIs. Im Rahmen materialflusstechnischer Anwendungen folgt die Erprobung des Vorgehens zum EEDS in einer Lernfabrik mit Studierenden. Der Beitrag der Entwicklungen zu einer hohen Qualität der Planung von Materialflusssystemen ist an zwei Anwendungsbeispielen im Gasturbinenbau und in der Motorenmontage beschrieben.

5.1 Gamifizierte App PSIMA

Zunächst sind im folgenden Abschnitt die Software-Werkzeuge vorgestellt, die zur Erstellung und für den Betrieb der Cloud-IT-Architektur erforderlich sind. Hieran schließt sich eine Erläuterung des Aufbaus und der Funktionalitäten von PSIMA an. Daraufgehend sind konkrete Beispiele für IKS benannt. Hierzu zählen ausgewählte vorkonfigurierte Bausteine und Objekte sowie anwenderdefinierte Bausteine zur Abbildung von Multiagentensystemen. In den weiteren Unterkapiteln ist sowohl die (teil-)automatische Modellgenerierung im Simulationsinstrument als auch die VBPIs am Beispiel des ereignisdiskreten Simulationsinstruments Tecnomatix Plant Simulation prototypisch umgesetzt und diskutiert. Bevor die Erprobung, Chancen und Entwicklungsperspektiven von PSIMA mit Studierenden beschrieben sind, wird ein Ansatz zur Animation in der App aufgezeigt.

Es wurden 37 Level exemplarisch konzipiert und im Simulationsinstrument getestet. In der App PSIMA sind 29 Level in vier Levelpacks umgesetzt worden. Die App ist in englischer Sprache implementiert worden. An den folgenden Textstellen mit direktem Bezug zur Umsetzung der App PSIMA finden sich englische Begriffe wieder.

5.1.1 Verwendete Software-Werkzeuge und Cloud-IT-Architektur

Das Konzept ist unter Zuhilfenahme existierender Software-Werkzeuge umgesetzt worden. Zu besseren Nachvollziehbarkeit weiterer Ausführungen werden diese Werkzeuge kurz vorgestellt. Hierzu zählen die open-source Datenbanken *MySQL* und *SQLite*, der quelloffene Webserver *Apache HTTP Server*, die Game-Engine *Unity3D* sowie das Simulationsinstrument *Tecnomatix Plant Simulation*.

MySQL ist eine frei verfügbare und gemäß den Angaben auf der Website des Herstellers das weltweit beliebteste open-source Datenbankmanagementsystem. Es wird von einer großen und aktiven Community von Entwicklern unterstützt [MyS-19b]. Es handelt sich um ein weitverbreitetes Datenbankmanagementsystem zur Erstellung und Wartung relationaler Datenbanken. Es baut auf der SQL auf. Hierbei handelt es sich um eine standardisierte Sprache zur Manipulation von Daten in Datenbanken [MyS-19a].

SQLite ist ebenso eine frei verfügbare relationale Datenbank. Im Vergleich zu den meisten anderen SQL Datenbanken werden Daten ohne separate Server-Prozesse direkt vom Datenträger gelesen oder dorthin geschrieben [SQL-19]. Die plattformunabhängig einsetzbare Datenbank erfordert wenig Speicherplatz, Arbeitsspeicher und Prozessorleistung. Aus diesen Gründen kommt sie oft auf Smartphones oder MP3-Playern zum Einsatz.

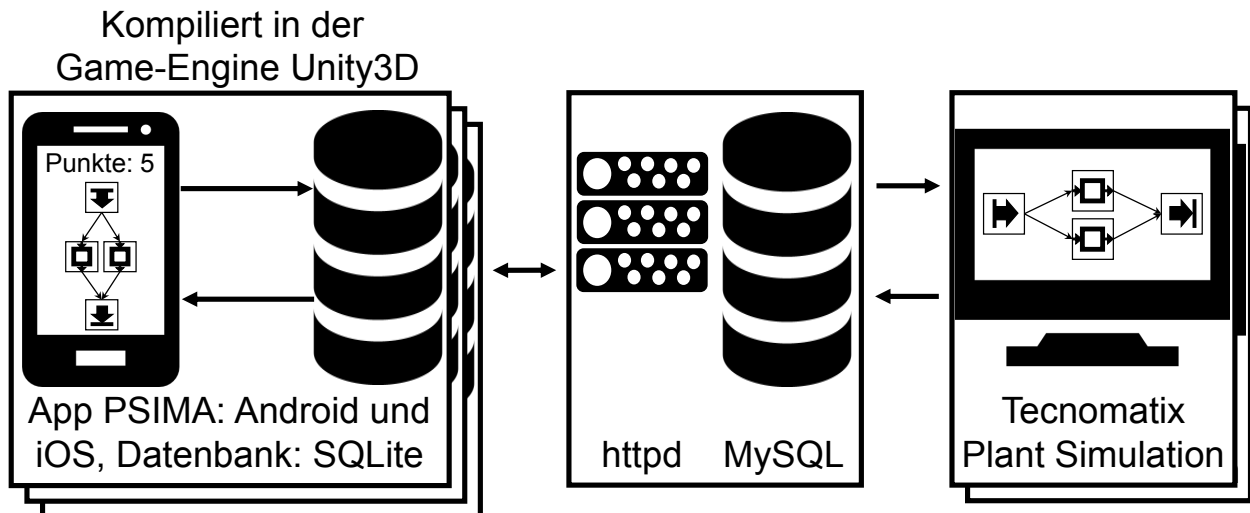
Der Webserver Apache HTTP (=Hypertext Transfer Protocol) Server, oder kurz httpd, ist einer der beliebtesten open-source Webserver [Apa-19]. Webserver ermöglichen den Austausch von Daten im Web. Wenn z.B. eine mobile App Daten aus einer Datenbank benötigt, ruft die App eine gewisse Website auf und liest die relevanten Daten aus, die so mittels Webserver bereitgestellt wurden. Um Daten auf den Server zu laden wird von der App automatisch eine Website aufgerufen, die es ermöglicht, Daten auf den Webserver und somit in eine Datenbank zu übertragen. Sogenannte php (=Hypertext Preprocessor)-Skripte, die auf dem Webserver abgelegt sind, werden bei diesem Prozess ausgeführt und bewirken, dass die angefragten Daten bereitgestellt werden.

Die Game-Engine Unity3D dient der Erstellung von zwei- und dreidimensionalen digitalen Spielen. Unity3D eignet sich insbesondere zur Umsetzung des Konzepts aus Kapitel 4, da alle gängigen Plattformen durch die Software unterstützt werden, die sogenannte Multiplattform-Fähigkeit [Uni-19]. Das bedeutet, dass Unity3D Anwendungen erstellt, die auf Betriebssystemen wie iOS, Android oder Windows ausgeführt werden können. Weiterhin ermöglicht die Software die Erstellung von AR- und Virtual Reality (VR)-Anwendungen.

Folgende Simulationsinstrumente erfüllen die technischen Voraussetzungen, um als Teil der Cloud-IT-Architektur nutzbar zu sein: AnyLogic, Arena, Flexsim, Tecnomatix Plant Simulation und Witness [Mou-14]. Die Wahl fällt auf das Simulationsinstrument *Tecnomatix Plant Simulation* der Firma Siemens. Das ereignisdiskrete Simulationsinstrument ist im deutschsprachigen Raum weit verbreitet. ELEY (2012) beschreibt, dass diese Software bereits mehrere Tausend Male installiert wurde [Ele-12, S.VIII]. Der VDA benennt das Simulationsinstrument als Standard zur Simulation von Materialflüssen bei den Mitgliedern AUDI, BMW, Daimler, Opel und VW [VDA-09, S.10]. Es dient gemäß den Angaben des Herstellers der Simulation, Visualisierung, Analyse und Verbesserung von Produktionssystemen und Logistik-Prozessen [Sie-14]. Die Software ist objektorientiert aufgebaut und unterstützt wichtige Schnittstellenstandards wie ODBC (=Open Database Connectivity). Dabei handelt es sich um eine allgemeingültige SQL-Schnittstelle, die einen Datenaustausch mit unterschiedlichen Datenquellen und -senken ermöglicht [Sie-19]. Hierdurch lassen sich automatisiert Daten mit weiteren Software-Anwendungen austauschen. Tecnomatix Plant Simulation lässt die Erweiterung der Modellwelt zu. Beispiele für vorkonfigurierte Bausteine sind Einzelstationen oder Förderanlagen für den Transport von Materialien. Bei Einzelstationen handelt es sich um Maschinen, die Zeit verbrauchen. Sie haben eine Kapazität von ei-

nem sogenannten beweglichen Element¹⁰. Bei der Software Tecnomatix Plant Simulation handelt es sich um ein kommerzielles Simulationsinstrument. Die an C orientierte Programmiersprache SimTalk wird vom Hersteller in der Software mitgeliefert [Lan-10]. Dies ermöglicht es Anwendern, individuelle Abläufe durch eine Programmiersprache zu implementieren. Mit Tecnomatix Plant Simulation lassen sich deterministische als auch stochastische Modelle erstellen.

In Abbildung 5.1 sind die konkreten Namen der exemplarisch gewählten Softwarepakete im Kontext ihres Zusammenspiels als Cloud-IT-Architektur benannt. Abbildung 5.1 ist eine vertiefende Darstellung der in Abbildung 4.2 gezeigten generischen Cloud-IT-Architektur.



Legende:

Unity3D	– Game-Engine
Android/iOS	– Betriebssysteme
SQLite	– Open-source Datenbank
MySQL	– Open-source Datenbank
httpd	– Apache HTTP Server
Tecnomatix Plant Simulation	– Simulationsinstrument
	– Bausteine/Objekte, vgl. Tabelle C.2 im Anhang C
	– für bereits eingeführte Grafiken siehe Legende der Abbildung 4.2

Abbildung 5.1: Cloud-IT-Architektur der prototypischen Implementierung: Abbildung 4.2 ergänzt um die gewählten Softwarepakete der prototypischen Implementierung

MySQL und SQLite übernehmen folgende Funktionen:

- Sie speichern Modelldaten und Attribute. Hierzu zählt der strukturelle Aufbau eines spezifischen Simulationsmodells. Ein konkretes Beispiel hierfür ist der Typ eines instanziierten Bausteins, z.B. Einzelstation, und dessen Position in x- und y- Richtung in der grafisch-interaktiven Modellierungsumgebung. Weiterhin wird der Spielstand des Benutzers im Sinne der bereits abgeschlossenen

¹⁰Die Bezeichnung *bewegliches Element* ist spezifisch für das Simulationsinstrument Tecnomatix Plant Simulation. Gemeint ist damit ein temporäres physisches Objekt.

Level darin dokumentiert. Auch die Ergebnisse einzelner Simulationsläufe, die durch den Benutzer angestoßen wurden, sind in der Datenbank verfügbar. Schließlich sind die Benutzerprofile der Teilnehmer zu benennen.

- In der App PSIMA lassen sich Daten manipulieren:
 - Lesen/Schreiben der Attribute von Objekten und Bausteinen eines bestimmten Modells.
 - Lesen/Schreiben spezifischer Projekt-Informationen.
 - Lesen der Simulationsergebnisdaten eines Simulationslaufs. Die Simulationsergebnisdaten entstehen im Simulationsinstrument.
 - Lesen des Benutzerfortschritts aller Benutzer sowie Aktualisierung des eigenen Fortschritts.
- Das Simulationsinstrument kann die folgenden Daten manipulieren:
 - Lesen aller Attribute von Objekten und Bausteinen eines bestimmten Modells.
 - Lesen spezifischer Projekt-Informationen.
 - Schreiben von Simulationsergebnisdaten.

Um die benannten Funktionen zu ermöglichen, ist ein Datenbankmodell errichtet worden. Dieses ist im Anhang B, in Abbildung B.1, visualisiert. Die Cloud-IT-Architektur beinhaltet die zwei fast identischen Datenbanken MySQL auf einem Server und SQLite auf dem mobilen Endgerät als Bestandteil von PSIMA. Die SQLite Datenbank umfasst zusätzlich eine Tabelle „Config“, die Konfigurations- und Nutzerdaten enthält. Beispiele für solche Daten sind die IP (=Internet Protokoll)-Adresse der Datenbank und das persönliche Passwort. Die Funktion der Datenbank SQLite besteht darin, die für den entsprechenden Nutzer relevanten Daten offline manipulieren zu können. Die SQLite Datenbank beinhaltet jedoch ausschließlich Daten, für die der jeweilige Benutzer Lese- oder Schreibrechte hat. Modelle können somit in PSIMA durch die Datenbank auf dem Endgerät auch ohne bestehende Internetverbindung erstellt und manipuliert werden. Geht man bspw. von einer Benutzung in öffentlichen Verkehrsmitteln aus, kann kaum eine dauerhaft stabile Internetverbindung vorausgesetzt werden. Die SQLite Datenbank ermöglicht somit den Betrieb der App PSIMA auch ohne bestehende Internetverbindung. Lediglich Simulationsläufe können ohne Verbindung zum Internet nicht gestartet werden. In vom Anwender zu definierenden Synchronisierungsintervallen werden die SQLite und MySQL Datenbanken unter Zuhilfenahme des Webservers abgeglichen.

5.1.2 Aufbau und Funktionen

In diesem Kapitel sind die Bestandteile der Implementierung kurz beschrieben. Diese umfassen die Funktionalitäten der App PSIMA sowie die (teil-)automatische Modellgenerierung im Simulationsinstrument Tecnomatix Plant Simulation. Ergänzende Ausführungen finden sich im Anhang C. Die App erfüllt folgende Funktionen:

- Speichert die durch den Benutzer implementierten Simulationsmodelle sowie Simulationsergebnisdaten.
- Übermittelt Modelldaten zum Simulationsinstrument.
- Empfängt Simulationsergebnisdaten vom Simulationsinstrument.

- Unterstützt das Empfangen und Teilen von Abzeichen zwischen unterschiedlichen Anwendern. Ermöglicht den Empfang und die Vergabe von Punkten.
- Ermöglicht es dem Lernenden, individuelle Simulationsmodelle zu erstellen.
- Ermöglicht es dem Lernenden, Abläufe durch VBPIs zu formalisieren.

Die Menüführung der App ist in Abbildung C.1 im Anhang C gezeigt. In den Abbildungen 5.2 und 5.3 sind Bildschirmfotos der App abgebildet. Die Funktionen sind im Folgenden kurz erläutert [Mül-17a].

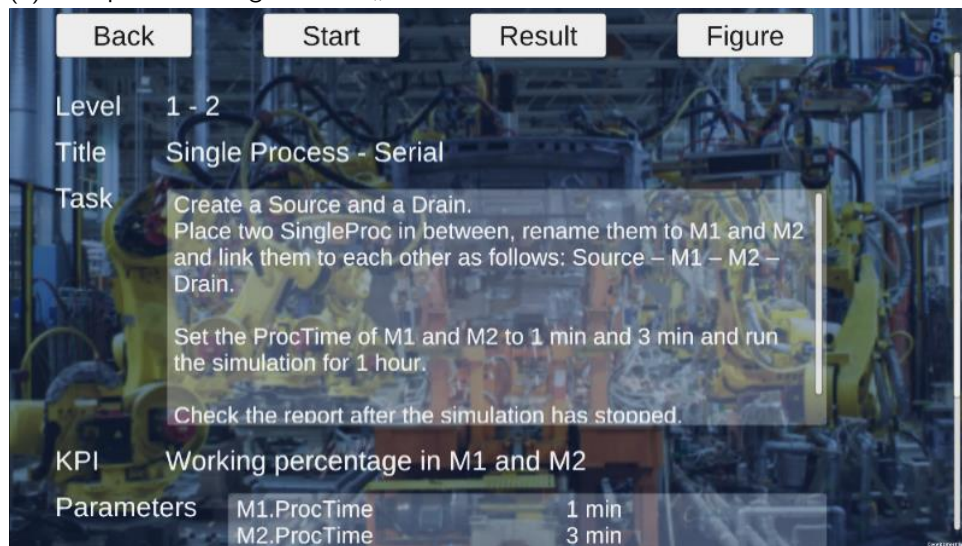
Nach dem Öffnen der App wählt der Lernende, ob er im Expertenmodus, sog. „Pro Mode“ oder im Levelmodus, sog. „Game Mode“, spielt. Im Expertenmodus kann er ein neues Projekt anlegen und wird dann sofort zur Modellierungsumgebung, vgl. Abbildung 5.2c, weitergeleitet. Der Expertenmodus eignet sich zur Abbildung der Lernfabrik im Sinne des Vorgehens zum EEDS oder für den freien Modell-aufbau nach individuellen Erfordernissen des Anwenders. Abbildung 5.2a zeigt Levelpack 1 im *Game Mode*. Das Levelpack besteht aus zehn Leveln und einem Bonus-Level. Bei einem Bonus-Level handelt es sich um ein Level, das dem Lernenden freiwillig und zusätzlich zur Bearbeitung angeboten wird. Der Abschluss eines Bonus-Levels wird durch zusätzliche Punkte und Abzeichen belohnt. Der Nutzer erhält für jedes erfolgreich abgeschlossene Level einen gelben Stern. Diese Sterne sollen dem Lernenden seinen Erfolg und Fortschritt bewusst machen. Abbildung 5.2b zeigt beispielhaft die Beschreibung von Level 1-2. Das Ziel von Level 1-2 besteht darin, dass der Lernende mehrere Einzelstationen sequenziell miteinander verbindet und weiß, dass Objekte und Bausteine umbenannt werden können. Abbildung 5.2c zeigt für diesen Fall die Modellierungsumgebung, in welcher der Lernende durch Ziehen und Ablegen die geforderten Bausteine und Objekte bereits instanziierte. In dieser Modellierungsumgebung bauen Lernende Materialflusssysteme auf. Die in der App PSIMA zu diesem Zweck implementierten Bausteine und Objekte sind kurz in Tabelle C.1 im Anhang C erläutert. In der Tabelle C.2 sind darüber hinaus die Symbole der Objekte/Bausteine ergänzt.

In Abbildung 5.3a ist am Beispiel der Quelle die Liste der zu manipulierenden Attribute ersichtlich. Die Werte der Attribute können vom Lernenden nach Belieben angepasst werden. Der Lernende soll zu häufiger Interaktion motiviert werden. Um dies zu erreichen, wird er mit virtuellen Abzeichen belohnt. Diese sind in der sogenannten „Hall of Fame“ für den Lernenden sichtbar. Die „Hall of Fame“ ist ein Bereich in der App, wo die Erfolge der Spieler für alle Mitspieler ersichtlich dargestellt sind. Abbildung 5.3b zeigt im Hintergrund die vom Lernenden erworbenen Abzeichen und im Vordergrund Details zum Angetippten. Im gezeigten Fall hat der Lernende das Abzeichen „Persistent“ durch zehnmaliges Öffnen der App erhalten. Weiterhin ist dort ersichtlich, dass *User_8* dieses Abzeichen bereits erhalten hat. Abbildung 5.3c zeigt das „Leaderboard“. Hierbei handelt es sich um ein global verfügbares Ranking, in dem die Lernenden gemäß ihren Punkten absteigend sortiert gelistet sind.

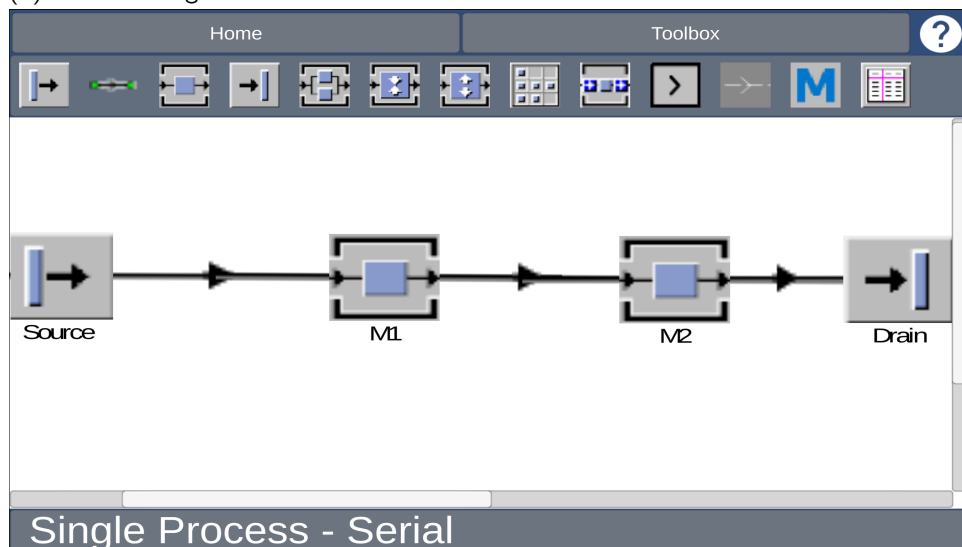
Von hoher Relevanz für diese Arbeit ist die Modellierungsumgebung, gezeigt in Abbildung 5.2c. In dem unteren Balken findet sich der aktuelle Projektname wieder: *Single Process - Serial*. Die Modellierungsumgebung von PSIMA ermöglicht den mobilen grafisch-interaktiven Aufbau von Simulationsmodellen. Lernende können durch Ziehen und Ablegen die Bausteine und Objekte, die in der Toolbox verfügbar sind, in der Modellierungsumgebung instanziiieren. Weiterhin lassen sich bereits in der Modellierungsumgebung instanziierte Objekte an einen anderen Ort ziehen und ablegen.



(a) Levelpack 1 im sogenannten „Game Mode“

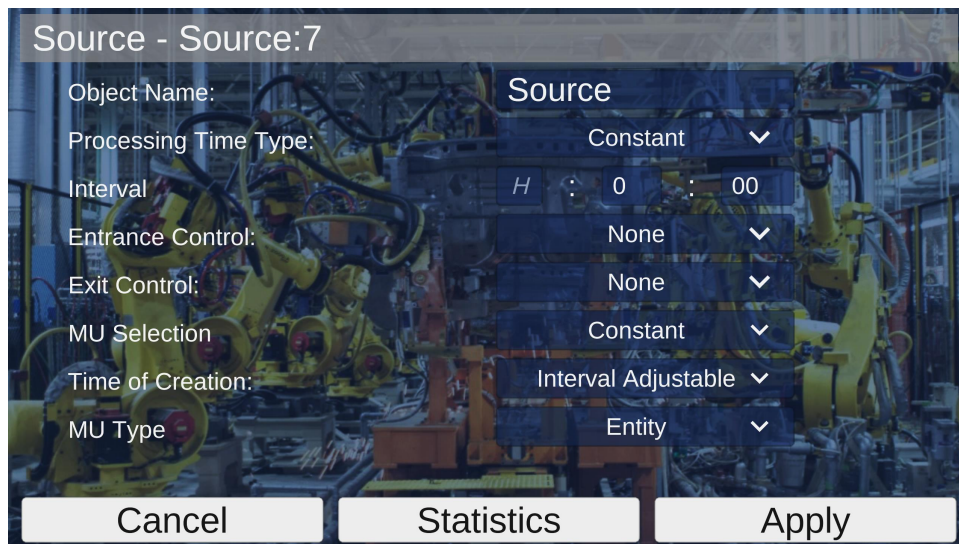


(b) Beschreibung von Level 1-2

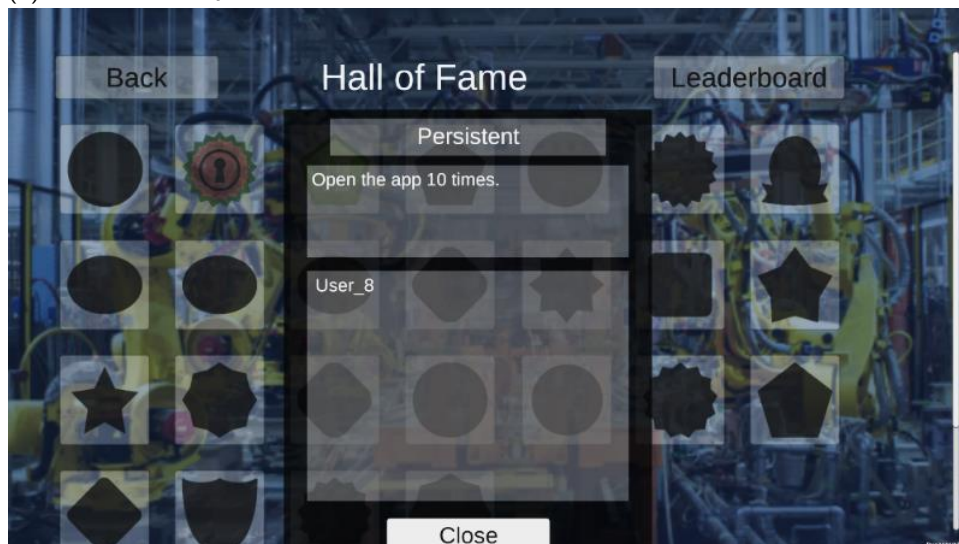


(c) Modellierungsumgebung, vgl. Tabelle C.2 im Anhang C

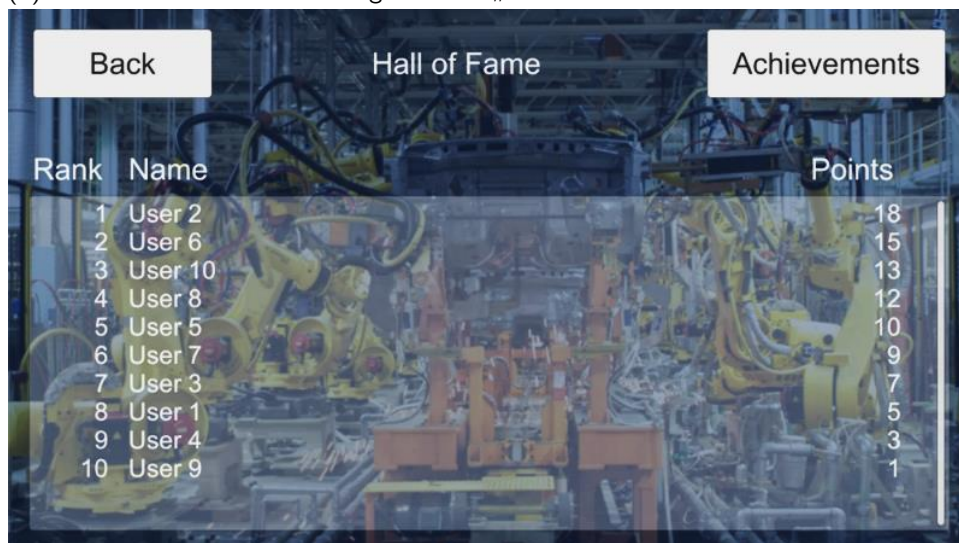
Abbildung 5.2: Bildschirmfotos der prototypischen App Teil 1 nach [Mül-17a]



(a) Attribute der Quelle



(b) Virtuelle Abzeichen in der sogenannten „Hall of Fame“



(c) Global verfügbares Ranking: Sogenanntes „Leaderboard“

Abbildung 5.3: Bildschirmfotos der prototypischen App Teil 2 nach [Mül-17a]

In Abbildung C.2 im Anhang C sind zwei beispielhafte Bildschirmfotos hinsichtlich der Darstellung der Simulationsergebnisdaten abgedruckt. Weiterhin finden sich in Abbildung C.2 zwei Bildschirmfotos von Tutorials in PSIMA über Bausteine und Objekte sowie über die VBPIs.

5.1.3 Inkrementell komplizierter werdende Simulationsmodelle

Die IKS sind in sechs Levelpacks unterteilt worden. In den folgenden Tabellen 5.1, 5.2, 5.3 und 5.4 sind exemplarische Level abgedruckt. Diese Aufgabenstellungen werden inklusive der Abbildungen dem Lernenden zur Verfügung gestellt, nachdem dieser das jeweilige Level-Icon, vgl. Abbildung 5.2a, angetippt hat. Das sechste Levelpack dient dazu, dass Nutzer selbst Aufgabenstellungen verfassen und in der Community veröffentlichen können. Im Folgenden sind die gezeigten Beispiele kurz beschrieben. Hinsichtlich der Symbole von Bausteinen/Objekten und deren Bedeutung sei auf die Tabellen C.1 und C.2 im Anhang C verwiesen. Die Nomenklatur von Levelpacks und Leveln ist im Folgenden 'Level x-y', wobei x = Levelpack und y = Level des jeweiligen Levelpacks.

Das Ziel von Level 1-1 (Tabelle 5.1) besteht in der erstmaligen Reproduktion des in der Aufgabenstellung beschriebenen Materialflusses in der Modellierungsumgebung von PSIMA. Bei der erstmaligen Interaktion mit PSIMA wird dies dem Lernenden auch in einem kurzen Video erläutert. Durch Bearbeitung des Levels 2-5 (Tabelle 5.2) erhöht der Lernende seine Kompetenz in der Programmierung. Die Programmierung ist zur flexiblen Abbildung von Abläufen in Materialflusssystemen erforderlich. Der Lernende programmiert hierzu sogenannte Methoden in der App PSIMA unter Nutzung der VBPIs. Im Level 4-4 (Tabelle 5.3) nutzt der Lernende anwenderdefinierte Bausteine zur Abbildung eines Multiagentensystems am Beispiel eines FTS zur Layoutplanung. Das Level 5-1 (Tabelle 5.4) führt in den Story-Modus ein. Die Maßnahmen zur Verbesserung des Materialflusses werden durch eine Geschichte eingegrenzt. Die ausführliche Beschreibung aller gestalteten Levelpacks ist online unter www.lernzeuge.de abrufbar [Sch-19]. In den Tabellen D.1, D.2, D.3 und D.4 im Anhang D ist eine Kurzbeschreibung der entwickelten Level an diese Arbeit angefügt. Eine Liste der Abzeichen, deren Grafiken, die Kriterien für deren Verleihung und der Stand der Umsetzung sind im Anhang E, in Tabelle E.1, abgedruckt [Sch-19].

Tabelle 5.1: Beschreibung von Level 1-1 nach [Sch-19]

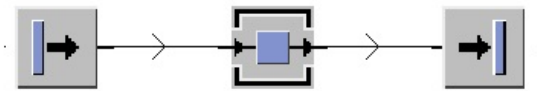
Aufgabenstellung	Abbildung
Create a Source and a Drain. Place a SingleProc in between and link them to each other as follows: Source – SingleProc – Drain.	
Run the simulation for one hour. Check the throughput in the drain.	

Tabelle 5.2: Beschreibung von Level 2-5 nach [Sch-19]

Aufgabenstellung	Abbildung
Create three Sources "S1" to "S3", three SingleProcs "M1" to "M3", a Buffer and a Drain. Add a Method called "Ctrl" and attach it to the exit of every SingleProc. Only the Sources are connected to their machine. The Buffer is connected to the Drain. In the method, write: <pre>waituntil Buffer.empty prio 1; @.move(Buffer);</pre> Run the simulation for one day. Check the throughput in the SingleProcs and the statistics of the Buffer.	

Tabelle 5.3: Beschreibung von Level 4-4 nach [Sch-19]

Aufgabenstellung	Abbildung
You are designing in this task a multiagent simulation example considering the routing of automated guided vehicles. Therefore, you are using additionally to the object "Area" the "AGVLoading" and "AGVUnloading". Create in the frame a grid of 14 x 7 areas, closely located next to each other. Two sources and one drain are to be instantiated in the frame. Check the figure for knowing where to place the objects. The source on top left named "Source_Transporter" creates 15 transporters in an interval of 30 seconds. The initial destination of the transporter is the area called "AGVLoading". The "Source_Entity" is connected to the area "AGVLoading". The "Drain_Entity" is to be connected to the area "AGVUnloading". Simulate the model for 1 hour. Check the behavior of the automated guided vehicles and determine the throughput of transported entities.	

Tabelle 5.4: Beschreibung von Level 5-1 nach [Sch-19];

Aufgabenstellung	Abbildung
You are a Junior Simulation Engineer who is hired by an automotive supply manufacturing company to establish a small simulation division. The management wants to improve the throughput through a lean production line. Upon arrival, you take a close look at all departments in the material flow chain. While inspecting your first station, the delivery of the goods, you see a lot of opportunities for improvement at a first glance. The delivery intake consists of three terminals. One of them has the capability of handling big parts, one can handle small parts and the third terminal is suitable for both categories. Since the third terminal requires high investments for maintenance, the management is looking for ways to invest the money on a better spot to increase the throughput. Aware of your troubleshooting skills, you set off to work. Hint 1: The parameters of both the machines and the Movable Units are not subject to change. Hint 2: Detect the bottleneck and implement a workaround. Hint 3: You must edit some code in the methods. One side note: In level pack 3, your solution will be reviewed by a fellow simulation expert who has already solved the certain level. After you have successfully mastered a level, you are also eligible to review other solutions and gain achievements for your reviewing contributions. Back to business: Run the simulation for 1 day. Check the contents of the stores and compare it with the previous set-up.	

Der Lernende sieht in der Übersicht der Level seinen individuellen Bearbeitungsstand dargestellt, vgl. Abbildung 5.2a auf Seite 79. Vier Status lassen sich unterscheiden. Bild 5.4a wird dem Lernenden angezeigt, wenn dieser für das jeweilige Level in der App noch kein Simulationsmodell aufgebaut und an das Simulationsinstrument übergeben hat. Wurde mindestens eines übergeben, so werden die restlichen drei Bilder relevant. Abbildung 5.4b wird dem Lernenden angezeigt, wenn die Simulationsergebnisdaten nicht den erwarteten entsprechen. In den Levelpacks eins bis vier werden die Simulationsergebnisdaten automatisch in der App auf Korrektheit überprüft. Das Symbol wird auch angezeigt, wenn Modellfehler im Simulationsinstrument zum Abbruch eines Simulationslaufs führen oder gar den Aufbau eines Modells verhindern. Abbildung 5.4c zeigt dem Lernenden an, dass Ergebnisse zur Ansicht bereitstehen. Bei einem transparenten Stern ist die Prüfung der KPIs positiv verlaufen. Dies bedeutet, dass die vom Simulationsinstrument übermittelten Simulationsergebnisdaten den erwarteten Werten zur Lösung des jeweiligen Levels entsprechen. Nach Ansicht der Ergebnisse wird aus dem transparenten Stern ein gelber, wie in Abbildung 5.4d abgedruckt. Der gelbe Stern signalisiert dem Lernenden ein erfolgreich abgeschlossenes Level.

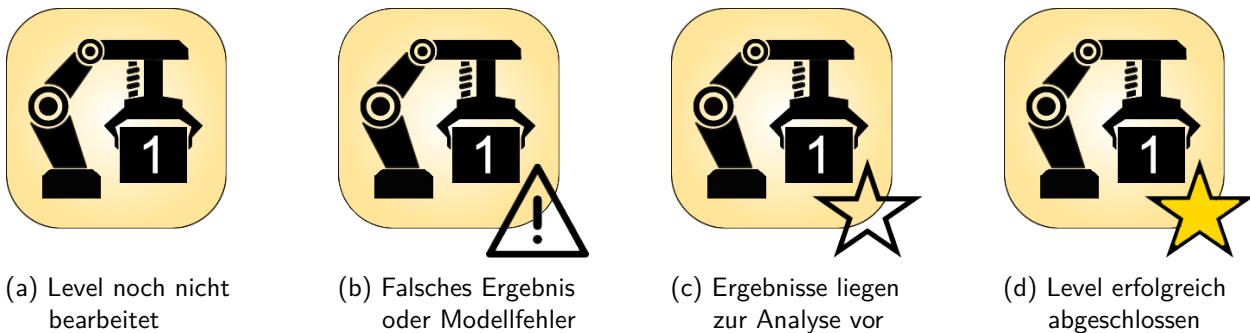


Abbildung 5.4: Status der Levelbearbeitung nach [Sch-19]

5.1.4 Multiagentensimulation

Die App PSIMA bietet die Möglichkeit zur Erweiterung der Modellwelt um anwenderdefinierte Bausteine und Objekte. Im Folgenden ist diese Erweiterungsfähigkeit am Beispiel einer anwenderdefinierten Bausteinbibliothek zur Multiagentensimulation beschrieben. Die Wegfindung und Layoutplanung stehen im Vordergrund. Das Ziel besteht darin, die Auslastung von Wegen ermitteln zu können. Es wird angenommen, dass autonome logistische Gegenstände des Materialflusses Entscheidungen hinsichtlich der Wegfindung treffen. Solch eine Multiagentensimulation ermöglicht es dem Planer von Materialflusssystemen Erkenntnisse zu generieren, die auf die Gestaltung des Materialflusses von Fabriken im Sinne des Layouts übertragbar sind. Vor diesem Hintergrund wurde der generische Baustein *Area* in dem Simulationsinstrument Tecnomatix Plant Simulation entwickelt. Abbildung 5.5 zeigt das Symbol sowie den strukturellen Aufbau einer *Area*.

Im Beispiel beschreibt eine *Area* einen Bereich mit der Kapazität von einem mobilen Materialflussbaustein wie einem Fahrzeug. Das Durchlaufen einer *Area* erfordert eine vom Anwender zu definierende Zeit, die von der physischen Größe einer *Area* abhängig ist. Jede *Area* ermöglicht dem mobilen Material-

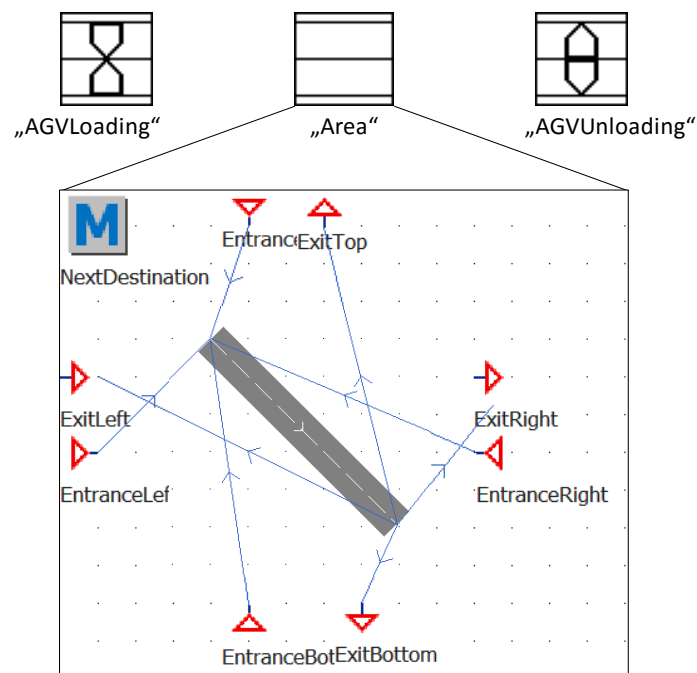
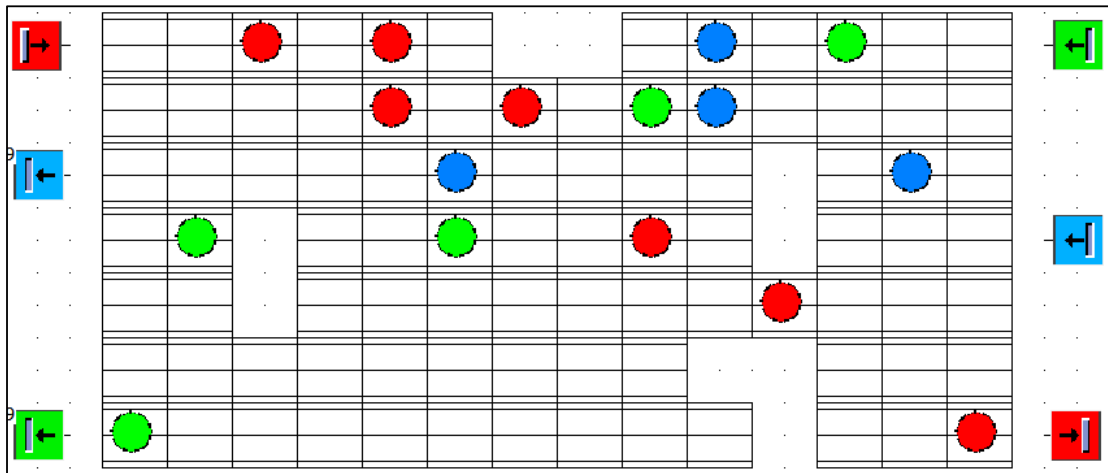


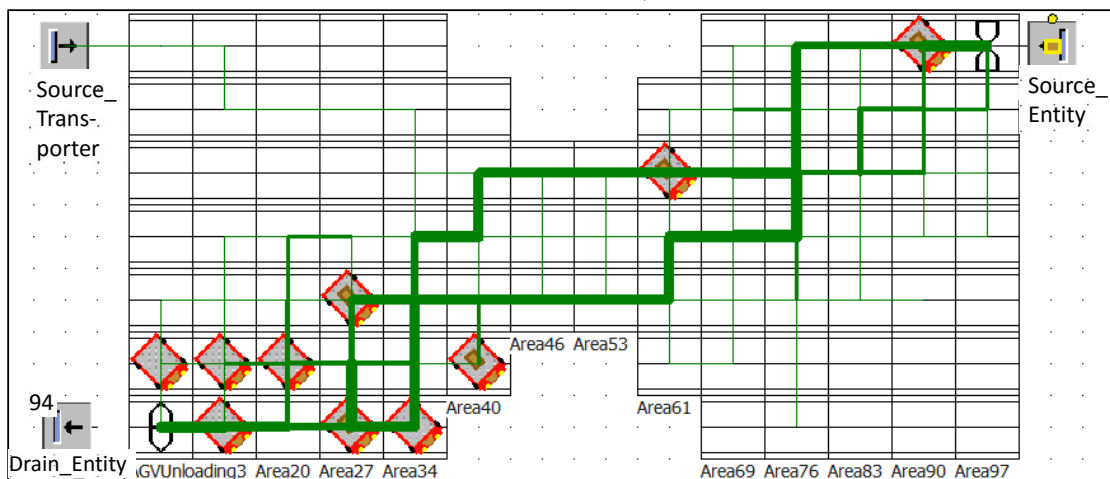
Abbildung 5.5: Hierarchie der Implementierung der Multiagentensimulation im Simulationsinstrument:
Oben: Symbol des anwenderdefinierten Bausteins Area; Unten: Aufbau bestehend aus Baustein Weg und mehreren Übergängen zu benachbarten Areas.

flussbaustein einen Übergang in benachbarte Areas. Zu einem Zeitpunkt kann sich maximal ein mobiler Materialflussbaustein in einer Area befinden. Areas können nach Belieben vom Benutzer miteinander verbunden werden. So lässt sich im Simulationsmodell ein Layout durch Areas beschreiben. Es liegt die Annahme zugrunde, dass ein mobiler Materialflussbaustein den schnellsten Weg zum Ziel von seiner aktuellen Position kennt. Eine Kommunikation ist aber nur mit angrenzenden Areas möglich. Im Falle, dass die nächste gewünschte Area bereits durch einen anderen mobilen Materialflussbaustein belegt ist, lagert dieser auf eine benachbarte freie Area um und sucht sich von dort aus den kürzesten Weg zum Ziel. Priorität hat im vorgestellten Beispiel der mobile Materialflussbaustein, der zuerst auf eine freie Area umlagert.

In Abbildung 5.6a ist dieses Prinzip an einem Minimalbeispiel mit drei Quellen und drei Senken verdeutlicht. Die Farben der mobilen Materialflussbausteine stehen in Beziehung zu den Farben der Quellen und Senken. Die Quellen und Senken beschreiben den Start und das Ziel. Die Materialflüsse kreuzen sich im gezeigten Beispiel. Abbildung 5.6b zeigt eine Erweiterung der Funktionalitäten um ein FTS. Eine vom Lernenden einzugebende Anzahl an Fahrerlosen Transportfahrzeugen (FTF) (=mobiler Materialflussbaustein) wird zu Simulationsbeginn von der gezeigten *Source_Transporter* generiert. Die FTF transportieren dann temporäre physische Objekte von der *Source_Entity* zur *Drain_Entity*. In der Abbildung zeigen die grün dargestellten Linien die Auslastung der Wege als Sankey-Diagramm. Die Funktionalität der Be- und Entladung wurde in den Bausteinen *AGVLoading* und *AGVUnloading* implementiert.



(a) Multiagentensimulation mit sich kreuzenden Materialflüssen. Die Abbildung zeigt drei Quellen und drei Senken. Die runden mobilen Materialflussbausteine bewegen sich je von den gleichfarbigen Quellen zu den Senken. Vgl. Symbole von Bausteinen/Objekten mit Tabelle C.2 im Anhang C.



(b) Die FTF transportieren wiederholt temporäre physische Objekte von der *Source_Entity* zur *Drain_Entity*. Das in grüner Farbe eingezeichnete Sankey-Diagramm lässt Rückschlüsse auf die Auslastung der *Areas* zu.

Abbildung 5.6: Beispiele für Multiagentensimulation unter Nutzung von anwenderdefinierten Bausteinen

Der Quellcode F.1 im Anhang F dokumentiert die Wegfindung. Der Quellcode wird im Simulationsinstrument Tecnomatix Plant Simulation ausgeführt, wenn ein mobiler Materialflussbaustein eine *Area* durchlaufen hat. Im Ergebnis wird durch den Quellcode ermittelt, in welche benachbarte *Area* umgelagert wird.

Abbildung 5.7 zeigt schließlich das Prinzip der Modellierung mit den entwickelten anwenderdefinierten Bausteinen in der App PSIMA am Beispiel des zuvor beschriebenen Levelpacks 4-4. In der Toolbox stehen die benannten Bausteine zur Modellierung zur Verfügung. Sie lassen sich durch Ziehen und Ablegen instanzieren. Die Instanzen lassen sich parametrisieren.

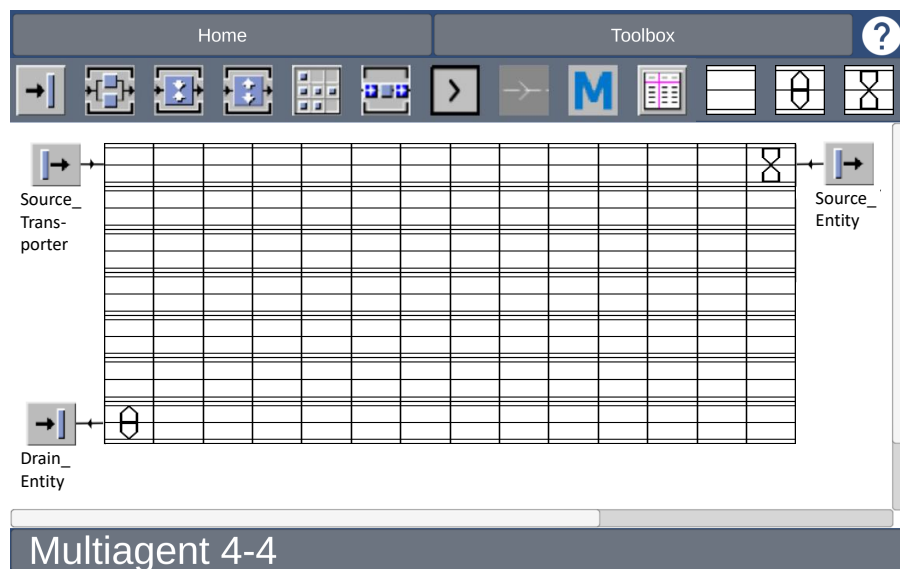


Abbildung 5.7: Prinzip der Modellierung von Agenten in PSIMA am Beispiel von Levelpack 4-4; vgl. Symbole von Bausteinen/Objekten mit Tabelle C.2 im Anhang C

5.1.5 (Teil-)automatische Modellgenerierung im Simulationsinstrument

Die (teil-)automatische Modellgenerierung wird im Simulationsinstrument ausgeführt. Diese ermöglicht, dass das durch den Lernenden in der vereinfachten GUI PSIMA erstellte Modell im Simulationsinstrument nachgebaut und simuliert wird. Anforderungen an das Simulationsinstrument sind:

- Das Simulationsinstrument Tecnomatix Plant Simulation soll automatisch - also ohne das Eingreifen eines Menschen - das vom App-Benutzer entwickelte Modell generieren, parametrisieren, simulieren und die Simulationsergebnisdaten in die Datenbank schreiben.
- Fehler, die während der (teil-)automatischen Modellgenerierung auftreten, müssen abgefangen werden. Beispiele können vom Benutzer der App programmierte Endlosschleifen sein. Auch Simulationsmodelle, die keine Stopp-Bedingung wie ein zeitlich definiertes Simulationsende aufweisen, dürfen die Stabilität des Software-Aufbaus nicht negativ beeinflussen.
- Skalierbarkeit.

Vor dem Hintergrund dieser Anforderungen sind zur (teil-)automatischen Modellgenerierung zwei sogenannte Instanzen des Simulationsinstruments Tecnomatix Plant Simulation erforderlich. Bei einer Instanz des Simulationsinstruments handelt es sich um eine Kopie des Simulationsinstruments im Hauptspeicher des Computers [Com-03]. Übertragen auf die Cloud-IT-Architektur wird auf dem Computer, auf dem das Simulationsinstrument betrieben werden soll, die Software Tecnomatix Plant Simulation in zwei Instanzen mit unterschiedlichen Simulationsmodellen geöffnet. Die Instanz des Simulationsinstruments mit dem nachgebildeten Simulationsmodell, das der Spieler in der App formalisierte, ist im Folgenden als „PSIMA_Executer“ bezeichnet. Dieses wird dabei von einer dauerhaft aktiven Instanz von Tecnomatix Plant Simulation namens „PSIMA_Controller“ gesteuert. Hierzu wird die Hypertext Markup Language (=HTML) Schnittstelle von Tecnomatix Plant Simulation genutzt [Sie-19]. Das Zusammen-

spiel ist in Form von zwei Programmablaufplänen in den Abbildungen B.2a und B.2b im Anhang B dargestellt. Die beschriebene Umsetzung ist kaum fehleranfällig und lässt den Betrieb weiterer Instanzen des Simulationsinstruments auf räumlich verteilten Rechnern zu. Durch diese skalierbare Architektur kann also eine kurze Zeit bis zur Bereitstellung von Simulationsergebnisdaten auch bei hohem Simulationsaufkommen gewährleistet werden. Weitere Details hinsichtlich der Umsetzung finden sich im Anhang B.

5.1.6 Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen

Die VBPIs ist am Beispiel der Sprache *SimTalk* des objektorientierten Simulationsinstruments Tecnomatrix Plant Simulation umgesetzt worden [Sie-19]. Dieser Arbeit liegt SimTalk 1.0 zugrunde. Der Hersteller veröffentlichte mit der Software-Version 12.1 die Programmiersprache SimTalk 2.0. Da die Implementierung der in dieser Arbeit beschriebenen VBPIs vor der Veröffentlichung von SimTalk 2.0 begann, bezieht sich diese auf SimTalk 1.0. Diese Sprache kann nach wie vor auch in aktuellen Software-Versionen verwendet werden und ist nicht obsolet.

Die wichtigsten Konstrukte wie wenn-dann-Funktionen oder Schleifen sind in der App PSIMA implementiert. Zur Veranschaulichung sind in Abbildung 5.8 oben die Block-Bibliothek und unten der Block-Editor dargestellt. Im Block-Editor wurde ein beispielhaftes Programm in der neuartigen VBPIs erstellt. Dem in der Abbildung 5.8 in Form von Blöcken dargestellten Programm sind die Anweisungen im Quellcode 5.1 in der Sprache SimTalk gegenübergestellt. Es ist ein Compiler erforderlich, der die VBPIs, die in der App zum Einsatz kommt, in SimTalk übersetzt. Es folgt nun eine Erläuterung der in der App implementierten Funktionalitäten hinsichtlich der VBPIs und daraufhin eine kurze Beschreibung des Compilers.

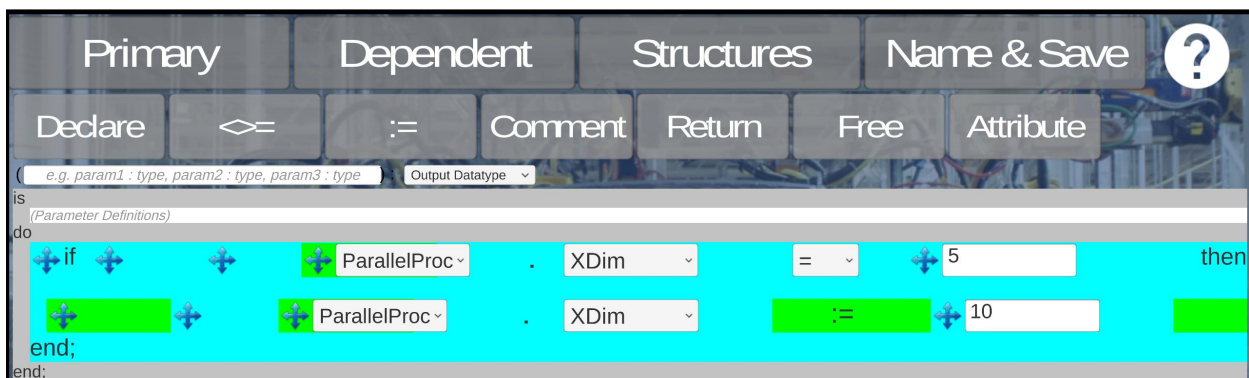


Abbildung 5.8: Bildschirmfoto der VBPIs: Ein beispielhaftes Programm wurde vom Benutzer zwischen dem „do“ und „end;“ erstellt

Quellcode 5.1 Skript aus Abbildung 5.8 in der Sprache SimTalk

```

is
do
    if ParallelProc.XDim = 5 then
        ParallelProc.XDim := 10;
    end;
end;

```

Abbildung 5.9 zeigt den Block-Editor der VBPI in der App. Die Block-Bibliothek ist in die vier Teile *Primary*, *Dependent*, *Structures* und *Name & Save* untergliedert. In jedem Teil befinden sich Blöcke, die der Nutzer antippen und in die weiß hervorgehobenen Felder zwischen die Schlüsselworte *is* und *do* sowie *do* und *end;* durch erneutes Tippen einfügen kann, sogenannte Instanziierung. Zwischen dem *is* und *do* können ausschließlich Variablen definiert werden. Der Bereich zwischen dem *do* und *end;* ist für das individuelle Programm des Anwenders bestimmt. Über dem *is* kann die Übergabe von Parametern und deren Spezifikation des Datentyps hinterlegt werden. Tabelle 5.5 fasst die dem Lernenden in der Block-Bibliothek bereitgestellten Blöcke kurz zusammen.

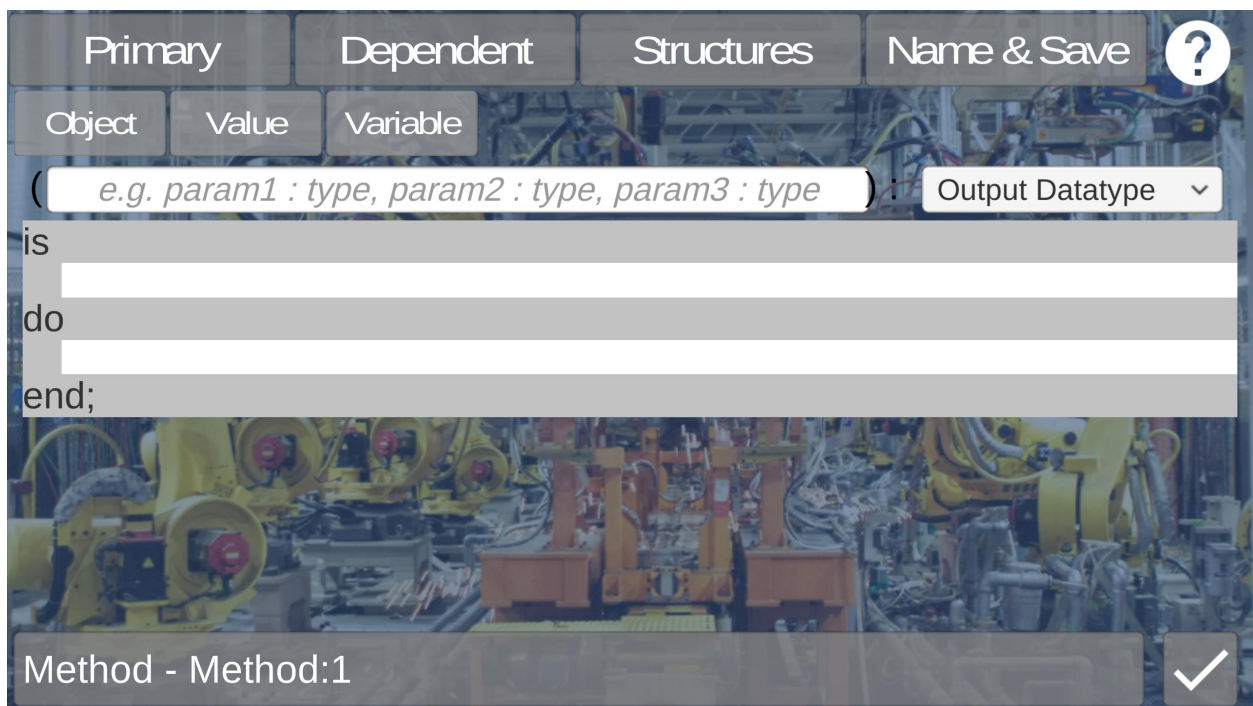


Abbildung 5.9: Bildschirmfoto der Benutzungsoberfläche für die VBPI

Primary: In dieser Rubrik sind Blöcke zusammengefasst, die für sich allein stehen können. Hierzu zählen Blöcke zum Deklarieren von Variablen (*Declare*), Vergleich von Werten ($<>=$), zur Zuweisung von Werten ($:=$), zum Einfügen von Kommentaren (*Comment*), Zurückgeben von Werten (*Return*), freien schreiben von Quellcode (*Free*) sowie zur Auswahl von Attributen (*Attribute*).

Dependent: Hier sind Blöcke gruppiert, die nur als Bestandteil in anderen Blöcken zum Einsatz kommen. Beispielsweise macht es keinen Sinn, einen Wert (Block *Value*) ohne den Kontext eines weiteren Blocks wie zum Vergleich (*Compare*) zu verwenden. Zu den in dieser Gruppe umgesetzten Blöcken gehört der Block *Object*. Er bietet durch ein Auswahlfeld die Möglichkeit, auf im Modell instanziierte Bausteine und Objekte zu referenzieren. Weiterhin ist der Block *Value* implementiert. Dieser erlaubt es Nutzern, bspw. einen Wert des Datentyps String oder Integer einzugeben. Der Block *Variable* ermöglicht die Manipulation von Werten von zuvor zwischen dem *is* und *do* definierten Variablen.

Structures: In diesem Bereich sind Programm-Strukturen zusammengefasst. Hierzu zählt die *If-Struktur*, die aus den Schlüsselworten *If*, *Then* und *End*; besteht. Weiterhin existiert auch eine *If-Else-Struktur*, die die *If-Struktur* um ein *Else* erweitert. Hinsichtlich der Schleifen ist eine *For*- sowie eine *While-Struktur* implementiert. Schließlich ist auch eine *Inspect-Struktur* verfügbar, die als Alternative zu einer verschachtelten *If-Else-Struktur* verwendet werden kann.

Name & Save: In dieser Rubrik kann der Anwender den Namen der spezifischen Methode ändern. Des Weiteren befindet sich in der Oberfläche auch eine Schaltfläche zum Speichern des implementierten Quellcodes. Diese Schaltfläche führt die gleiche Funktion aus wie der Haken im unteren Balken.

Tabelle 5.5: VBPIs: In der Block-Bibliothek der App bereitgestellte Blöcke

Name	Kürzel	Kurzbeschreibung
Declare	Var	Deklaration von Variablen.
<>=	Com	Vergleich von Attributen/Variablen.
:=	Set	Zuweisung von Werten.
Comment	Comment	Kommentar.
Return	Return	Definition eines Rückgabewerts einer Methode.
Free	Free	Freies schreiben von Quellcode.
Attribute	Prop	Zugriff auf Attribute von Objekten.
Object	Ref	Adressieren von Objekten.
Value	Value	Eingabe von Werten durch Benutzer.
Variable	VarRef	Erlaubt den Zugriff auf eine durch den Benutzer definierte Variable.
if	If	Verzweigung.
if ... else	If-Else	Verzweigung mit einem else-Zweig.
while	While	Beschreibung von while-Schleifen.
for	For	Beschreibung von for-Schleifen.
inspect	Inspect	Inspect-Anweisung zur Verwendung bei einer Vielzahl an Alternativen.
	Main	Block, der die Struktur von SimTalk durch <i>Is</i> , <i>Do</i> und <i>End</i> ; repräsentiert.

Die Oberfläche der VBPIs in PSIMA ist so gestaltet, dass sich die Blöcke gemäß den syntaktischen Regeln der Sprache SimTalk instanziierten lassen. Bei der Nutzung von Strukturen, die weitere Argumente wie eine Bedingung bei der *If-Struktur* erfordern, erhält der Lernende während der Modellierung den Hinweis *Condition* an der entsprechenden Stelle im Code der VBPIs angezeigt, nachdem die Struktur durch Tippen instanziiert wurde. Dieser Hinweis soll dem Lernenden helfen, den folgerichtigen Block (<>=) an entsprechender Stelle im Code zu verwenden, um eine syntaktisch korrekte Methode zu programmieren.

Neben der grafischen Oberfläche der VBPIs ist ein Programm zur Übersetzung der Sprachen zwischen PSIMA und Tecnomatix Plant Simulation erforderlich. Anforderungen an das im Folgenden Compiler genannte Programm sind, dass es über die Datenbank speicherbar ist, möglichst wenig Speicherplatz beansprucht und die Syntax sowie Semantik korrekt übermittelt. Die Funktionsweise des Compilers ist im Anhang G dokumentiert. Im Anhang G ist darüber hinaus ein konkretes Minimal-Beispiel abgedruckt. Die Arbeitsweise des Compilers ist darin nachvollziehbar beschrieben.

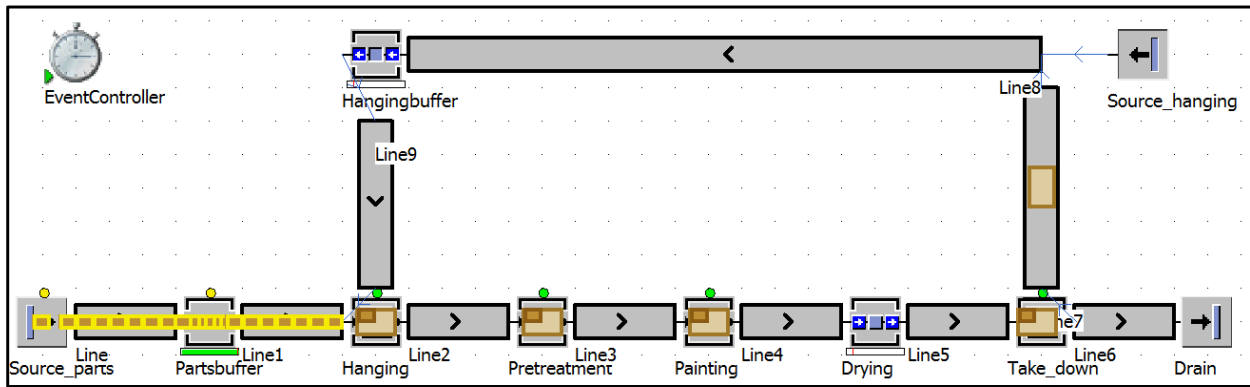
Im Kontext der IKS werden Lernende schrittweise in die Nutzung der Programmiersprache zur Implementierung von Simulationsmodellen eingearbeitet. Die Tabellen D.1 bis D.4 im Anhang D zeigen, welche Bausteine/Objekte erstmalig je Level durch den Lernenden zu instanziierten sind. Eine fehlerhafte Syntax kann durch die VBPIs nahezu ausgeschlossen werden, da nur „passende“ Blöcke durch den Benutzer instanziiert werden können. Indem in der VBPIs alle relevanten Trennzeichen wie „;“ in den Blöcken abgebildet sind, wird der Anwender beim Erlernen der Syntax unterstützt. Zutraglich zur Einarbeitung in die Semantik ist, dass Lernende im Sinne der IKS zunächst den in der Aufgabenbeschreibung gezeigten Code replizieren, später erstellen sie ihn basierend auf Pseudocode und schließlich auf Basis einer textuellen Beschreibung.

5.1.7 Animation

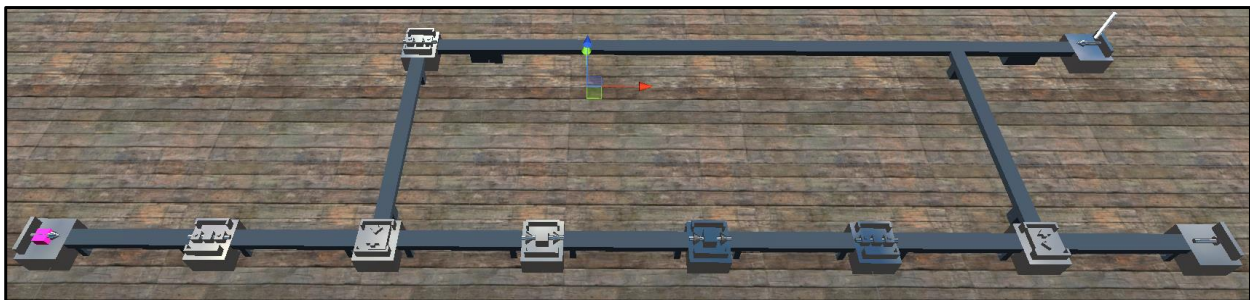
RABE ET AL. (2008) benennen die Animation als Technik der Validierung und Verifikation [Rab-08, S.95ff.]. Da die Animation sehr förderlich für das Verständnis des Lernenden hinsichtlich des Modellverhaltens ist, ist im Folgenden eine exemplarische Umsetzung diskutiert. Vorgestellt ist ein Ansatz, der im Ergebnis neben einer dynamischen zweidimensionalen auch eine dreidimensionale immersive Darstellung zulässt. Dies ist möglich, da die App PSIMA in einer Game-Engine erstellt wurde. Game-Engines sind sehr mächtige Werkzeuge im Hinblick auf die Visualisierung und Animation. Exemplarisch soll dieses Potenzial an einem Simulationsmodell namens *PaintShop* ausgeschöpft werden. Es wird im Folgenden aufgezeigt, wie solch eine Animation in PSIMA erzeugt und gestaltet sein kann. Es handelt sich um einen Ansatz der *Fern-Simulation und Visualisierung / Animation*.

Der *PaintShop*, vgl. Abbildung 5.10, bildet eine Lackieranlage ab, in welcher Haken in einem geschlossenen Kreislauf zirkulieren. Die Haken nehmen an der Station *Hanging* ein zu lackierendes Teil auf, welches wiederum von der *Source_parts* erzeugt wird. Ein beladener Haken bringt das zu lackierende Teil zu den Stationen *Pretreatment*, *Painting* und *Drying*. In der Station *Take_down* werden Haken und Teil voneinander getrennt. Der Haken geht zum *Hangingbuffer*, das lackierte Teil in die *Drain*.

Ausgangspunkt ist das in dem Simulationsinstrument Tecnomatix Plant Simulation erzeugte Modell. Dabei ist es unerheblich, ob das Simulationsmodell (teil-)automatisch durch den Algorithmus zur Modellgenerierung oder manuell durch einen Simulationsanwender aufgebaut wurde. Methoden werden dazu verwendet, die stationären Bausteine und permanenten Objekte, bspw. Quellen, Einzelstationen oder Puffer, hinsichtlich deren Position und Orientierung im Modell zu erfassen. Des Weiteren werden die Positionen temporärer physischer Objekte, die Teile und Haken, mit dem Eintritt eines Er-



(a) Bildschirmfoto des Simulationsmodells *PaintShop* in der Software Tecnomatix Plant Simulation; vgl. Symbole von Bausteinen/Objekten mit Tabelle C.2 im Anhang C



(b) Dreidimensionale Animation des Simulationsmodells in der Game-Engine Unity3D

Abbildung 5.10: Bildschirmfoto Simulationsmodell und Animation in Game-Engine

eignisses in eine Tabelle geschrieben. Die in die Datenbank exportierten Tabellen sind im Anhang H, in Abbildung H.1, auszugsweise für das Modell *PaintShop* dokumentiert. Diese und die im Folgenden benannten Abbildungen und Quelltexte sind im Anhang H abgedruckt. Die Daten werden von der Datenbank in die Software Unity3D eingelesen. Der Quellcode H.1 zeigt die Schritte des Datenimports im Detail auf. Die stationären Bausteine und permanenten Objekte, siehe Abbildung H.1a, werden im nächsten Schritt instanziiert, vgl. Quellcode H.2. In Unity3D wurden eine Quelle, ein Puffer, eine Montagestation, Einzelstation, Demontagestation und ein Lager 3D-grafisch vorbereitet. Skript H.3 dient während der Laufzeit der Animation als Grundlage für die lineare Interpolation der Bewegungen anhand der Positionsdaten der temporären physischen Objekte, vgl. Abbildung H.1b. Kern der linearen Interpolation ist die open-source Lösung *iTween* [iTW-19]. *iTween* animiert temporäre physische Objekte von einer gegebenen Koordinate hin zu einer anderen während eines gegebenen Zeitintervalls. *iTween* erzeugt also eine kontinuierliche Transformation der Objekte auf der Basis der übergebenen Ereignisliste. Zusammenfassend zeigt Abbildung 5.10a das exemplarische Simulationsmodell *PaintShop* in der Software Tecnomatix Plant Simulation und Abbildung 5.10b im Ergebnis die animierte dreidimensionale Darstellung in der Software Unity3D. Ein Unity3D Plugin ermöglicht das Rendern der Darstellung auch für die dreidimensionale immersive stereoskopische Darstellung im Sinne der VR. Dies ist beispielhaft in Abbildung H.2 im Anhang H gezeigt.

Diese VR-Umgebung lässt sich mit einer handelsüblichen VR-Brille wie der Samsung Gear VR betrachten. Zur Steuerung der Perspektive ist ein Gamepad wie das EI-GP20 der Firma Samsung sinnvoll. Vorteilhaft

an dem benannten Ansatz zur Animation ist, dass diese um weitere grafische Elemente erweitert werden kann, zumal diese mit einer hierauf spezialisierten Game-Engine realisiert ist. Als negative Eigenschaft des Ansatzes ist die Latenz zwischen Simulationsstart und Animation zu benennen. Für einen definierten Simulationszeitraum, z.B. 30 min., müssen zuerst die relevanten Ereignisse aus der Ereignisliste des Simulationsinstruments gesammelt werden.

Mit der Implementierung des Anwendungsbeispiels *PaintShop* ist der Nachweis erbracht, dass sich eine Animation mit dem zuvor beschriebenen Verfahren in einer mobilen Umgebung anschaulich verwirklichen lässt. Im Rahmen der mobilen App PSIMA kann solch eine Umgebung einen Beitrag zum Verständnis des Lernenden über das Modellverhalten und somit zu einer hohen Qualität der Planung leisten.

5.1.8 Erprobung und Chancen

Die App PSIMA wurde von einer Studierendengruppe hinsichtlich des Beitrags zur Einarbeitung in das Simulationsinstrument Tecnomatix Plant Simulation erprobt. In den folgenden Abschnitten sind die Planung der Erprobung, deren Durchführung sowie die Ergebnisse hinsichtlich der Häufigkeit der Installation, des Motivationsempfindens, des Nutzungsverhaltens und der Wissensvermittlung dokumentiert.

Die teilnehmenden Studierenden sind überwiegend in den Studiengängen Produktionstechnik und Wirtschaftsingenieurwesen eingeschrieben. Begleitend zu einer Lehrveranstaltung über ereignisdiskrete Simulation im Kontext des Anwendungsbereichs der Planung von Materialflusssystemen wurde 31 Studierenden die App zur Nutzung auf ihren Smartphones angeboten. Die Studierenden konnten vor der Nutzung von PSIMA kaum oder keine Vorkenntnisse in ereignisdiskreter Simulation vorweisen. Die Probanden erhielten keine konkrete Vorgabe zur Nutzungsintensität der App. Während des 19-tägigen Versuchszeitraums wurde das Nutzungsverhalten der Probanden analysiert. Durch einen Fragebogen ist das Empfinden der Probanden erfasst worden. Folgende Hypothesen stehen im Vordergrund: [Mül-17a]

Hypothese 1 - Installation: Die Probanden zeigen Bereitschaft zur Installation der App auf ihren mobilen Endgeräten. Freiwillig installieren die Lernenden die App PSIMA und übergeben mindestens ein Simulationsmodell an das Simulationsinstrument.

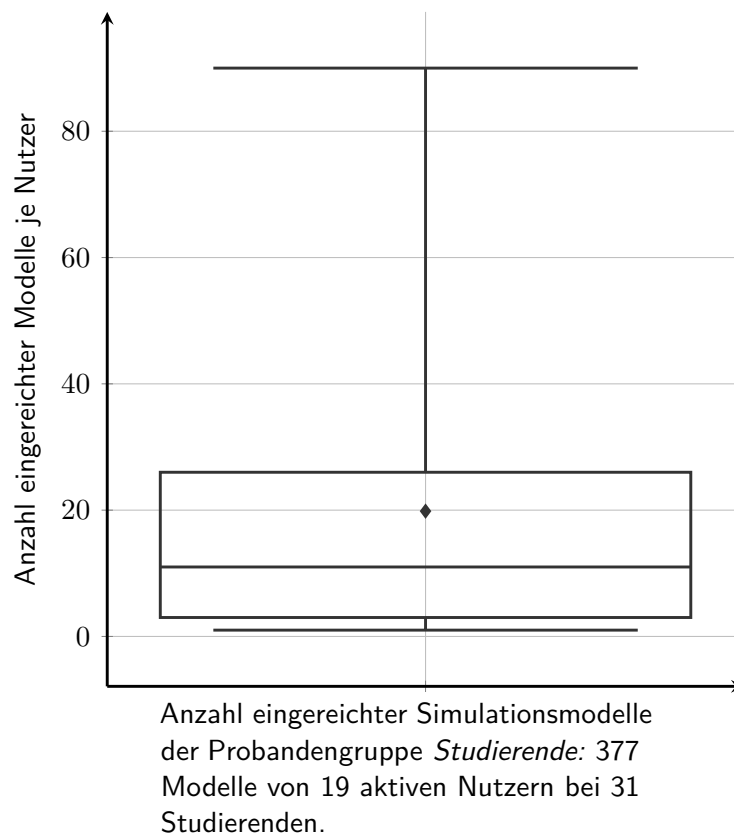
Hypothese 2 - Motivationsempfinden: Die Probanden empfinden die App als motivierend, um IKS aufzubauen. Das subjektive Empfinden der individuellen Motivation wird über einen Fragebogen ermittelt. Ergebnisse aus der Befragung werden mit Nutzungsdaten von PSIMA korreliert. Hierzu zählen z.B. die Anzahl absolvierter Level je Nutzer.

Hypothese 3 - Nutzungsverhalten: Die Anwender nutzen PSIMA mehrmals in kurzen Zeitabständen, z.B. während Reise- oder Wartezeiten oder kurzen Kreativpausen. Zur Überprüfung der Hypothese erfolgt eine Auswertung der Nutzungsdaten und somit des tatsächlichen Nutzungsverhaltens der Probanden.

Hypothese 4 - Wissensvermittlung: Nach der Nutzung der Lernumgebung sind die Probanden in der Lage, für einfache Problemstellungen Simulationsmodelle in dem Simulationsinstrument Tecnomatix Plant Simulation aufzubauen und zu simulieren. Zur Validierung der Hypothese wird eine textuelle Aufgabe gestellt. Diese gilt es unter Nutzung des Simulationsinstruments Tecnomatix Plant Simulation in begrenzter Zeit zu bearbeiten.

Insgesamt wurden während einer 19-tägigen Testphase 377 Simulationsmodelle durch Studierende in der App PSIMA erstellt und an das Simulationsinstrument übergeben [Mül-17a]. Die benannten Daten wurden nach der Testphase anonymisiert am Webserver und von der Datenbank abgegriffen, aufbereitet und interpretiert.

Mit Blick auf die **Hypothese 1 - Installation** ist festzustellen, dass von den 31 eingeladenen Probanden 19 die App nutzten (61%). Im Folgenden werden diese Nutzer als *aktive Nutzer* bezeichnet. 18 dieser aktiven Nutzer schafften es, ein oder mehrere Level erfolgreich abzuschließen. Diese sind im Folgenden als *aktive und erfolgreiche Nutzer* bezeichnet. Die statistische Verteilung der Anzahl der durch die aktiven Nutzer erzeugten Simulationsmodelle ist als Boxplot (Intervalle mit Quartilen) in Abbildung 5.11 dargestellt. Die Daten zeigen, dass die aktiven Nutzer die App mit durchschnittlich je knapp 20



Legende:

Minimum:	1	Median:	11
Maximum:	90	Mittelwert:	19,84

Abbildung 5.11: Statistische Verteilung der Anzahl durch aktive Nutzer erzeugter Simulationsmodelle: Der Boxplot zeigt Intervalle mit Quartilen

Einreichungen verwendeten. Da die App zum Zeitpunkt der Versuchsdurchführung nicht über einen App-Store zur Verfügung stand, sondern durch ein manuelles Prozedere zu installieren war, verzichteten mehrere interessierte Studierende auf eine Installation. Es ist davon auszugehen, dass sich die Bereitschaft von Studierenden zur Installation von PSIMA erhöht, wenn die App im App-Store zur Verfügung steht.

Mit Blick auf die **Hypothese 2 - Motivationsempfinden** ist in Abbildung 5.12 je Level die Anzahl der eingereichten Simulationsmodelle gezeigt. Die Levelpacks eins und zwei wurden dahingehend erprobt. Es ist ersichtlich, dass die Anzahl eingereichter Simulationsmodelle mit steigendem Level sinkt. Während der ersten Level ist die Anzahl der eingereichten Simulationsmodelle relativ konstant, während diese ab Level 1-7 kontinuierlich abnimmt. Die besonders hohe Zahl eingereichter Simulationsmodelle zur Lösung von Level 1-10 stellt einen Ausreißer dar. Ein Proband übermittelte knapp 50 Simulationsmodelle, bis er das richtige Ergebnis erzielte.

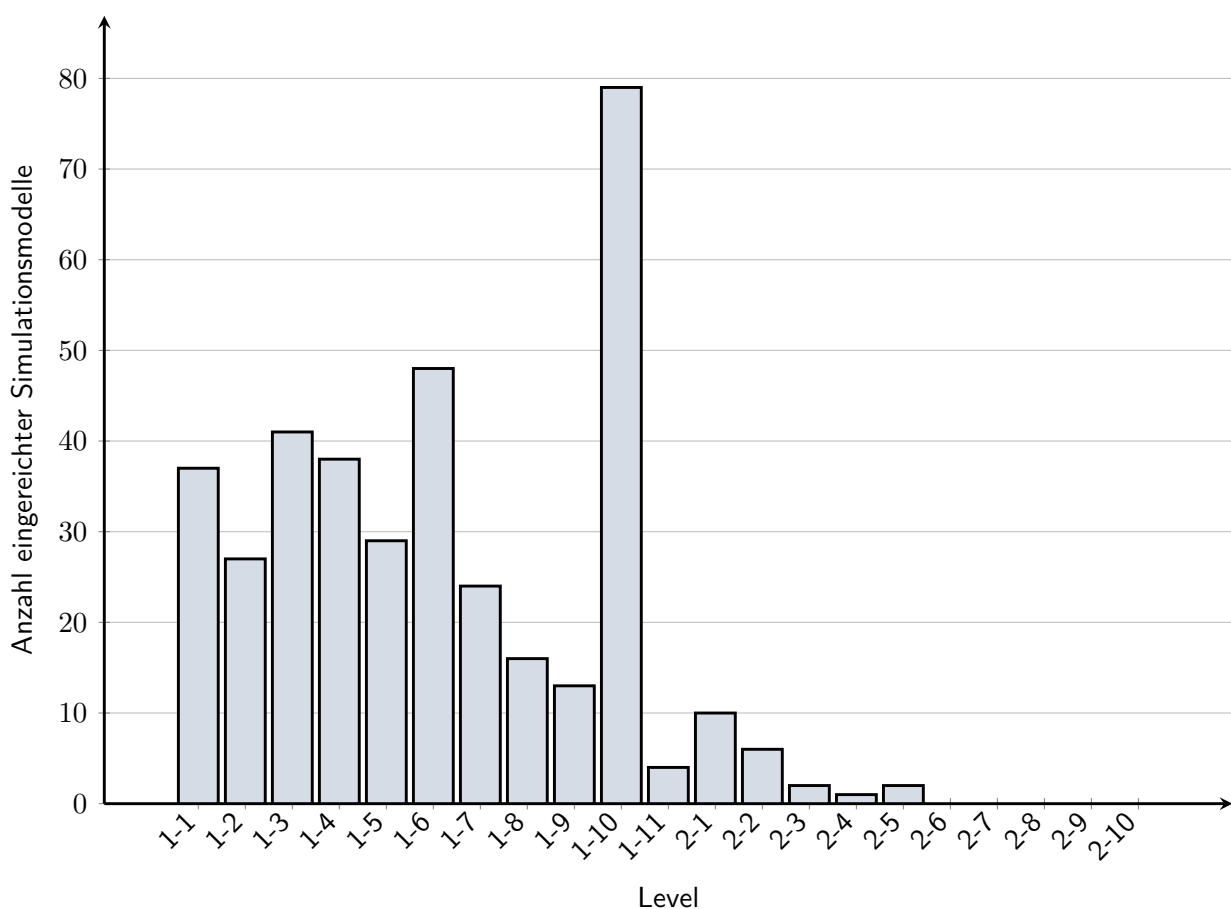


Abbildung 5.12: Eingereichte Simulationsmodelle je Level nach [Mül-17a]

Abbildung 5.13 zeigt die Anzahl erfolgreich abgeschlossener Level. Beispielsweise wurde Level 1-6 im Vergleich zu 1-5 weniger oft erfolgreich abgeschlossen, drei zu sieben Probanden. Stellt man dazu die eingereichten Simulationsmodelle gegenüber, vgl. Abbildung 5.12, wird deutlich, dass Level 1-6 offenbar schwerer zu lösen ist als die vorangegangenen. Bei höherer Anzahl übermittelter Simulationsmodelle liegt eine geringere Erfolgsquote vor. Dies könnte bei manchem Probanden eine demotivierende Wirkung gehabt haben. Gleichzeitig schlossen erneut sieben Probanden die Level 1-7 bis 1-9 erfolgreich ab und

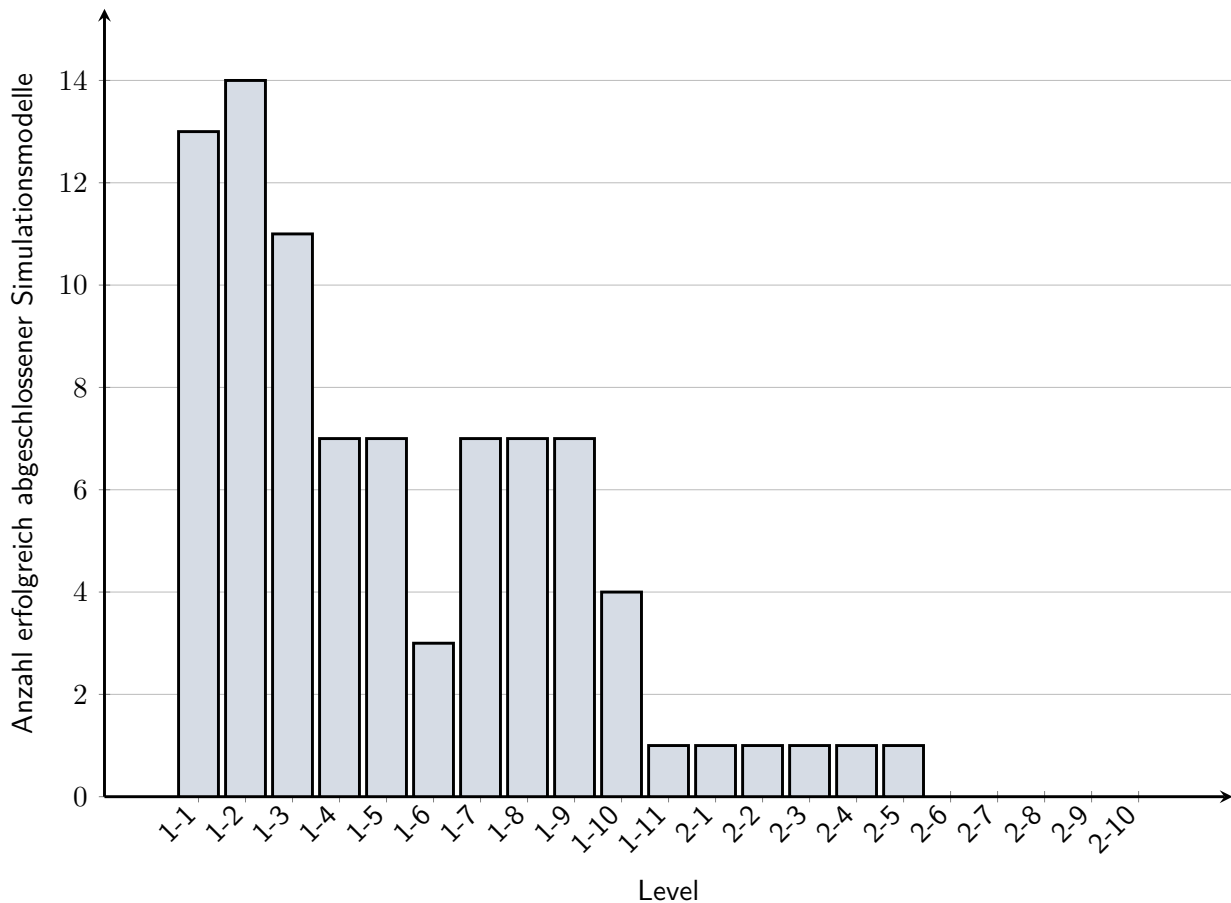


Abbildung 5.13: Erfolgreich abgeschlossene Level nach [Mül-17a]

benötigten dazu weniger Versuche, vgl. Abbildung 5.12. Dieser Zusammenhang legt die Vermutung nahe, dass bereits ein Lerneffekt eingetreten ist.

Die benannte Hypothese lässt sich durch zwei Aussagen stützen. Zum einen wurden knapp 400 Modelle zur Simulation übermittelt, obwohl den Probanden keine Vorgaben zur Intensität der Nutzung gemacht wurden. Zum anderen wird die Hypothese auch durch die Fragebögen gestützt. Die Probanden wählten auf einer Skala mit fünf Stellen zwischen den Aussagen „Instead of playing, I could have done something better“ und „I was bored anyway“. 17 von 23 Stimmen können dem mittleren Bereich zugeordnet werden. Es lässt sich also ableiten, dass die App kaum aufgrund von Langeweile noch während der Bearbeitung anderer wichtiger Aufgaben genutzt wurde. Anders formuliert lässt sich feststellen, dass die Bearbeitung der Level weniger als wichtigste Aufgabe wahrgenommen wurde und auch nicht als „Lückenfüller“ für Phasen von Langeweile diente. Als demotivierend wurden Software-Fehler der App benannt, die teilweise den Levelabschluss verhinderten. Ein Großteil dieser Fehler wurde zwar noch während des Erprobungszeitraums behoben, aber dennoch als störend von den Probanden wahrgenommen.

Dass sich die Lernenden intensiv und freiwillig mit der mobilen Umgebung beschäftigt haben, wird durch die hohe Anzahl eingereichter Simulationsmodelle deutlich. Die Studierenden lobten das bislang meist unbekannte Mittel der digitalen Spiele zur Instruktion.

Mit Blick auf **Hypothese 3 - Nutzungsverhalten** zeigt Abbildung 5.14 die Anzahl aktiver Probanden in Abhängigkeit von der Uhrzeit über alle Tage des Nutzungszeitraums. Bei den Studierenden traten mehrere Häufungen auf: 11-13 Uhr, 14-15 Uhr und 17-18 Uhr.

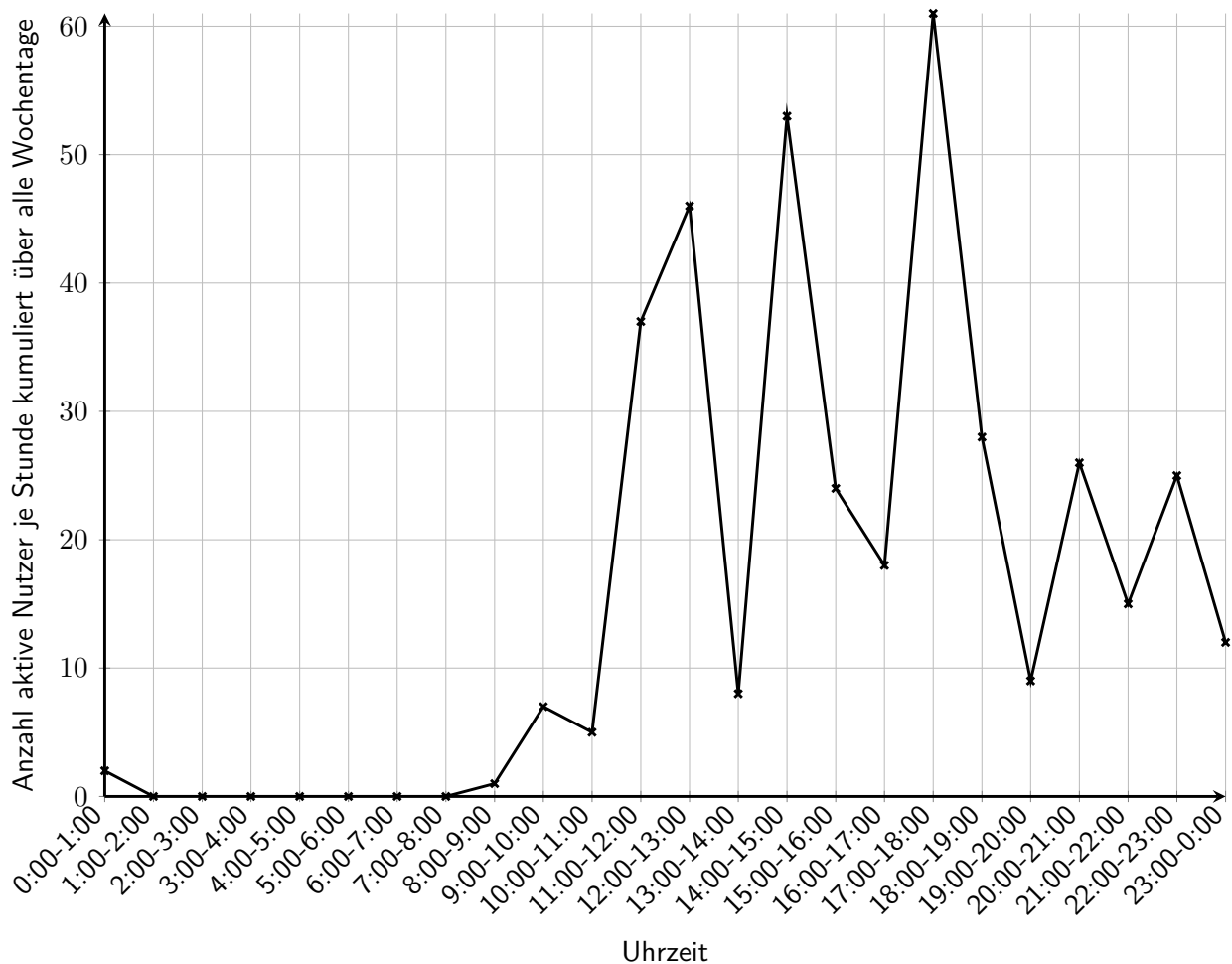


Abbildung 5.14: Summe aktiver Probanden je Stunde kumuliert über alle Tage des Studienzeitraums nach [Mül-17a]

Abbildung 5.15 zeigt die Nutzungsintensität in Abhängigkeit der Uhrzeit über alle Tage des Nutzungszeitraums. Bei den Studierenden ist eine Aktivität mit Ø20 Simulationsmodellen je Proband festzustellen. Die intensivsten Nutzungszeiträume bei den Studierenden zeigen sich in der Tagesmitte gegen 14 Uhr und spät am Abend um 22 Uhr.

In der Befragung gaben die Studierenden an, die App überwiegend zuhause genutzt zu haben. Als zweithäufigste Nennung wurde „in öffentlichen Verkehrsmitteln“ angegeben. Die Daten lassen darauf schließen, dass die mobile Lernumgebung mehrmals in kurzen Zeitabständen genutzt wurde.

Betrachtet man die Anzahl der eingereichten Simulationsmodelle in Abhängigkeit des Wochentags, vgl. Abbildung 5.16, so zeigt sich eine deutlich erhöhte Aktivität der Studierenden an Mittwochen und Donnerstagen. Die begleitende Lehrveranstaltung für die Studierenden fand während des Untersuchungs-

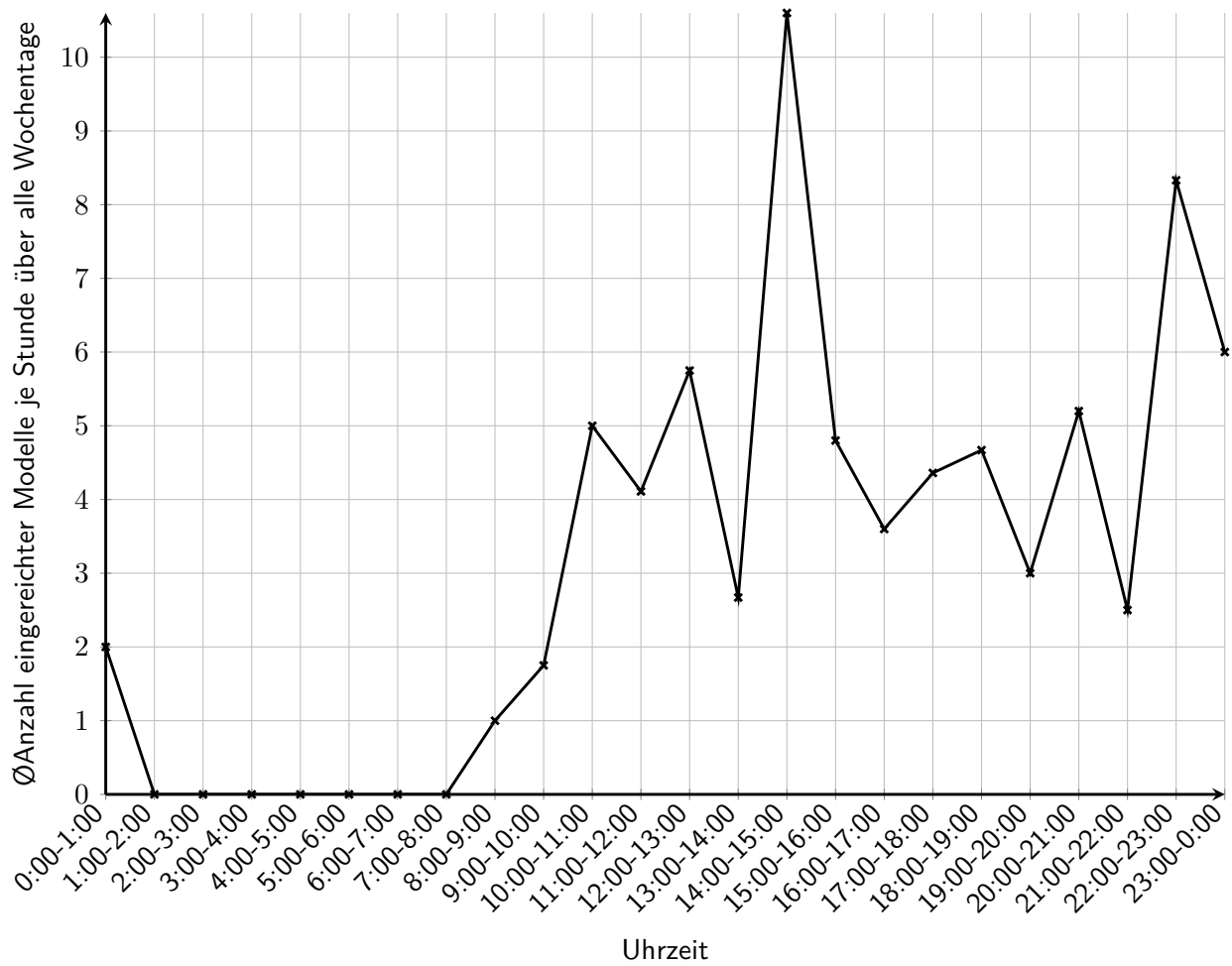


Abbildung 5.15: Durchschnittliche Nutzungsintensität je Stunde pro aktivem Nutzer nach [Mül-17a]

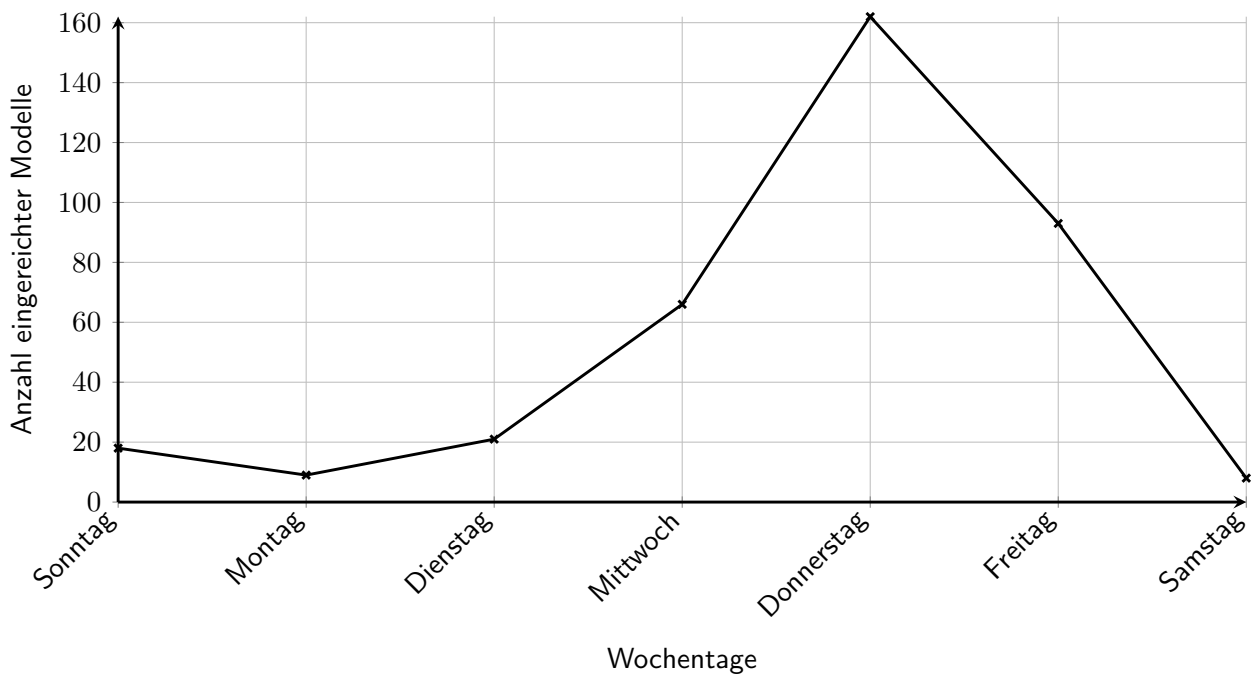


Abbildung 5.16: Kumulierte Anzahl eingereichter Simulationsmodelle nach Wochentagen

zeitraums an Donnerstagen statt. Offensichtlich haben die Studierenden die App überwiegend zur Vor- und Nachbereitung der Veranstaltung genutzt. Dass sich die App als Ergänzung zum regulären Lehrbetrieb eignet, kann durch die Wirkungsweise von Gamification begründet werden. Gamification und die Notengebung erzeugen meist extrinsische Motivation beim Studierenden. Aus persönlicher Erfahrung ist der studentische Fokus auf Noten maßgeblich, um diese zu einer Handlung zum Wissenserwerb zu bewegen. PSIMA ist durch bloßen Fokus auf Punkte und Abzeichen weniger extrinsisch motivierend einzustufen. Das bedeutet, dass Aktivitäten zum Erwerb von Noten durch den Studierenden meist als wichtiger bewertet werden. Folglich eignet sich die App als Ergänzung zum regulären Lehrbetrieb.

Zur Untersuchung von **Hypothese 4 - Wissensvermittlung** wurden einem Probanden ohne Vorwissen in der Simulationstechnik nach der Nutzung von PSIMA zwei Aufgaben gestellt, die in dem Simulationsinstrument Tecnomatix Plant Simulation zu lösen waren. Die erste war identisch mit der Beschreibung eines Levels der Lernumgebung, die zweite erforderte die Verwendung bzw. Implementierung bislang unbekannter Bausteine, Objekte und Abläufe. Der Proband konnte ohne weitere Hilfe das Simulationsmodell der ersten Aufgabe korrekt aufbauen. Zum Simulationsstart war jedoch etwas Hilfe erforderlich, da sich das Symbol zum Start der Simulation zwischen dem Simulationsinstrument und der App unterscheidet. Die zweite Aufgabe konnte erst nach kurzer Erläuterung der erforderlichen, bislang unbekannten Objekte, modelliert und gelöst werden. Lernende finden sich nach der Nutzung der App in der Bedienung des Simulationsinstruments zurecht. Sie erkennen Objekte/Bausteine im Simulationsinstrument wieder.

Insgesamt zeigt die Erprobung, dass PSIMA ein geeignetes Werkzeug ist, um Studierende motiviert in ereignisdiskrete Simulationsinstrumente einzuarbeiten. Studierende bevorzugen die mehrmalige Nutzung in kurzen Zeitabständen zur ortsunabhängigen Vor- und Nachbereitung von Lehrveranstaltungen.

PSIMA ist ein Prototyp mit Blick auf den Stand der Umsetzung. In etwa die Hälfte der Probanden wünschen sich Verbesserungen der Usability und kleinere Fehlerbehebungen. Der Vollständigkeit halber seien die Grenzen der beschriebenen Erprobung kurz diskutiert. Mit Blick auf die *Stichprobe* weisen HOAD UND KUNC (2015) darauf hin, dass Ergebnisse, die mit einer Kohorte an Studierenden erzielt wurden, Abhängigkeiten zu deren Auswahl haben können [Hoa-15]. Gemäß der subjektiven Einschätzung des Lehrenden bestanden keine wichtigen Unterschiede zu anderen Studierendengruppen.

Die Chancen und Entwicklungsperspektiven des App-Prototyps PSIMA lassen sich in die Bereiche *IT-Umsetzung* und *Lerninhalte* unterteilen. Das Kapitel wird mit einem Ausblick hinsichtlich der *potenziellen Anbieter und Nutzer* abgeschlossen.

Die IT-Architektur wurde so konzipiert, dass auch *Simulationsinstrumente anderer Hersteller* betrieben werden können. Notwendige Eigenschaften der Simulationsinstrumente sind die Unterstützung einer Programmiersprache zur Implementierung von Simulationsmodellen, Ereignisorientierung, Objektorientierung, Schnittstellen zu gängigen Datenbankmanagementsystemen, ein grafisch-interaktiver Modellaufbau unter Nutzung von Bausteinen/Objekten und die Möglichkeit zur Realisierung einer (teil-)automatischen Modellgenerierung. Bei einem Wechsel des Simulationsinstruments gilt es zu bedenken, dass PSIMA auf

die jeweiligen Symbole, Objekte und Bausteine inklusive deren Attribute sowie die Syntax der jeweils verwendeten VBPIs anzupassen ist.

Game-Engines sind hinsichtlich ihrer grafischen Eigenschaften als sehr leistungsfähige Programme zur Erstellung digitaler Spiele einzuschätzen. Da PSIMA in einer Game-Engine entwickelt wurde, können beispielsweise VR- oder AR-Anwendungen mit geringem Aufwand ergänzt werden. PSIMA wurde als sogenannte native App umgesetzt. Sie wurde also spezifisch für das jeweilige Betriebssystem entwickelt. Dies erlaubt die einfache Einbindung der Sensoren mobiler Endgeräte wie Kamera oder solche zur Quantifizierung der Beschleunigung. Eine Weiterentwicklung vom PSIMA hinsichtlich einer Animation im Sinne einer AR-Anwendung ist vorstellbar [Mun-17]. Die Lernförderlichkeit könnte durch solch eine hoch interaktive Projektion weiter erhöht werden.

Als weitere Chance für eine Weiterentwicklung sei die *Implementierung weiterer anwenderdefinierter Bausteine oder gar ganzer Bausteinkästen* benannt. Derzeit ist ein Bausteinkasten, der die wichtigsten vorkonfigurierten Objekte und Bausteine wie Quelle, Senke und Puffer zur Abbildung von Materialflusssystemen beinhaltet, umgesetzt. Ein anwenderdefinierter Bausteinkasten zur Abbildung von Multiagentensystemen wurde vorbereitet. Weitere Bausteinkästen wie der VDA-Bausteinkasten, der als Standard für den Automobilbau zu benennen ist [VDA-09] oder zum Aufbau von Simulationsmodellen im Bereich der Fertigung von Gasturbinen [Sto-17] können mit geringem Aufwand in PSIMA implementiert werden. Selbst für den Bereich der Krankenhausplanung ist ein Bausteinkasten vorstellbar [Har-02].

Die Weiterentwicklung der prototypischen Implementierung zur Verringerung der Latenz bedingt durch die Synchronisierung über den Webserver ist vorzunehmen. Die lokale SQLite Datenbank wird in festgelegten Synchronisierungsintervallen abgeglichen. Ereignisorientierte Ansätze wie *Firestore* oder *RethinkDB* scheinen geeignet zu sein, diese Latenz zu verringern. Bei *Firestore Realtime Database* handelt es sich um ein Produkt von Google [Goo-19a]. *Unity3D* unterstützt zwar *Firestore*, jedoch ist der Datenaustausch mit der Software *Tecnomatix Plant Simulation* noch zu klären, zumal keine direkte Schnittstelle zu *Firestore* vom Softwarehersteller angeboten wird. Weiterhin werden Daten auf Servern des Herstellers gespeichert. *RethinkDB* ist ein open-source Produkt [Ret-19]. Zwar können Daten auf eigenen Servern gespeichert werden, jedoch ist die Programmierung von Schnittstellen zu *Unity3D* und *Tecnomatix Plant Simulation* erforderlich.

Die Weiterentwicklung der grafisch-ästhetischen Darstellung der VBPIs ist erforderlich. Es scheint attraktiv, die von Google bereitgestellte Sprache *Blockly* [Goo-19b] näher auf die Übernahme von grafischen Eigenschaften in PSIMA zu untersuchen.

Der Fokus der prototypischen App ist auf die spielerische Einarbeitung von Studierenden in die Bedienung eines Simulationsinstruments gerichtet. *Lerninhalte* wie Beiträge zur *Sicherstellung der Qualität von Simulationsstudien* wurden von der Betrachtung oft ausgeschlossen. Diese wurden in der begleitenden Lehrveranstaltung eingehend adressiert. Hierzu gehören beispielsweise die Validierung und Verifikation. Übergeordnetes Ziel der Validierung und Verifikation besteht darin zu verhindern, dass „aus einer Simulationsstudie fehlerhafte Aussagen gewonnen werden, die zu einer Fehlentscheidung führen“ [Rab-08, S.2]. RABE ET AL. (2008) beschreiben in ihrem Buch *Verifikation und Validierung für die Si-*

mulation in Produktion und Logistik hierzu umfangreich Vorgehensmodelle und Techniken [Rab-08]. WIMPF ET AL. (2011) haben im Projekt simKMU bereits umfangreich beschrieben, wie diese relevanten theoretischen Inhalte unter Nutzung von Web 2.0 Technologien vermittelt werden können, vgl. die Ausführungen im Kapitel 2.5 [Wim-11]. Die ansprechende Integration theoretischer Inhalte kann in PSIMA dazu beitragen, die Instruktion unabhängig von einer begleitenden Lehrveranstaltung zu gestalten.

Potenzielle Anbieter einer marktreifen Version von PSIMA sind *Hersteller kommerzieller ereignisdiskreter Simulationsinstrumente*. Die App dient dazu, um auf deren Simulationsinstrument aufmerksam zu machen und Interesse bei potenziellen Kunden zu wecken. Neue Nutzer mit geringen Vorkenntnissen, die über einen projektbegleitenden Einsatz von ereignisdiskreter Simulation im Unternehmen nachdenken, können sich so einen ersten Überblick über das Simulationsinstrument verschaffen. Dies kann insbesondere bei KMUs relevant sein, die in der Regel keine Simulationsexperten beschäftigen [Ola-14]. Es ist davon auszugehen, dass Hersteller, die eine ansprechende Lernumgebung für ihr Simulationsinstrument anbieten, am Markt erfolgreicher sind. Potenzielle Nutzer werden somit bereits in einer frühen Phase des Erlernens auf das Simulationsinstrument aufmerksam.

Getrieben durch *Unternehmensberatungen* können die Ziele verfolgt werden, im Sinne einer Weiterbildung Berater oder Kunden in ein Simulationsinstrument einzuarbeiten.

Eine von Hochschulen angebotene App hat das Potenzial, mehr Studierende in entsprechende Lehrveranstaltungen zu locken. Darüber hinaus können Studierende in anderen Kursen wie zur Logistik oder Fabrikplanung Nutzen daraus ziehen. Um allgemeine Zusammenhänge wie zwischen dem Lagerbestand und der Auslastung von Maschinen aufzuzeigen oder die Layoutplanung abzusichern, kann die Lernumgebung im Sinne einer SIM in EDU-Anwendung zum Einsatz kommen.

5.2 Beispiele materialflusstechnischer Anwendungen

Unter Nutzung der im Kapitel 5.1 beschriebenen App PSIMA sind in diesem Kapitel Beispiele für materialflusstechnische Anwendungen aufgezeigt. Im universitären Versuchsfeld ist im Kapitel 5.2.1 die Erprobung des Vorgehens für das EEDS in einer Lernfabrik mit vietnamesischen Studierenden beschrieben. Basierend auf den industriellen Beispielen der Fertigung von Gasturbinen (Kapitel 5.2.2) und der Montage von Motorradantrieben (Kapitel 5.2.3) sind Potenziale für eine hohe Planungsqualität von Materialflusssystemen durch die Entwicklungen in dieser Arbeit im abschließenden Kapitel 5.2.4 abgeleitet.

5.2.1 Universitäres Versuchsfeld

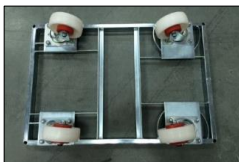
In diesem Kapitel ist die Erprobung des Vorgehens zum EEDS in Lernfabriken im Sinne der Durchführung der Lehre beschrieben. Die Erprobung fand im Kurs *Simulation of Production Systems* im

produktionstechnischen Labor der Vietnamesisch-Deutschen Universität (VGU) in Ho-Chi-Minh-Stadt, Vietnam, statt. Im Folgenden ist die Infrastruktur an der VGU sowie der Verlauf der Lehre dokumentiert [Mül-17b].

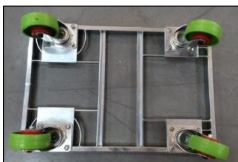
Auf sieben Arbeitsstationen (OPs) werden fünf Varianten sogenannter Trolleys montiert. Die OPs sind in einem U-Layout angeordnet und durch Rollenförderer verkettet. Die Trolleys werden in einem One Piece Flow montiert. In Abbildung 5.17a ist die Lernfabrik dargestellt. Die Abbildungen 5.17b bis 5.17f illustrieren die fünf Produktvarianten der Trolleys.



(a) Foto des U-Layouts der Lernfabrik an der VGU mit sieben OPs



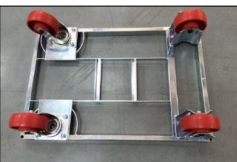
(b) 4 schwenkbare Räder, weiß



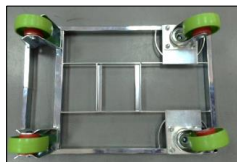
(c) 4 schwenkbare Räder, grün



(d) 4 schwenkbare Räder, rot



(e) 2 schwenkbare Räder, rot



(f) 2 schwenkbare Räder, grün

Abbildung 5.17: U-Layout der Lernfabrik an der VGU und Produkte, sog. Trolleys nach [Mül-17b]

An OP 1 wählt der Montierende einen von zwei möglichen Rahmen und legt diesen auf einen Werkstückträger. Je nach Variante werden zwei oder vier sogenannte Radhalter aufgelegt. Auf den OPs 2-5 werden je ein Radhalter durch das Festschrauben von vier Muttern mithilfe eines elektrischen Schraubers befestigt. Darüber hinaus wird je OP ein Rad durch das Einschieben einer Achse befestigt. Die Achse wird ebenso mit einer Mutter verschraubt. An OP 6 wird der Trolley durch zwei Schrauben mit einem Etikett versehen. Schließlich wird der Trolley an OP 7 in einen Karton verpackt. Die beschriebenen Vorgänge können in einer Taktzeit von einer Minute vollzogen werden und führen somit zu einer Durch-

laufzeit von sieben Minuten. In Tabelle I.1 im Anhang I sind die Arbeitsabläufe je Station detailliert beschrieben [Rei-16] [Vie-19].

Um die Simulationswürdigkeit sicherzustellen, wurden weitere Annahmen getroffen. Ein Engpass soll in der Lernfabrik entstehen, den die Studierenden simulativ erkennen und aufweiten sollen:

- Der Kunde entnimmt minütlich einen verpackten Trolley aus dem Lager der Fertigerzeugnisse. Im beschriebenen Beispiel ist der Produkttyp, Trolley 1-5, zufällig ausgewählt. Konkret ist eine Normalverteilung mit dem Mittelwert $\mu = 2$, Varianz $\sigma^2 = 3$, untere Schranke = 1, obere Schranke = 5, angenommen worden. Die Lernenden kennen diese Verteilung nicht.
- Initial ist ein Lagerbestand von zwei verpackten Trolleys je Variante verfügbar. Zu Beginn des Betriebs der Lernfabrik finden sich unfertige Erzeugnisse auf den OPs.
- Die Kapazität des Lagers wird als unbegrenzt angenommen.
- Falls ein Kundenwunsch nicht erfüllt werden kann, wird ein entgangener Gewinn von fünf Euro angenommen. Ein Kundenwunsch kann nicht zu einem späteren Zeitpunkt befriedigt werden. Dieser Annahme liegt zugrunde, dass es sich bei dem Trolley um ein ubiquitäres Produkt handelt, das auch bei Konkurrenten zu beziehen ist.
- Für jedes erfolgreich verkaufte Produkt werden fünf Euro Gewinn angenommen.
- Der Kunde bringt einen Trolley nach der Nutzungsphase zurück in die Lernfabrik. Dort wird der Trolley demontiert. Da die Anzahl der Rahmen, die in der Lernfabrik zirkulieren, auf 20 begrenzt ist und der Kunde in 30 Minuten Betriebszeit entsprechend 30 verpackte Trolleys nachfragt, müssen zwingend zurückgebrachte Trolleys durch die Lernenden demontiert werden.
- Maximal fünf Werker dürfen montieren.

Um die Profitabilität zu erhöhen, sind die Studierenden dazu angehalten, die Verfügbarkeit der fertig montierten Trolleys im Lager zu verbessern. Dies kann beispielsweise durch eine geringere Wiederbeschaffungszeit für vom Kunden entnommene Produktvarianten erreicht werden. Beispiele für Ideen und Entscheidungen, die von Lernenden getroffen werden, sind:

- Priorisierung von Montage oder Demontage: Entscheidungsvariablen sind der aktuelle Lagerbestand an verpackten Trolleys und die Anzahl der Rahmen, die für die Montage zur Verfügung stehen.
- Wahl der Produktvariante für Montage und Demontage: Entscheidungsvariablen sind der Lagerbestand an Trolleys und der aktuelle Bestand an verpackten Trolleys der jeweiligen Variante.
- Die Lernenden legen fest, wie die Arbeitsaufgaben unter den Workern verteilt werden.
- Die Lernenden definieren die Bestandsgrenzen für die Trolleys im Lager.
- Die Lernenden verschieben den Kundenentkopplungspunkt.

Lernende konkurrieren gruppenweise mit dem Ziel, sich in der Profitabilität zu überbieten.

Simulationswürdigkeit besteht gemäß der Richtlinie VDI 3633 bei günstigem Kosten-Nutzen-Verhältnis, bei Komplexität der Aufgabe, bei Unsicherheit bezüglich der Daten oder der mathematischen Verteilung streuender Werte und ihres Einflusses auf die Ergebnisgrößen und bei Fehlen analytischer mathematischer Modelle bzw. ungerechtfertigt hohem Aufwand zu deren Erstellung [VDI-3633a, S.19f.]. Im vorlie-

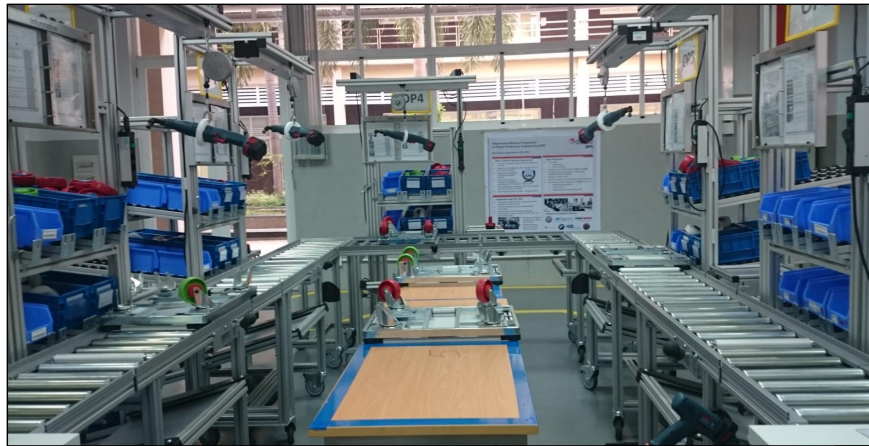
genden Beispiel wurde die Simulationswürdigkeit der Problemstellung mit Unsicherheit hinsichtlich der mathematischen Verteilung der streuenden Werte und deren Einfluss auf die Ergebnisgrößen begründet. Beispiele sind das zufällige Entnahmeverhalten des Kunden, die Zeiten zur manuellen Montage, Störungen bei der Montage oder Probleme in der Abstimmung zwischen den Werkern. Ein Sicherheitsbedürfnis durch die unscharfen Vorgaben hinsichtlich potenzieller Verbesserungen stützen die Simulationswürdigkeit zusätzlich. Da die Studierenden Ihre Ideen zur Verbesserung nicht iterativ durch den Betrieb der Lernfabrik überprüfen dürfen, herrscht Beweisnot.

Mit 30 Studierenden, eingeteilt in fünf Gruppen, wurde das Vorgehen zum EEDS unter Nutzung der Lernfabrik an der VGU erprobt. Die Lerneinheit dauerte fünf Tage und wurde als finale Aktivität einer Lehrveranstaltung zum Einsatz der ereignisdiskreten Simulation durchgeführt. Durch die Bearbeitung textueller Aufgaben wurden die Lernenden bereits in das Simulationsinstrument Tecnomatix Plant Simulation eingearbeitet. In einem anderen zuvor besuchten Kurs zum Thema Lean Management haben alle Lernenden bereits die Lernfabrik selbst betrieben. Folglich hatten die Studierenden bereits Fertigkeiten in der Montage von Trolleys.

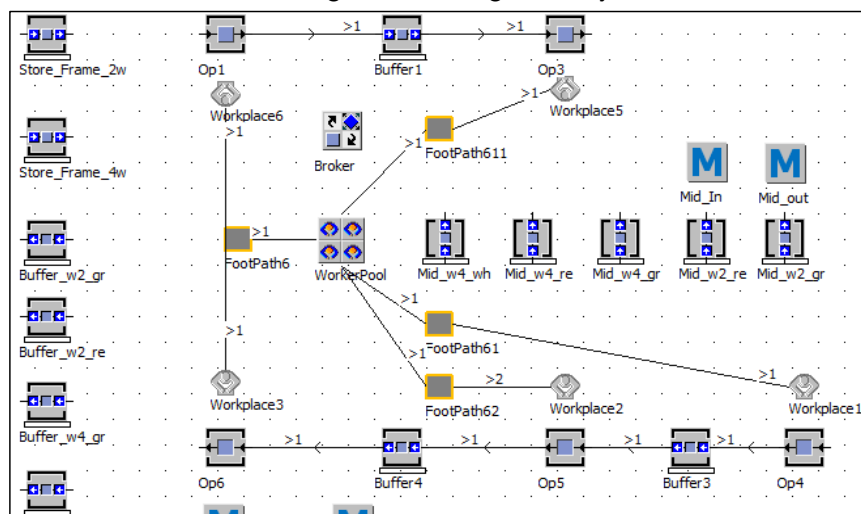
Am ersten Tag der Übung wurden zufällig ausgewählte Studierende aufgefordert, die Lernfabrik für eine Dauer von 45 Minuten zu betreiben. Alle weiteren Studierenden erfassten die aus ihrer Sicht relevanten Daten wie die Dauer der einzelnen Arbeitsschritte, die Wartezeiten von (un-)fertigen Erzeugnissen und Werkern als auch das Verhalten des Kunden, der jede Minute einen zufällig ausgewählten Trolley entnahm. Studierende beobachteten die Arbeitsaufteilung zwischen den Werkern und notierten deren Entscheidungen, beispielsweise wann montiert und wann demontiert wurde. Wie zu erwarten war, konnten in der beschriebenen Beobachtung nicht alle Kundenbedarfe befriedigt werden. Präzise ausgedrückt wurden 47% der Kundenwünsche erfüllt. Mit dem Ziel, die Profitabilität der Lernfabrik zu erhöhen, bestand das Ziel der Studierenden darin, die Verfügbarkeit von Fertigerzeugnissen im Lager zu verbessern.

Am zweiten und dritten Tag war den Studierenden der Zugang zur Lernfabrik verwehrt. Sie durften keine Experimente mit dem realen System durchführen. Dies wird als zulässige Annahme gesehen, da insbesondere in der Industrie kaum erhebliche Änderungen zum „Experimentieren“ vorgenommen werden können. Grund hierfür sind Sicherheitsrisiken, da z.B. Menschen verletzt werden oder Werkzeuge oder Geräte Schaden nehmen können. Somit konnten die erforderlichen Experimente ausschließlich mit einem durch Studierende erstellten digitalen Simulationsmodell der Lernfabrik vorgenommen werden. Hierzu bildeten die Studierenden ein Simulationsmodell auf Basis der zuvor aufgenommenen Daten und validierten bzw. verifizierten dieses. Ergebnisse aus den Simulationsläufen des Ist-Modells wurden mit den KPIs verglichen, die sich aus dem 45-minütigen Betrieb der Lernfabrik errechneten.

Am vierten Tag wurde die Lernfabrik für jede Studiengruppe eine Stunde zugänglich gemacht. In den ersten 30 Minuten konnten die Studierenden das von ihnen gewünschte initiale Layout herstellen. Dieser Zeitraum wurde bewusst knapp bemessen, um die Studierenden zur detaillierten Planung der Umsetzung ihrer simulativ ermittelten Maßnahmen zu bewegen. Typische simulativ durch die Studierenden ermittelte Verbesserungen sind Supermärkte zur Entkopplung. In den Supermärkten werden unfertige Erzeugnisse zwischengelagert. Ein konkretes Beispiel findet sich in Abbildung 5.18. Bild 5.18a zeigt die umgestaltete Lernfabrik. Die mittig zu sehenden Tische dienen den Studierenden als provisorische Su-



(a) Beispielhafte Verbesserung der Lernfabrik durch einen Supermarkt in Form von Tischen zur Pufferung von halbfertigen Trolleys



(b) Bildschirmfoto des zugehörigen Simulationsmodells in dem Simulationsinstrument Tecnomatix Plant Simulation; vgl. Symbole von Bausteinen/Objekten mit Tabelle C.2 im Anhang C

Abbildung 5.18: Beispiel für die Verbesserung des Layouts der Lernfabrik. Die Studierenden haben gezeigt, dass sie schneller auf Kundenbedarfe reagieren können [Mül-17b].

permärkte. Abbildung 5.18b zeigt das dazugehörige Simulationsmodell der Studierenden. Nach der Hälfte der Montagetätigkeiten eines Trolleys definierten die Studierenden diesen Supermarkt als Kundenentkopplungspunkt, um die Wiederbeschaffungszeit zu verringern. In den darauffolgenden 30 Minuten betrieben Studierende die Lernfabrik gemäß dem gruppenweise individuell simulativ ermittelten Ablauf. Um möglichst Beschleunigungseffekte aufgrund hoher Motivation bei der Ausführung der Arbeitsvorgänge zu vermeiden, wurde je eine konkurrierende Gruppe an Studierenden gebeten, Trolleys in der Lernfabrik zu montieren. Die montierende Gruppe stand unter Anleitung der Gruppe, die den verbesserten Ablauf der Montage simulativ ermittelte. Die nun beobachtende Gruppe stand der montierenden Gruppe mit Anweisungen und Empfehlungen zur Seite. Somit konnte sichergestellt werden, dass die Montierenden sich weitestgehend so verhielten, wie die Studierenden dies zuvor simulativ ermittelt hatten. Darüber hinaus nahmen die Studierenden erneut relevante Daten während der 30 Minuten auf.

Am letzten Tag präsentierten die Studierenden ihre Arbeit im Rahmen eines Kolloquiums. Die Studierenden waren dazu angehalten, ein besonderes Augenmerk auf die womöglich existenten Unterschiede zwischen dem Betrieb des verbesserten Simulationsmodells und dem Betrieb der physisch verbesserten Lernfabrik zu legen. Des Weiteren fertigten sie eine Dokumentation an.

Die Auswertung der studentischen Arbeiten zeigt, dass alle Gruppen mehr als ein Simulationsexperiment durchgeführt haben, um den Engpass zu beseitigen. Alle Gruppen stellten fest, dass der Engpass die sehr hohe Arbeitsbelastung einzelner Werker war bzw. die ungenügende Anzahl an Werkern, die montierten und demontierten. Alle Gruppen führten mindestens ein Simulationsexperiment mit veränderter Arbeitsteilung durch. Weiterhin entschieden sich die Studierenden dafür, Supermärkte zur Zwischenlagerung unfertiger Produkte zu simulieren, an denen der Kundenauftrag entkoppelt wurde, vgl. Abbildung 5.18b.

Vier der fünf Studierendengruppen konnten nach Verbesserung der physischen Lernfabrik den Kundenbedarf nicht wie simulativ erwartet befriedigen. Die vier Gruppen stellten fest, dass die Werker fehlerhafte Produkte herstellten oder langsamer als erwartet montierten. Darüber hinaus berichtet eine Gruppe über unzureichende Kommunikation mit den Werkern. Eine weitere benannte Verzögerung trat durch leere Akkus von Elektroschraubern auf, die während der Betriebszeit ausgetauscht werden mussten. Alle Studierenden waren nach der Übung in der Lernfabrik mit ihren Ergebnissen nicht zufrieden und dokumentierten ohne spezifische Aufforderung durch den Lehrenden weitere Simulationsexperimente.

Lediglich eine der fünf Studierendengruppen erreichte die Zielwerte, die sie auf Basis ihrer Simulationsstudie erwartet haben. Dies deutet darauf hin, dass die Mehrzahl der Studierenden falsche oder zu allgemeine Annahmen in der Modellierungsphase im Hinblick auf die relevanten Parameter vornahmen, die das Systemverhalten beeinflussen. Beispiele sind:

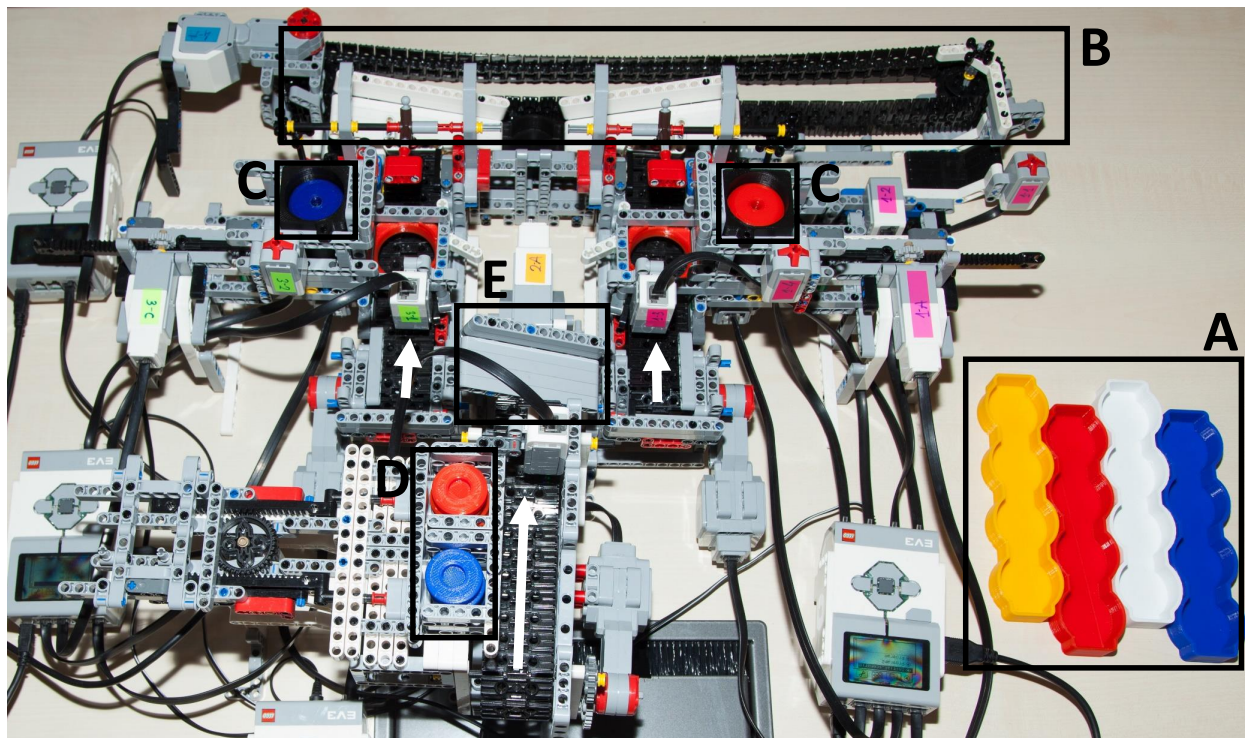
- Fehlerrate von Werkern und Ausschuss = 0%,
- keine Kommunikationsprobleme und
- Verfügbarkeit der Werkzeuge = 100%.

Dies lässt darauf schließen, dass die Studierenden während des Betriebs der verbesserten Lernfabrik bemerkten, die Realität zu sehr abstrahiert zu haben. Durch die Arbeit in der Lernfabrik konnten die Studierenden ein Bewusstsein über die Folgen unzureichender Abstraktion der Realität und Interpretation der Simulationsergebnisdaten erlangen. Dass die Studierenden freiwillig im Nachgang zur Veranstaltung weitere Verbesserungen simulativ ermittelten, lässt auf eine hohe Motivation schließen.

Es konnte gezeigt werden, dass das Vorgehen bei Anfängern anwendbar ist. Kein Studierender hatte Vorkenntnisse in der Durchführung von Simulationsstudien. Das Vorgehen zum EEDS trägt dazu bei, dass Simulationsstudien von erfahrenen Menschen durchgeführt werden, die hinsichtlich der Wichtigkeit der Abstraktion und Ergebnisinterpretation sensibilisiert sind. Zeitraubende Fehler lassen sich folglich vermindern.

Das Vorgehen zum EEDS sollte öfter als einmal durchlaufen werden, um die Möglichkeit zu schaffen, iterativ von vorherigen Erfahrungen zu lernen.

Lernfabriken sind nicht für alle Ingenieure oder Studierende zugänglich und in der Anschaffung und dem Betrieb häufig teuer. Physische Modelle von Materialflusssystemen, die günstig zu errichten und betreiben sind, können eine Lösung für dieses Problem sein. Abbildung 5.19 zeigt einen Ansatz mit Lego Mindstorms EV3.



Legende:

- A – Behälter
- B – Schleppkette
- C – Deckel
- D – Unterteil
- E – Wippe

weiße Pfeile – Richtung der Bewegung von Fördereinrichtungen

Abbildung 5.19: Materialflusssystem errichtet mit Lego als Alternative zur Lernfabrik beim EEDS

Zu sehen sind rechts vier verschieden farbige Behälter. Diese werden durch den Bediener in die Schleppkette oben eingesetzt. Daraufhin füllt das automatisierte Materialflusssystem diesen Behälter. Hierzu werden vier blaue und/oder rote Unterteile, mittig im Foto zu sehen, auf das Förderband gedrückt. Eine Wippe sortiert daraufhin die blauen und roten Unterteile. Die blauen werden nach links transportiert, wo sie einen blauen Deckel erhalten. Analog hierzu werden die roten Unterteile nach rechts transportiert. Nachdem die Deckel aufgesetzt wurden, werden die Behälter mit den vier Unter- und Oberteilen befüllt. Jede Behälter-Farbe steht dabei für einen anderen Mix aus blauen und roten Produkten.

Im Rahmen einer studentischen Arbeit wurde das Vorgehen zum EEDS mit diesem Versuchsaufbau erprobt. Dabei lag der Fokus auf der Verbesserung des Ablaufs der Befüllung der Behälter. Dieser ist durch eine speicherprogrammierbare Steuerung anpassbar. Der Ablauf wird dabei vorab simulativ ermittelt, indem vom Lego-Modell ein digitales Simulationsmodell abstrahiert wird.

Es zeigte sich, dass für den Einsatz an Universitäten der Ansatz mit Lego kaum die Verwendung einer Lernfabrik substituiert. Dieser Aussage liegt die Erkenntnis zugrunde, dass viel Zeit für den Aufbau und die Programmierung von Lego aufgewendet werden muss. Dies hat mit dem eigentlichen Lernziel der Einarbeitung in die Durchführung von Simulationsstudien kaum etwas zu tun und wirkt folglich eher ablenkend auf den Lernenden. Trotzdem ist festzustellen, dass kaum eine andere Möglichkeit für Lernende besteht, wenn diese keinen Zugang zu einer Lernfabrik haben.

Potenzielle Anbieter für Schulungen unter Beachtung des EEDS sind Institutionen wie Hochschulen und Industrie- oder Beratungsunternehmen, die eine wie zuvor spezifizierte Lernfabrik sowie Know-how im Bereich der ereignisdiskreten Simulation vorweisen können. Beratungsunternehmen können solch eine Aktivität für die Berater als auch für Kunden anbieten.

5.2.2 Gasturbinen

Eine Gasturbine ist eine kontinuierlich durchströmte ortsfeste Antriebsmaschine mit einer hohen Leistungsdichte [Wie-10]. Gasturbinen können mit flüssigen oder gasförmigen Brennstoffen betrieben werden und lassen sich anhand der Anzahl der Starts, Betriebsstunden pro Jahr und der abzugebenden Leistung klassifizieren [DIN-4342]. Gasturbinenkraftwerke können für die Stromerzeugung genutzt werden [Bal-10].

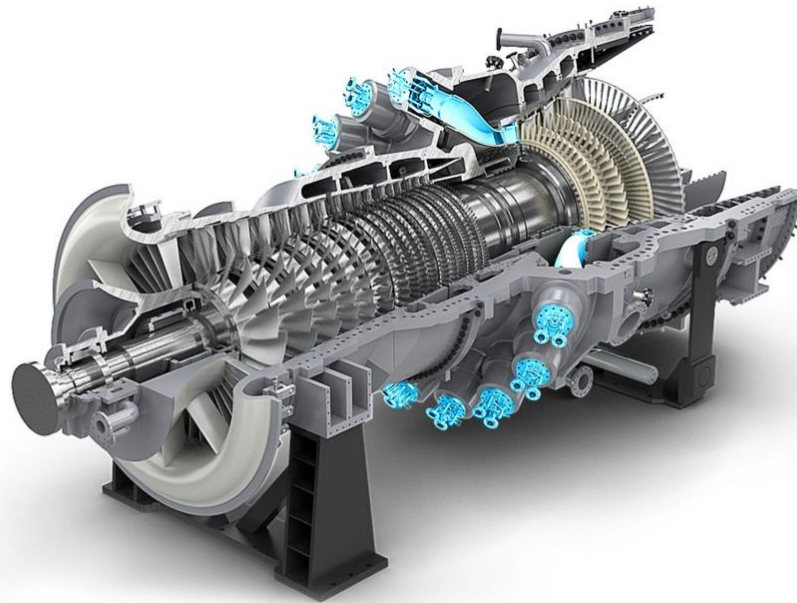
Die Montage von Gasturbinen ist in der Regel als Baustelle organisiert [Pas-10]. Bauteile und Baugruppen sind so zu disponieren, dass sie zu einem gegebenen Termin für die Montage bereitstehen. Im Folgenden ist die Fertigung der Baugruppe *Brenner* als exemplarisches Beispiel für die Investitions- und Zeitersparnis bei der erstmaligen Durchführung einer Simulationsstudie unter Nutzung von PSIMA erläutert. Die Absicherung der Planung ist das übergeordnete Ziel der Simulationsstudie.

Der Brenner einer Gasturbine besteht aus mehreren Baugruppen. In Abbildung 5.20a ist exemplarisch die Baugruppe *Support Housing* dargestellt. Ein Support Housing besteht wiederum aus sog. *Swirlern*. Durch Swirler wird die Stabilität der Verbrennung sichergestellt. Abbildung 5.20b zeigt den Brenner am Beispiel einer Gasturbine des Herstellers Siemens. Die Fertigung der Swirler erfordert die Arbeitsschritte Drehen, Erodieren, Fräsen, Massendurchsatztest, Messen, Oberflächenrissprüfung und Schweißen [Sto-17]. Mit Blick auf den Materialfluss können Arbeitsplätze mehrere Vorgänger und Nachfolger haben. Eine erneute Bearbeitung eines Werkstücks auf einem Arbeitsplatz zu einem späteren Zeitpunkt ist möglich. Dieser durch Verzweigungen und Rückflüsse bestimmte Materialfluss ist als komplex einzustufen [Löd-16, S.133f.]. Geplante und ungeplante Wartungen der Maschinen, stochastische Bearbeitungszeiten, vom Zufall beeinflusste Personalverfügbarkeiten, Schichtpläne, Reparaturen defekter



©Siemens AG

(a) Stützgehäuse (sog.
Support Housing)
[Sie-20]



©Siemens AG

(b) Gasturbine, blau markiert: Brenner [Sie-20]

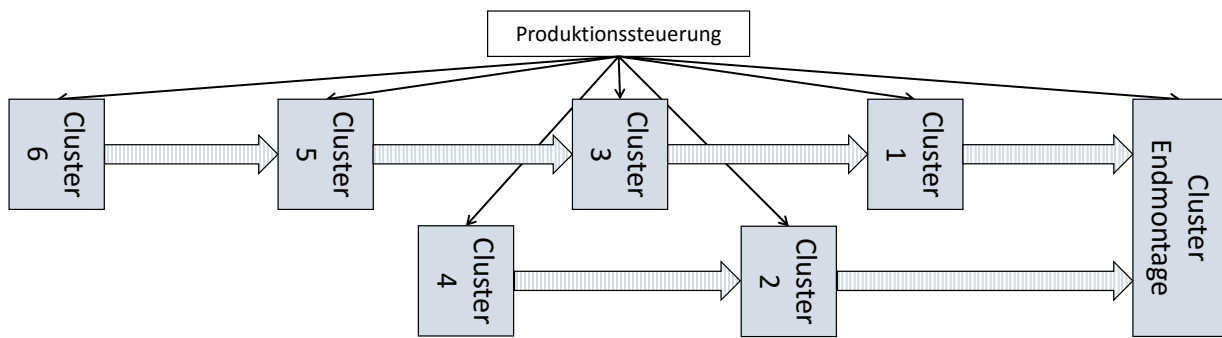
Abbildung 5.20: Anwendungsbeispiel Gasturbinenbau: Übersicht über die Produkte

Bauteile und von mehreren Produktvarianten abhängige Bearbeitungsreihenfolgen und Stücklisten erhöhen die Komplexität. Ein Mitarbeiter kann je nach Qualifikation an mehreren geeigneten Arbeitsplätzen parallel eingesetzt werden. Bei dieser sog. Mehrmaschinenbedienung rüsten weniger Mitarbeiter mehr verfügbare Maschinen. Der Mitarbeiter kehrt während der automatischen Fertigung zur Kontrolle und bei Fehlern zur Maschine zurück.

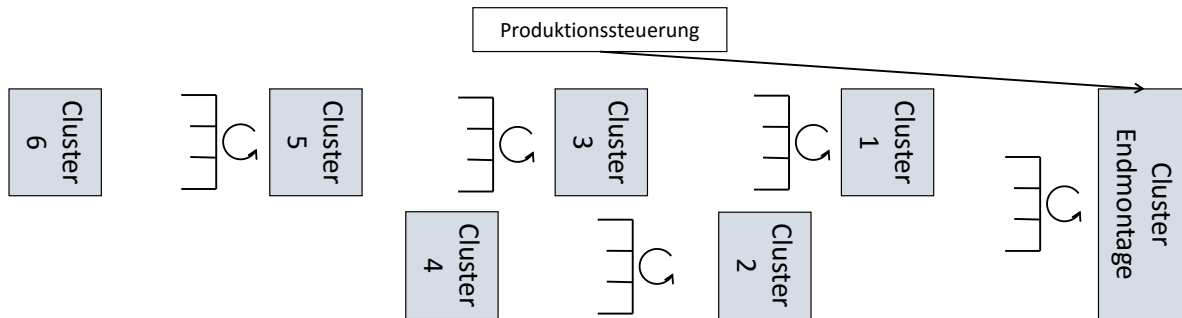
Für die Absicherung der Planung stehen die KPIs Durchlaufzeit je Produkt, Durchsatz pro Tag, Flächenverbrauch, Mitarbeiteranzahl und Mitarbeiterauslastung im Vordergrund. Die Arbeitsplätze wurden durch das Planungsteam in sogenannten Clustern zusammengefasst. Im Folgenden ist der Fokus auf den Materialfluss zwischen den Clustern gerichtet.

In einer Simulationsstudie soll die Gestaltung des Materialflusssystems überprüft werden. Konkret wird eine Push- mit einer Pull-Steuerung mit Blick auf die KPIs verglichen. Die vereinfachte Prinzipskizze in Abbildung 5.21a zeigt das Fließprinzip. Aufträge werden nach Art, Zeit und Menge durch die zentrale Produktionssteuerung an die Cluster zugewiesen. Abbildung 5.21b zeigt das Pull-Prinzip. Hierbei sind alle Cluster durch Supermärkte voneinander entkoppelt. Konkret werden Pull- von Pull-Bereichen entkoppelt. Das Cluster Endmontage bestimmt somit verbrauchsorientiert die Fertigungsmenge der vorgelagerten Cluster. Die Bestände der Supermärkte werden durch ein elektronisches Kanban bestimmt.

Der simulative Vergleich von Fließprinzip und Pull-Prinzip zeigt, dass der Durchsatz produzierter Swirler um durchschnittlich 4,5% sank [Sto-17]. Die nähere Analyse zeigt, dass bei der Pull-Steuerung die Mehrmaschinenbedienung ein starkes periodisches Verhalten hinsichtlich der Auslastung von Werkern verursacht. Im Tagesablauf zeigen sich so einerseits Phasen, in denen die Mitarbeiter mehr Maschinen bedienen müssten, als sie können. Dies resultiert in einem geringeren Durchsatz, da die Maschinen auf



(a) Fließprinzip: Aufträge werden durch zentrale Produktionssteuerung an Cluster vergeben



(b) Pull-Prinzip: Cluster sind durch Supermärkte entkoppelt

Legende:

-  – Informationsfluss
-  – Push
-  – Supermarkt
-  – Entnahme

Abbildung 5.21: Simulationsszenarien Gasturbinenbau

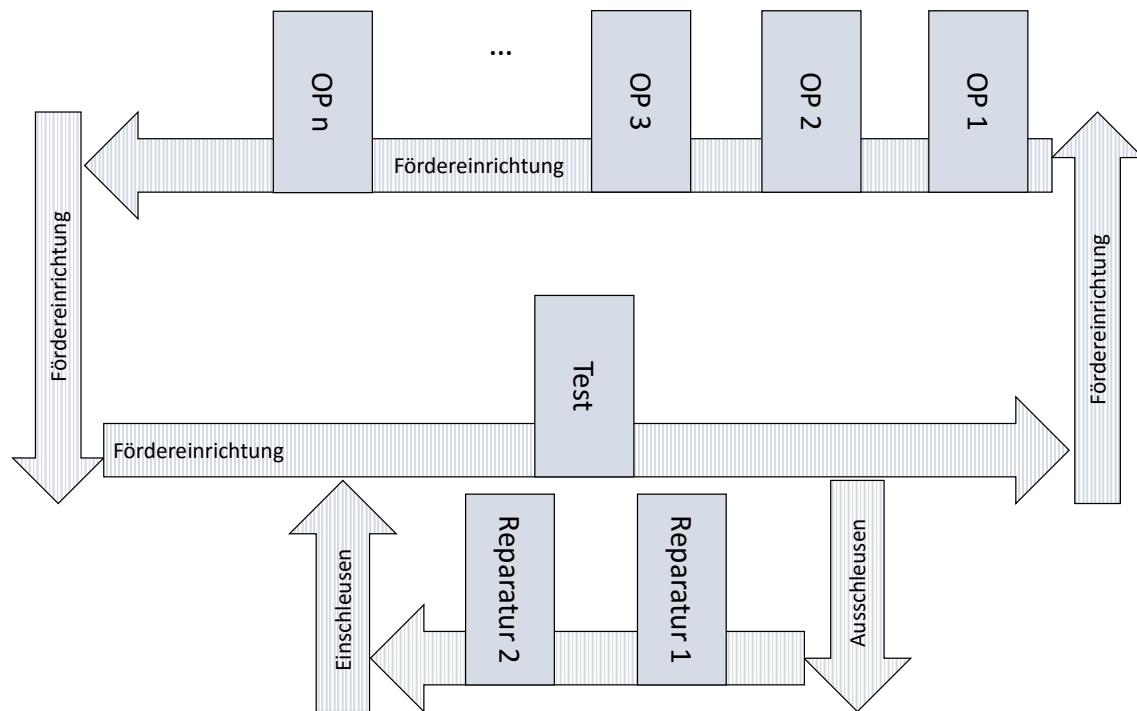
den Mitarbeiter „warten“. Andererseits treten regelmäßig Zeiträume auf, in denen die Mitarbeiter nicht durch die Maschinen erfordert werden.

Das beschriebene Beispiel wird aufgrund der Komplexität des Materialflusssystems im Simulationsinstrument modelliert und simuliert. Im Kapitel 5.2.4 ist das Potenzial für die Investitions- und Zeiterparnis durch die erhöhte Planungsqualität erörtert.

5.2.3 Motorradantriebe

Die Montage von Verbrennungsmotoren ist häufig als Linie gestaltet. Zur Absicherung von Investitionen in eine neu- oder umgeplante Montagelinie wird insbesondere in der Automobilindustrie und im Motorradbau häufig die Durchführung einer Simulationsstudie gefordert. Im Folgenden wird aufgezeigt, wie Nutzer einfache Simulationsmodelle ohne direkte Interaktion mit einem Simulationsinstrument in der mobilen App PSIMA aufbauen und Simulationsexperimente durchführen können.

Abbildung 5.22 gibt einen prinzipiellen Eindruck von einer Montagelinie für Motoren. Der Materialfluss ist als One Piece Flow gestaltet. Beim vorliegenden Beispiel handelt es sich um eine getaktete Linie. Ein Überholen ist aufgrund beengter Platzverhältnisse nicht möglich. Die zu montierenden Motoren werden auf Werkstückträgern transportiert. Die Werkstückträger zirkulieren im Sinne eines geschlossenen Kreislaufs auf der Montagelinie.



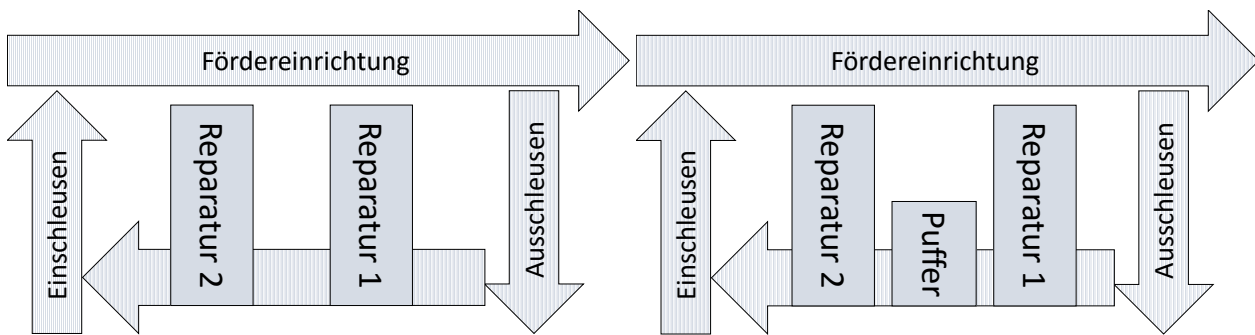
Legende:

OP – Arbeitsstation

Abbildung 5.22: Prinzipskizze Layout Motormontage

Typisch für Motormontagen sind umfangreiche Prüfungen an sogenannten Kalt- und Heißtestständen sowie Ventildichtheitsprüfungen [Kro-09, S.18]. Vorrichtungen für die Nacharbeit sind in der Regel in die Linie integriert.

Die konkrete Fragestellung an die Simulation liegt im Bereich der Reparatur. So sind in der Planung zwei Reparaturarbeitsplätze vorgesehen, die linear verkettet sind. Die vom Zufall abhängige Zeit zur Reparatur eines Motors erfordert die Unterbrechung des FIFO-Prinzips der Montagelinie. Das Ziel besteht darin, eine Steuerung für den Materialfluss dieses Reparaturbereichs zu ermitteln. Ein eventueller Rückstau auf die Montagelinie soll vermieden werden. Der Einfluss eines Puffers zwischen den beiden Reparaturarbeitsplätzen, der einen Motor fassen kann, soll mit Blick auf den Durchsatz der Linie abgeschätzt werden. Stochastische Effekte wie zufällige Häufungen defekter Motoren sowie über- oder unterdurchschnittlich lange Reparaturzeiten machen eine analytische grobe Abschätzung kaum möglich. Der Rückstau auf die Montagelinie, der sich an den Reparaturarbeitsplätzen ergeben kann, ist zufallsabhängig. Die Abbildungen 5.23a und 5.23b stellen die beiden Szenarien gegenüber.



(a) Zwei sequenzielle Reparaturarbeitsplätze

(b) Zwei sequenzielle entkoppelte Reparaturarbeitsplätze

Abbildung 5.23: Prinzipiskezze Layout Nacharbeitsbereich Motormontage

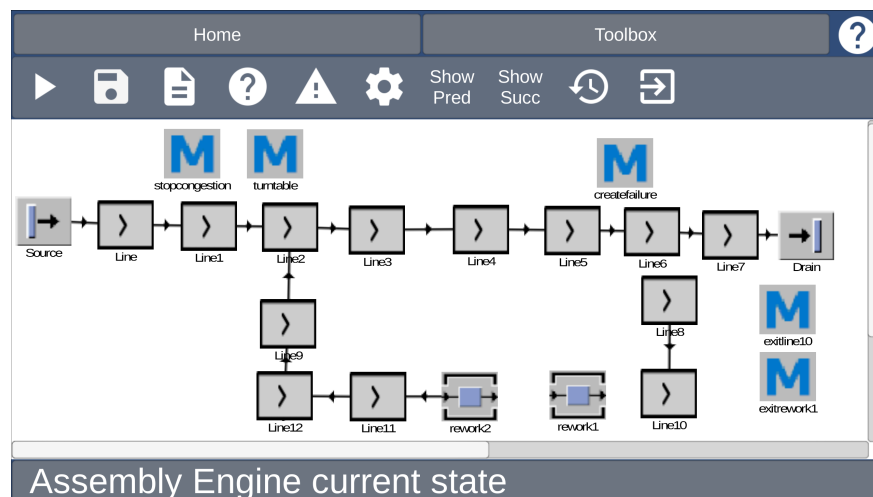
Beide Szenarien wurden im Expertenmodus der App PSIMA implementiert. Der Implementierung liegt eine Taktzeit von 2:30 Minuten zugrunde. 12% der Motoren bestehen den Test nicht und müssen repariert werden. Die Reparatur eines Motors dauert gleichverteilt zwischen 15 Minuten und einer Stunde. Beengte Platzverhältnisse machen eine sequenzielle Anordnung der beiden Reparaturarbeitsplätze erforderlich. Motoren können sich im Reparaturbereich nicht überholen.

Die Abbildungen 5.24a und 5.24b zeigen die in der App formalisierten Simulationsmodelle. Im Experiment *Assembly Engine current state* lagert ein temporäres physisches Objekt vorzugsweise auf *Reparatur 2* um. *Reparatur 1* wird nur vorgesehen, wenn *Reparatur 2* belegt ist. Möchte ein temporäres physisches Objekt, das auf *Reparatur 1* bearbeitet wurde den Reparaturarbeitsplatz verlassen, so kann es das nur, wenn auch *Reparatur 2* frei ist. Das zweite Szenario *Assembly Engine improvement buffer* zeigt einen *Puffer* zwischen *Reparatur 1* und *Reparatur 2*. Die Logik zur Steuerung ist der zuvor beschriebenen gleich, jedoch wird statt auf *Reparatur 2* auf den *Puffer* umgelagert. Zusammenfassend ist der Materialfluss in beiden Szenarien aufgrund der fehlenden Überholmöglichkeit strikt gemäß dem FIFO-Prinzip organisiert.

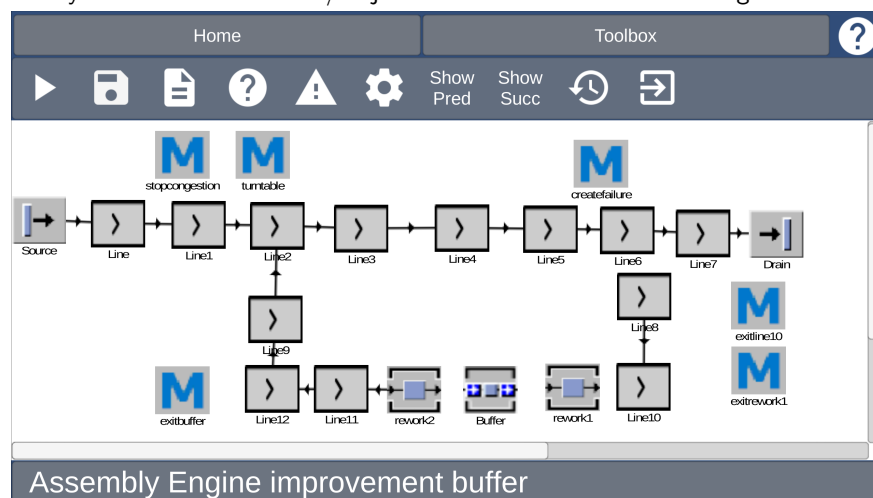
In Tabelle 5.6 ist der Durchsatz erfolgreich montierter Motoren für beide Szenarien gegenübergestellt. Den Zahlen liegen die Mittelwerte von 20 Simulationsläufen in einem Simulationszeitraum von je 100 Tagen zugrunde. Der unterschiedliche Durchsatz resultiert aus Rückstaus, die sich im Reparaturbereich bilden und die Montagelinie blockieren. Ersichtlich ist, dass durch den Puffer zwischen den beiden Reparaturarbeitsplätzen der Durchsatz um 13,9% gesteigert und die Auslastung von *Reparatur 2* um knapp 1/3 erhöht werden konnte.

Tabelle 5.6: Motormontage Ergebnisse Simulationsszenarien

	<i>Assembly Engine current state</i>	<i>Engi-Assembly Engine improvement buffer</i>	Veränderung
Durchsatz	50.497 Motoren	57.523 Motoren	+ 13,9%
<i>Reparatur 1</i> im Status <i>arbeitend</i>	99,8%	93%	- 6,8 Prozentpunkte
<i>Reparatur 2</i> im Status <i>arbeitend</i>	64,4%	94%	+ 29,6 Prozentpunkte



(a) Zwei sequenzielle Reparaturarbeitsplätze modelliert in PSIMA; vgl. Symbole von Bausteinen/Objekten mit Tabelle C.2 im Anhang C



(b) Zwei sequenzielle entkoppelte Reparaturarbeitsplätze modelliert in PSIMA; vgl. Symbole von Bausteinen/Objekten mit Tabelle C.2 im Anhang C

Abbildung 5.24: Nacharbeitsbereich Motormontage modelliert in PSIMA

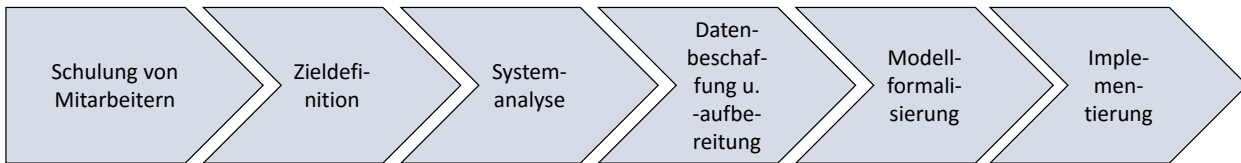
5.2.4 Planungsqualität

PSIMA trägt dazu bei, dass die ereignisdiskrete Simulation für Planer von Materialflusssystemen mit geringen Kenntnissen in der Simulationstechnik zugänglich wird. So lassen sich bei der Neu- und Umpassung von Materialflusssystemen zeitliche und finanzielle Vorteile ausschöpfen. Diese Vorteile ergeben sich aus einer höheren Qualität der Planung von Materialflusssystemen durch ereignisdiskrete Simulation [Gut-17, S.48].

Bezogen auf die beschriebenen Beispiele im Gasturbinen- und Motorenbau startet der Lernende mit Beginn des Planungsprojektes seinen Lernverlauf mit der App PSIMA. Der Lernende hat während kurzer Pausen im Arbeitsalltag die Chance, sich projektbegleitend „on the Job“ mit dem Simulationsinstrument vertraut zu machen. Nach Vollendung der Schulung führt der Lernende das Planungsprojekt mit dem

Simulationsinstrument oder im Expertenmodus von PSIMA durch. In der Abbildung 5.25 sind die Abläufe einer Simulationsstudie mit und ohne Nutzung der Lernumgebung gegenübergestellt.

Ohne Lernumgebung:



Mit Lernumgebung:

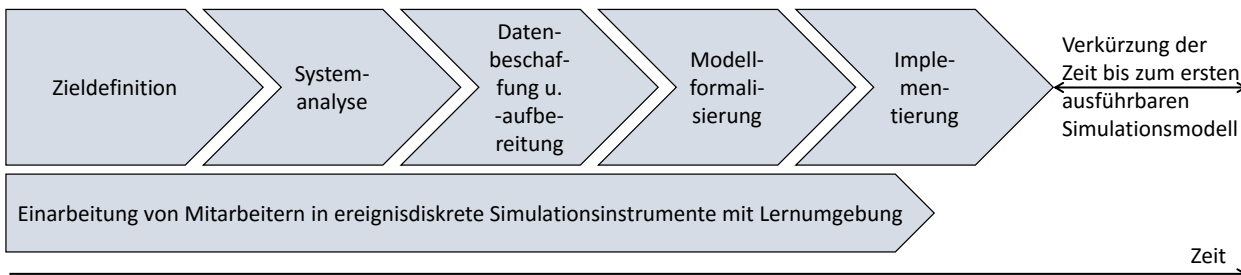


Abbildung 5.25: Gegenüberstellung Ablauf Simulationsstudie mit und ohne Lernumgebung

Eine ein- bis zweiwöchige Schulung, wie sie beispielsweise von Unternehmensberatungen angeboten werden, ist obsolet. Das Risiko einer Fehlinvestition wird verringert, da ein teures Simulationsinstrument nicht beschafft werden muss, um herauszufinden, ob damit die Fragestellung des Projekts überhaupt beantwortet werden kann. Die spielerischen und motivierenden Aspekte der Lernumgebung haben darüber hinaus das Potenzial, die Arbeitszeit effektiv und effizient zu nutzen. Durch die begleitende Einarbeitung besteht die Chance, die Projektlaufzeit zu verkürzen. Die „up-to-date“ Lernumgebung hat eine motivierende Wirkung auf den Planer von Materialflusssystemen, der den Einsatz von ereignisdiskreter Simulation im Projekt weniger als zusätzliche anstrengende Aufgabe, sondern vielmehr als sinnvoll und nützlich ansieht. Kurz zusammengefasst eignet sich der Expertenmodus der App PSIMA zur Modellierung einfacher und komplizierter Materialflusssysteme. Die Verwendung des Simulationsinstruments ist bei komplexen Materialflusssystemen - wie am Beispiel des Gasturbinenbaus gezeigt - empfehlenswert.

Die ereignisdiskrete Simulation wird durch die konzipierte Lernumgebung einfach und rasch für Anfänger in der Simulationstechnik zugänglich. Stochastische und dynamische Effekte sind somit für den Planer von Materialflusssystemen quantifizierbar. Der vermehrte Einsatz von ereignisdiskreter Simulation im Unternehmen trägt so zum kontinuierlichen Verbesserungsprozess bei.

6 Zusammenfassung

Die vorliegende Arbeit leistet einen Beitrag dazu, ereignisdiskrete Simulation zur Entscheidungsunterstützung bei der Neu- und Umplanung von Materialflusssystemen zu verbreiten, indem die Einarbeitung in ereignisdiskrete Simulationsinstrumente zur Durchführung von Simulationsstudien erleichtert wird. Spielerische Ansätze konnten als vielversprechende Ansätze identifiziert werden, da diese lern- und motivationsförderlich wirken. Die Entwicklungen richten sich an qualifizierte Hochschulabsolventinnen und -absolventen als zukünftige Akteure bei der Planung von Materialflusssystemen. Materialflusssysteme wurden im Kontext der Planung von Ganzheitlichen Produktionssystemen mit den Schwerpunkten Fließprinzip und Pull-Prinzip betrachtet [VDI-2870a], da diese in industriellen Anwendungen weitverbreitet sind. Die in der Arbeit beleuchteten Handlungsfelder sind die Einarbeitung in *ereignisdiskrete Simulationsinstrumente* und die *Durchführung von Simulationsstudien*. Die Analyse der Literatur zeigte, dass gängige flexible ereignisdiskrete Simulationsinstrumente, die im Kontext der Cyber-Physical Production System (CPPS) auch die Abbildung von Multiagentensystemen ermöglichen, sich an Simulationsexperten richten. Hinsichtlich des Handlungsfelds der Durchführung von Simulationsstudien konnte durch eine Literaturanalyse bestätigt werden, dass die Abstraktion beim Aufbau von Simulationsmodellen und die Interpretation von Simulationsergebnisdaten Erfahrungen des Durchführenden von Simulationsstudien erfordern. Erfahrungen, die durch die praktische Durchführung von Simulationsstudien gesammelt werden, sind Voraussetzung für Verbesserungen von Materialflusssystemen, die in der Realität wie durch Simulation ermittelt, wirken.

Spielerische Ansätze für handelsübliche mobile Endgeräte können orts- und zeitunabhängig genutzt werden. Digitale Spiele versprechen zudem eine hohe Lehr- und Lernproduktivität. Spiele erhöhen die studentische Motivation, beeinflussen Gruppenarbeiten positiv und ermöglichen ein Experimentieren in einer sicheren Umgebung. Belohnungen wie Punkte oder Abzeichen werden in den Spielen unter Beachtung eines Regelwerks an Spieler vergeben. Simulationsinstrumente, die Bausteinkästen und einen grafisch-interaktiven Modellaufbau sowie eine Programmiersprache zur Implementierung von Simulationsmodellen unterstützen, konnten als marktüblich für die Neu- und Umplanung von Materialflusssystemen identifiziert werden. Spielerische Ansätze zur Einarbeitung in diese Simulationsinstrumente existierten bislang nicht. Hinsichtlich der Einarbeitung in die Durchführung von Simulationsstudien konnten in der Literatur weder Ansätze mit Fokus auf den Aufbau von Erfahrungen noch Vorarbeiten unter Verwendung spielerischer Elemente identifiziert werden.

Eine Lernumgebung wurde auf Basis von 17 Anforderungen, die durch eine Literaturanalyse ermittelt wurden, konzipiert. Die Lernumgebung ist im Sinne eines Spiel- und Lernverlaufs in der Arbeit beschrieben. Die Lernumgebung vereint Entwicklungen zur Einarbeitung des Lernenden in ereignis-

diskrete Simulationsinstrumente und die Durchführung von Simulationsstudien. Der Lernende wird in den grafisch-interaktiven Modellaufbau unter Nutzung von Bausteinen/Objekten und in eine Programmiersprache zur Implementierung von Simulationsmodellen durch eine spielerisch gestaltete App eingearbeitet. Konkret kommen hierzu sogenannte inkrementell komplizierter werdende Simulationsmodelle (IKS) und visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen (VBPIS) zum Einsatz.

Bei IKS handelt es sich um kurze, abgeschlossene Lerneinheiten, die induktiv aufeinander aufbauen. In schrittweise komplizierter werdenden Levels werden an den Lernenden immer neue Aufgaben gestellt. Zur Lösung der Aufgaben muss der Lernende bislang unbekannte Objekte und Bausteine nutzen. Objekte und Bausteine bezeichnen Modellelemente ohne und mit interner Ablauflogik [VDI-3633b, S.12].

Mit steigender Herausforderung muss auch eine Programmiersprache zur Implementierung von Simulationsmodellen durch den Lernenden verwendet werden. Durch die VBPIS wird den Lernenden die Syntax und Semantik der beispielhaft gewählten objektorientierten Programmiersprache zur Implementierung von Simulationsmodellen *SimTalk* des Simulationsinstruments *Tecnomatix Plant Simulation* vermittelt. Die VBPIS vereint die grafischen Eigenschaften der visuellen blockbasierten Programmiersprache mit Funktionalitäten von Programmiersprachen zur Implementierung von Simulationsmodellen.

In einer App, die die IKS und VBPIS vereint, sind Elemente des Spielens wie Punkte und Abzeichen integriert. Diese sollen den Lernenden zur häufigen Interaktion motivieren. Spielelemente wirken sich positiv auf die Produktivität des Lernens und Lehrens aus. Für die Interaktion des Benutzers mit der App und für korrekte Lösungen werden Punkte vergeben, die in einem sogenannten Leaderboard für alle Lernenden sichtbar sind. Ein Leaderboard stellt den Rang eines Spielers in einer Gruppe von Spielern dar. Die Oberfläche für die Modellierung in der App ist ähnlich der des Simulationsinstruments gestaltet.

Die App ist als Teil einer Cloud-IT-Architektur konzipiert worden. Diese Architektur ermöglicht eine einfache Wartung und nutzt etwaige Lizenzen effektiv. Der Aufbau besteht aus der mobilen App, einem Webserver, einer Datenbank und einem Simulationsinstrument. Die Cloud-IT-Architektur ist skalierbar im Sinne der Verwendung mehrerer mobiler Endgeräte und Simulationsinstrumente. Die App dient als Benutzungsoberfläche zur Datenein- und -ausgabe. Die in der App aufgebauten Simulationsmodelle werden so kodiert, dass sie durch das Web übermittelt und in einer Datenbank gespeichert werden können. Im Simulationsinstrument wird dann das Simulationsmodell (teil-)automatisch generiert, simuliert und die Simulationsergebnisdaten in die Datenbank übertragen. Schließlich kann der Lernende sich die Simulationsergebnisdaten in der App visualisieren lassen.

Sobald der Benutzer Objekte und Bausteine anwenden kann, findet eine praxisnahe Einarbeitung in die Durchführung von Simulationsstudien statt. Hier kommt das Vorgehen zum erfahrungsbasierten Erlernen der Durchführung von Simulationsstudien (EEDS) zum Einsatz. Ausgangspunkt ist eine Lernfabrik, die ein Materialflusssystem abbildet. Lernfabriken, die Lerninhalte zum Fließprinzip und Pull-Prinzip wie das U-Layout oder Kanban abbilden, stehen im Fokus. Lernende betreiben diese Fabrik und nehmen Daten wie Durchlaufzeiten auf. Diese dienen als Grundlage für die Erzeugung eines Simulationsmodells, mit dem sich Simulationsexperimente durchführen lassen. Lernende ermitteln an diesem Modell Maßnahmen

zur Verbesserung des Materialflusses. Diese werden in die Lernfabrik implementiert. Durch den erneuten Betrieb der Lernfabrik erhält der Lernende folglich eine Rückmeldung hinsichtlich der Auswirkungen der simulativ ermittelten Verbesserungen auf den Betrieb der Lernfabrik.

Die Cloud-IT-Architektur wurde errichtet und die App Production Simulation Application (PSIMA) inklusive IKS und VBPIs prototypisch am Beispiel des weitverbreiteten Simulationsinstruments *Tecnomatix Plant Simulation* umgesetzt [Mül-17a]. In einer Erprobung wurde PSIMA von Studierenden hinsichtlich der Motivation, des Nutzungsverhaltens und des Lernerfolgs über 19 Tage getestet. Das Motivationsempfinden wurde durch zwei Drittel der Probanden bestätigt. Ein Proband war nach der Nutzung von PSIMA in der Lage, Simulationsmodelle direkt in dem Simulationsinstrument aufzubauen und mit geringer Hilfe zu simulieren. Von 19 aktiven Spielern wurden 377 Modelle eingereicht. Aktive Spieler bezeichnet Studierende, die mindestens ein Modell einreichten. Diese hohe Anzahl eingereichter Modelle zeigt die große Motivation, PSIMA zu nutzen und viele Punkte zu erreichen. Die Auswertung der Nutzungsdaten zeigt, dass PSIMA überwiegend zur Vor- und Nachbereitung einer begleitenden Lehrveranstaltung zur ereignisdiskreten Simulation durch die Studierenden genutzt wurde.

Im Rahmen einer materialflusstechnischen Anwendung im universitären Versuchsfeld wurde das Vorgehen zum EEDS in einer Lehrveranstaltung mit Studierenden der Vietnamesisch-Deutschen Universität erprobt [Mül-17b]. Die Studierenden konnten erfahren, dass eine sorgfältige Vorbereitung, Durchführung und Interpretation von Simulationsstudien Voraussetzungen für zufriedenstellende Ergebnisse sind. Sie wurden aufgefordert, in Kleingruppen unter gegebenem Zielsystem simulativ ermittelte Verbesserungsmaßnahmen für den Materialfluss einer Lernfabrik vorzuschlagen. In der Lernfabrik war initial ein Engpass vorhanden, den es zu erkennen und beheben galt. Nach der Umsetzung der simulativ ermittelten Verbesserungen in der Lernfabrik, wie der Einführung von Supermärkten zur Verschiebung des Kundenentkopplungspunkts, wurde die vermeintlich verbesserte Montage durch die Studierenden betrieben. Die meisten Studierendengruppen konnten die simulativ ermittelte verbesserte Kundenzufriedenheit mit einem erneuten Betrieb der Lernfabrik nicht reproduzieren. Sie abstrahierten die Realität beim Aufbau ihres Simulationsmodells zu sehr, z.B. Verfügbarkeit von Betriebsmitteln = 100%. Das Potenzial zur Investitions- und Zeitersparnis bei der Durchführung von Simulationsstudien durch Nutzung der Lernumgebung im industriellen Einsatz wurde in der Arbeit an zwei weiterführenden Anwendungsbeispielen im Gasturbinenbau und der Motormontage illustriert. Eine push- wurde einer pull-gesteuerten Fertigung von Brennern von Gasturbinen gegenübergestellt. Es wurde aufgezeigt, dass die Lernumgebung im projektbegleitenden Einsatz „Training on the Job“ unterstützt. Am Beispiel eines Nacharbeitsbereichs in der Motormontage wurde aufgezeigt, dass sich ein praktisches Problem in PSIMA lösen lässt. In PSIMA wurde eine Sensitivitätsanalyse vollzogen. Das Potenzial zur Abbildung einfacher und komplizierter Materialflusssysteme in PSIMA wurde an diesem Beispiel erörtert.

Die Entwicklungen in dieser Arbeit tragen dazu bei, dass Maßnahmen zur Analyse und Verbesserung des Materialflusses, deren Auswirkungen sich aufgrund dynamischer und stochastischer Effekte mit analytischen Methoden nur schwer abschätzen lassen, durch die Lernumgebung für Anfänger in der Simulationstechnik quantifizierbar werden.

Literaturverzeichnis

- [Abd-13] Abdul Rahman, A. A.; *Approach for Integrating Predictive-Reactive Job Shop Scheduling with PLC-Controlled Material Flow*; Dissertation; Technische Universität Berlin; Berlin, 2013.
- [Abe-12] Abele, E.; Bechtloff, S.; Cachay, J.; Tenberg, R.; *Lernfabriken einer neuen Generation: Entwicklung einer Systematik zur effizienten Gestaltung von Lernfabriken*; in *Zeitschrift für wirtschaftlichen Fabrikbetrieb*; 107(3), 2012:147–151.
- [Abe-14] Abele, E.; *Learning Factory - Living reference work entry - 07.03.2016*; in Laperrière, L.; Reinhart, G. (Hrsg.), *CIRP Encyclopedia of Production Engineering*; Springer, Heidelberg, New York, Dordrecht, London, 2014; 1–5, Ergänzung online 2016.
- [Abe-15] Abele, E.; Metternich, J.; Tisch, M.; Chryssolouris, G.; Sihn, W.; ElMaraghy, H.; Hummel, V.; Ranz, F.; *Learning Factories for Research, Education, and Training*; in *Procedia CIRP, the 5th Conference on Learning Factories 2015*; 32, 2015:1–6.
- [Abe-17] Abele, E.; Chryssolouris, G.; Sihn, W.; Metternich, J.; ElMaraghy, H.; Seliger, G.; Sivar, G.; ElMaraghy, W.; Hummel, V.; Tisch, M.; Seifermann, S.; *Learning Factories for future oriented research and education in manufacturing*; in *CIRP Annals - Manufacturing Technology*; 66, 2017:803–826.
- [Abt-70] Abt, C. C.; *Serious games*; The Viking Press, New York, 1970.
- [Ada-11] Adair, A.; *Siemens launches Plantville: Press Release Siemens Canada Limited*; <http://www.siemens.ca/web/portal/en/Press-Archive/2012/Documents/Plantville%20Canada%20Press%20Release-FINAL.pdf>; letzte Aktualisierung am 04.07.2011; zuletzt geprüft am 10.11.2017, 2011.
- [Alv-19] Alvarez, J.; Djaouti, D.; Rampnoux, O.; *Serious Game Classification: A Collaborative Classification of Serious Games*; <http://serious.gameclassification.com>; zuletzt geprüft am 04.03.2019, 2019.
- [And-01] Anderson, L. W.; Krathwohl, D. R.; Airasian, P. W.; Cruikshank, K. A.; Mayer, R. E.; Pintrich, P. R.; Raths, J.; Wittrock, M. C.; *A taxonomy for learning, teaching, and assessing: A revision of Bloom's Taxonomy of Educational Objectives*; Longman, New York, 2001.
- [Apa-19] Apache; *Apache: HTTP Server Project*; <https://httpd.apache.org>; zuletzt geprüft am 04.03.2019, 2019.

- [ASI-19] ASIM; *ASIM - Arbeitsgemeinschaft Simulation: Fachgruppen: Edukation und Simulation*; <https://www.asim-gi.org/fachgruppen/edukation-und-simulation/>; zuletzt geprüft am 05.03.2019, 2019.
- [Bäc-08] Bäck, A.; *Spielerische Optimierung der Logistik*; in Engelhardt-Nowitzki, C.; Nowitzki, O.; Krenn, B. (Hrsg.), *Praktische Anwendung der Simulation im Materialflussmanagement: Erfolgsfaktoren und Implementierungsszenarien*; Gabler Edition Wissenschaft. Leobener Logistik Cases; Gabler, Wiesbaden, 2008; 35–48.
- [Bal-10] Balling, L.; *GT-Kraftwerke*; in Lechner, C.; Seume, J. (Hrsg.), *Stationäre Gasturbinen*; VDI-Buch; Springer, Berlin, 2010; 49–99.
- [Ban-05] Banks, J.; Carson, J. S.; Nelson, B. L.; Nicol, D. M.; *Discrete-Event System Simulation*; Prentice-Hall international series in industrial and systems engineering; Pearson Prentice Hall, Upper Saddle River; 4th Edn., 2005.
- [Ban-08] Bangsow, S.; *Fertigungssimulationen mit Plant Simulation und SimTalk: Anwendung und Programmierung mit Beispielen und Lösungen*; Edition Cad.de; Hanser, München, 2008.
- [Ban-10a] Bangsow, S.; *Manufacturing Simulation with Plant Simulation and SimTalk: Usage and Programming with Examples and Solutions*; Springer, Berlin, Heidelberg, 2010.
- [Ban-10b] Banks, J.; Carson, J. S.; Nelson, B. L.; Nicol, D. M.; *Discrete-Event System Simulation*; Pearson, Upper Saddle River; fifth Edn., 2010.
- [Ban-11] Bangsow, S.; *Praxishandbuch Plant Simulation und SimTalk: Anwendung und Programmierung in über 150 Beispiel-Modellen*; Hanser, München, 2011.
- [Ban-15] Bangsow, S.; *Tecnomatix Plant Simulation: Modeling and Programming by Means of Examples*; Springer, Heidelberg, New York, Dordrecht, London, 2015.
- [Bar-96] Bartle, R.; *Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs*; in *Journal of MUD research*; 1(1), 1996.
- [Bar-13a] Barata, G.; Gama, S.; Jorge, J.; Goncalves, D.; *Engaging Engineering Students with Gamification: An empirical study*; in *5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2013:1–8.
- [Bar-13b] Barata, G.; Gama, S.; Jorge, J.; Goncalves, D.; *Improving participation and learning with gamification*; in *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, 2013:10–17.
- [Ber-14] Bergmann, S.; *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*; Dissertation; Technische Universität Ilmenau; Ilmenau, 2014.
- [Bie-13] Bielajew, A. F.; *History of Monte Carlo*; in Seco, J.; Verhaegen, F. (Hrsg.), *Monte Carlo Techniques in Radiation Therapy*; Taylor & Francis, Boca Raton, 2013; 3–16.

- [Bis-15] Bischoff, J.; Taphorn, C.; Wolter, D.; Braun, N.; Fellbau, A.; Goloverov, S. L.; Hegemann, T.; Prasse, C.; Henke, M.; ten Hompel, M.; Döbbeler, F.; Fuss, E.; Kirsch, C.; Mättig, B.; Braun, S.; Guth, M.; Kaspars, M.; Scheffler, D.; *Erschließung der Potenziale der Anwendung von Industrie 4.0 im Mittelstand: Studie im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi)*; agiplan GmbH, 2015.
- [Bit-17] Bitkom Research; *Anteil der Smartphone-Nutzer in Deutschland nach Altersgruppe im Jahr 2017*; <https://de.statista.com/statistik/daten/studie/459963/umfrage/anteil-der-smartphone-nutzer-in-deutschland-nach-altersgruppe/> ; letzte Aktualisierung im August 2017; zuletzt geprüft am 23.02.2019, 2017.
- [Blo-84] Bloom, B. S.; Krathwohl, D. R.; Masia, B. B.; *Taxonomy of educational objectives: Book 2 Affective Domain*; Longman, New York, 1984.
- [Boe-19] Boer, P. v. d.; *White Paper: Introduction to Gamification*; <https://www.cdu.edu.au/olt/ltresources/downloads/whitepaper-introductiontogamification-130726103056-phpapp02.pdf> ; zuletzt geprüft am 05.03.2019, 2019.
- [Böh-11] Böhme, S.; Heinemann, T.; Herrmann, C.; Mennenga, M.; Nohr, R.; Othmer, J.; *Business Game for Total Life Cycle Management*; in Hesselbach, J.; Herrmann, C. (Hrsg.), *Glocalized Solutions for Sustainability in Manufacturing: Proceedings of the 18th CIRP International Conference on Life Cycle Engineering*, Technische Universität Braunschweig; Springer, Berlin, Heidelberg, 2011; 531–536.
- [Bra-75] Branson, R. K.; Rayner, G. T.; Cox, L. J.; Furman, J. P.; King, F. J.; Hannum, W. H.; *Interservice Procedure for Instructional Systems Development: Executive Summary, Phase I, Phase II, Phase III, Phase IV, and Phase V*; Florida State University, Florida, 1975.
- [Bra-11] Bracht, U.; Geckler, D.; Wenzel, S.; *Digitale Fabrik: Methoden und Praxisbeispiele*; VDI-Buch; Springer, Berlin, Heidelberg, 2011.
- [Bra-18] Bracht, U.; Geckler, D.; Wenzel, S.; *Digitale Fabrik: Methoden und Praxisbeispiele*; VDI-Buch; Springer Vieweg, Berlin; 2., aktualisierte und erweiterte Edn., 2018.
- [Byr-10] Byrne, J.; Heavey, C.; Byrne, P. J.; *A review of Web-based simulation and supporting tools*; in *Simulation Modelling Practice and Theory*; 18(3), 2010:253–276.
- [Byr-15] Byrne, J.; Liston, P.; e Ferreira, D. C.; Byrne, P. J.; *Cloud based data capture and representation for Simulation in Small and Medium Enterprises*; in Yilmaz, L.; Chan, W.; Moon, I.; Roeder, T.; Macal, C.; Rossetti, M. (Hrsg.), *Proceedings of the 2015 Winter Simulation Conference: Social and Behavioral Simulations*; IEEE, 2015; 2195–2206.
- [Cag-14] Caggiano, A.; *Manufacturing Systems*; in Laperrière, L.; Reinhart, G. (Hrsg.), *CIRP Encyclopedia of Production Engineering*; Springer, Heidelberg, New York, Dordrecht, London, 2014; 830–836.
- [Cai-01] Caillois, R.; Barash, M.; *Man, play, and games*; University of Illinois Press, Urbana, 2001.

- [Chi-10] Chisu, R.; Kuzmany, F.; Günthner, W.; *Realisierung einer agentenbasierten Steuerung für Elektrohängebahnsysteme*; in Günthner, W.; ten Hompel, M. (Hrsg.), *Internet der Dinge in der Intralogistik*; VDI-Buch; Springer, Berlin, Heidelberg, 2010; 263–274.
- [Col-09] Coller, B. D.; Shernoff, D. J.; *Video Game-Based Education in Mechanical Engineering: A Look at Student Engagement*; in *International Journal of Engineering Education*; 25(2), 2009:308–317.
- [Com-03] Computerlexikon.com; *Instanzen*; <https://www.computerlexikon.com/was-ist-instanzen> ; letzte Aktualisierung am 08.12.2003; zuletzt geprüft am 19.09.2019, 2003.
- [Com-11] Computerwoche; *"Plantville". Siemens wandert auf Zyngas Spuren: Ganz im Stil der erfolgreichen "Farmville" oder "Cityville"-Reihe von Zynga will Siemens Industry mit einem eigenen Spiel Anlagenmanager ansprechen*; <https://www.computerwoche.de/a/siemens-wandert-auf-zyngas-spuren,2490808> ; letzte Aktualisierung am 18.07.2011; zuletzt geprüft am 05.03.2019, 2011.
- [Csi-14a] Csikszentmihalyi, M.; *The Concept of Flow*; in Csikszentmihalyi, M. (Hrsg.), *Flow and the Foundations of Positive Psychology: The Collected Works of Mihaly Csikszentmihalyi*; Springer, Dordrecht, Heidelberg, New York, London, 2014; 239–263.
- [Csi-14b] Csikszentmihalyi, M. (Hrsg.); *Flow and the Foundations of Positive Psychology: The Collected Works of Mihaly Csikszentmihalyi*; Springer, Dordrecht, Heidelberg, New York, London, 2014.
- [Del-16] Delago, L. C.; Machado, M. E. F. H. S.; Brito, F. O. d. B.; Landgraf, G. C.; Schroeder, M. d. A.; Torezzan, C.; *Learning Lean philosophy through 3D game-based simulation*; in Roeder, T. M.; Frazier, P. I.; Szechtman, R.; Zhou, E.; Huschka, T.; Chick, S. E. (Hrsg.), *Proceedings of the 2016 Winter Simulation Conference: Simulating complex service systems*; IEEE, 2016; 3385–3392.
- [Det-11] Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L.; *From Game Design Elements to Gamefulness: Defining "Gamification"*; in Lugmayr, A.; Franssila, H.; Hammouda, I.; Safran, C. (Hrsg.), *Proceedings of the 15th International Academic MindTrek Conference Envisioning Future Media Environments*; ACM, 2011; 9–15.
- [DFG-19] DFG - Deutsche Forschungsgemeinschaft; *Projekt SFB 637: Selbststeuerung logistischer Prozesse - Ein Paradigmenwechsel und seine Grenzen*; <http://gepris.dfg.de/gepris/projekt/5485934> ; zuletzt geprüft am 13.01.2019, 2019.
- [Dia-07] Dias, L. M.; Pereira, G. A.; Rodrigues, A.; *A Shortlist of the Most 'Popular' Discrete Simulation Tools*; in *SNE Simulation Notes Europe*; 17(1), 2007:33–36.
- [Dia-16] Dias, L. M.; Vieira, A. A. C.; Pereira, G. A. B.; Oliveira, J. A.; *Discrete simulation software ranking — A top list of the worldwide most popular and used tools*; in Roeder, T. M.; Frazier, P. I.; Szechtman, R.; Zhou, E.; Huschka, T.; Chick, S. E. (Hrsg.), *Proceedings of*

the 2016 Winter Simulation Conference: Simulating complex service systems; IEEE, 2016; 1060–1071.

- [Din-14] Dini, G.; Spath, D.; *Material Flow*; in Laperrière, L.; Reinhart, G. (Hrsg.), *CIRP Encyclopedia of Production Engineering*; Springer, Heidelberg, New York, Dordrecht, London, 2014; 844–851.
- [DIN-4342] *Deutsches Institut für Normung DIN 4342: Gasturbinen: Normbezugsbedingungen, Normleistungen, Angaben über Betriebswerte*; DIN 4342; Beuth Verlag, Berlin, Dezember 1979.
- [DIN-66001] *Deutsches Institut für Normung DIN 66 001: Sinnbilder und ihre Anwendung: Informationsverarbeitung*; DIN 66001; Beuth Verlag, Berlin, Dezember 1983.
- [DIN-66241] *Deutsches Institut für Normung DIN 66 241: Informationsverarbeitung: Entscheidungstabelle: Beschreibungsmittel*; DIN 66241; Beuth Verlag, Berlin, Januar 1979.
- [DIN-66261] *Deutsches Institut für Normung DIN 66 261: Sinnbilder für Struktogramme nach Nassi-Shneiderman*; DIN 66261; Beuth Verlag, Berlin, November 1985.
- [DIN-8580] *Deutsches Institut für Normung DIN 8580: Fertigungsverfahren: Begriffe, Einteilung*; DIN 8580; Beuth Verlag, Berlin, September 2003.
- [Dja-11] Djaouti, D.; Alvarez, J.; Jessel, J.-P.; Rampnoux, O.; *Origins of Serious Games*; in Ma, M.; Oikonomou, A.; Jain, C. L. (Hrsg.), *Serious Games and Edutainment Applications*; Springer, London, 2011; 25–43.
- [Dom-15] Dombrowski, U.; Mielke, T. (Hrsg.); *Ganzheitliche Produktionssysteme: Aktueller Stand und zukünftige Entwicklungen*; VDI-Buch; Springer Vieweg, Berlin, 2015.
- [Dro-15] Droste, M.; Hasselmann, V.-R.; Lenzian, H.; *Fließprinzip*; in Dombrowski, U.; Mielke, T. (Hrsg.), *Ganzheitliche Produktionssysteme: Aktueller Stand und zukünftige Entwicklungen*; VDI-Buch; Springer Vieweg, Berlin, 2015; 96–109.
- [Eck-02] Eckardt, F.; *Ein Beitrag zu Theorie und Praxis datengetriebener Modellgeneratoren zur Simulation von Produktionssystemen*; Dissertation; Technische Universität Ilmenau; Ilmenau, 2002.
- [Ele-12] Eley, M.; *Simulation in der Logistik: Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges "Plant Simulation"*; Springer-Lehrbuch; Springer, Berlin, Heidelberg, 2012.
- [EIM-12] ElMaraghy, W.; ElMaraghy, H.; Tomiyama, T.; Monostori, L.; *Complexity in engineering design and manufacturing*; in *CIRP Annals - Manufacturing Technology*; 61(2), 2012:793–814.
- [Eng-14] Engel, M.; Nehls, O.; *Personensimulation mit Plant Simulation am Beispiel "Evakuierungssimulation des Bahnhofs Stuttgart 21"*; <https://www.youtube.com/watch?v=AJp3jq9Od68>; letzte Aktualisierung am 05.02.2014; zuletzt geprüft am 17.01.2019, 2014.

- [Erl-10] Erlach, K.; *Wertstromdesign: Der Weg zur schlanken Fabrik*; Springer, Berlin, Heidelberg; 2., bearbeitete und erweiterte Edn., 2010.
- [Fec-19] Fechteler, T.; *SimPlan: SimChain: Supply Chain Simulation as a Service*; https://www.simchain.net/wp-content/uploads/SimChain_Presentation_Website.pdf ; zuletzt geprüft am 05.03.2019, 2019.
- [Fol-19] Folkmann, F.; *PapDesigner (Software)*; <https://www.heise.de/download/developer/friedrich-folkmann> ; zuletzt geprüft am 05.03.2019, 2019.
- [Fra-13] Franke, H.; *Multiagentensimulation in komplexen Logistiksystemen am Beispiel der Baustellenlogistik*; in Dangelmaier, W.; Laroque, C.; Klaas, A. (Hrsg.), *Simulation in Produktion und Logistik 2013: Entscheidungsunterstützung von der Planung bis zur Steuerung. 15. ASIM Fachtagung*; ASIM-Mitteilung; W.V. Westfalia Druck GmbH, Paderborn, 2013; 559–568.
- [Fre-04] Freitag, M.; Herzog, O.; Scholz-Reiter, B.; *Selbststeuerung logistischer Prozesse: Ein Paradigmenwechsel und seine Grenzen*; in *Industrie Management*; 20(1), 2004:23–27.
- [Gag-79] Gagné, R. M.; Briggs, L. J.; *Principles of Instructional Design*; Holt, Rinehart and Winston, New York, Chicago, San Francisco, Dallas, Montreal, Toronto, London, Sydney; second Edn., 1979.
- [gam-18] game – Verband der deutschen Games-Branche e.V.; *Deutscher Markt für digitale Spiele 2018*; <https://www.game.de/marktdaten/deutscher-games-markt-2018/> ; letzte Aktualisierung in 2018; zuletzt geprüft am 23.02.2019, 2018.
- [Geh-10a] Gehlich, D.; Luft, A.; Libert, S.; *Ein dezentral gesteuertes Kommissionierlager*; in Günthner, W.; ten Hompel, M. (Hrsg.), *Internet der Dinge in der Intralogistik*; VDI-Buch; Springer, Berlin, Heidelberg, 2010; 295–312.
- [Geh-10b] Gehrke, J. D.; Herzog, O.; Langer, H.; Malaka, R.; Porzel, R.; Warden, T.; *An Agent-based Approach to Autonomous Logistic Processes: Collaborative Research Centre 637: Autonomous Cooperating Logistic Processes*; in *KI - Künstliche Intelligenz*; 24(2), 2010:137–141.
- [Gei-17] Geissbauer, R.; Schrauf, S.; Bertram, P.; Cheraghi, F.; *Digital Factories 2020: Shaping the future of manufacturing*; PricewaterhouseCoopers GmbH Wirtschaftsprüfungsgesellschaft, PwC online, 2017.
- [Goe-11] Goertz, L.; *Einsatzmöglichkeiten für Serious Games in Unternehmen: Spielerisch lernen und Zusammenhänge erkunden*; in *Personalführung*; 2, 2011:58–65.
- [Göh-10] Göhring, S.; Lorenz, T.; *Agentenbasierte Staplerleitsysteme*; in Günthner, W.; ten Hompel, M. (Hrsg.), *Internet der Dinge in der Intralogistik*; VDI-Buch; Springer, Berlin, Heidelberg, 2010; 313–328.
- [Goo-19a] Google; *Firebase Realtime Database*; <https://firebase.google.com/docs/database/> ; letzte Aktualisierung am 24.01.2019; zuletzt geprüft am 05.03.2019, 2019.

- [Goo-19b] Google; *Google for Education - Blockly: Try Blockly*; <https://developers.google.com/blockly>; zuletzt geprüft am 05.03.2019, 2019.
- [Gre-09] Gregory, J.; *Game Engine Architecture*; Taylor & Francis, Boca Raton, 2009.
- [Grö-13] Gröger, C.; Silcher, S.; Westkämper, E.; Mitschang, B.; *Leveraging Apps in Manufacturing. A Framework for App Technology in the Enterprise*; in *Procedia CIRP, Forty Sixth CIRP Conference on Manufacturing Systems 2013*; 7, 2013:664–669.
- [Gru-13] Grundig, C.-G.; *Fabrikplanung: Planungssystematik, Methoden, Anwendungen*; Carl Hanser Verlag, München; 4., aktualisierte Edn., 2013.
- [Gru-15] Grundig, C.-G.; *Fabrikplanung: Planungssystematik - Methoden - Anwendungen*; Carl Hanser Verlag, München; 5., aktualisierte Edn., 2015.
- [Gün-15] Günthner, W. A.; Mandl, H.; Klevers, M.; Sailer, M.; *Forschungsbericht zu dem IGF-Vorhaben GameLog - Gamification in der Intralogistik*; fml - Lehrstuhl für Fördertechnik Materialfluss Logistik, Technische Universität München, Lehrstuhl für Fördertechnik Materialfluß Logistik (fml), Garching, 2015.
- [Gut-04] Gutenschwager, K.; Alicke, K.; *Supply Chain Simulation mit ICON-SimChain*; in Spengler, T.; Voß, S.; Kopfer, H. (Hrsg.), *Logistik Management: Prozesse, Systeme, Ausbildung*; Physica Paperback; Springer, Berlin, Heidelberg, 2004; 161–178.
- [Gut-17] Gutenschwager, K.; Rabe, M.; Spieckermann, S.; Wenzel, S.; *Simulation in Produktion und Logistik: Grundlagen und Anwendungen*; Springer, Berlin, Heidelberg, 2017.
- [Har-02] Harper, P. R.; *A Framework for Operational Modelling of Hospital Resources*; in *Health Care Management Science*; 5(3), 2002:165–173.
- [Har-13] Harjes, F.; Scholz-Reiter, B.; *Selbststeuernde Logistik im Umlaufmanagement von Verleihartikeln*; in Dangelmaier, W.; Laroque, C.; Klaas, A. (Hrsg.), *Simulation in Produktion und Logistik 2013: Entscheidungsunterstützung von der Planung bis zur Steuerung. 15. ASIM Fachtagung*; ASIM-Mitteilung; W.V. Westfalia Druck GmbH, Paderborn, 2013; 217–226.
- [Hei-08] Heinen, T.; Nyhuis, P.; Wiendahl, H.-P.; *Logistikprozesse in Industrie und Handel: Grundlagen der Produktionslogistik: Struktur- und Layoutplanung*; in Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H.; Furmans, K. (Hrsg.), *Handbuch Logistik*; Springer, Berlin, Heidelberg, 2008; 307–319.
- [Hen-07] Henning, P. A.; Hoffmann, D. W.; Vogelsang, H.; *Grundlagen*; in Henning, P. A.; Vogelsang, H. (Hrsg.), *Taschenbuch Programmiersprachen*; Carl Hanser Verlag, München, 2007; 24–38.
- [Her-14] Herger, M.; *Enterprise Gamification: Engaging people by letting them have fun*; CreateSpace Independent Publishing Platform, o. O., 2014.

- [Hey-16] Heynitz, H. v.; Bremicker, M.; Amadori, D. M.; Reschke, K.; *Fabrik der Zukunft: Industrie 4.0 - die Herausforderungen von morgen*; KPMG AG Wirtschaftsprüfungsgesellschaft, KPMG online, 2016.
- [Hit-08] Hitzler, P.; Krötzsch, M.; Rudolph, S.; Sure, Y.; *Semantic Web: Grundlagen*; EXamen.press; Springer, Berlin, Heidelberg; erste Edn., 2008.
- [Hoa-15] Hoad, K.; Kunc, M.; *Modeling Skills for DES and SD: An Exploratory Study on their Development in New Practitioners*; in Yilmaz, L.; Chan, W.; Moon, I.; Roeder, T.; Macal, C.; Rossetti, M. (Hrsg.), *Proceedings of the 2015 Winter Simulation Conference: Social and Behavioral Simulations*; IEEE, 2015; 3461–3468.
- [Hom-18] ten Hompel, M.; Schmidt, T.; Dregger, J.; *Materialflusssysteme: Förder- und Lagertechnik*; Springer Vieweg, Berlin; 4. Edn., 2018.
- [iTwe-19] iTween; *iTween: Easy Animation for Unity*; <http://www.pixelplacement.com/itween/index.php>; zuletzt geprüft am 05.03.2019, 2019.
- [Jah-10] Jahangirian, M.; Eldabi, T.; Naseer, A.; Stergioulas, L. K.; Young, T.; *Simulation in manufacturing and business: A review*; in *European Journal of Operational Research*; 203(1), 2010:1–13.
- [Kag-13] Kagermann, H.; Wahlster, W.; Helbig, J.; *acatech - Deutsche Akademie der Technikwissenschaften e.V.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Abschlussbericht des Arbeitskreises Industrie 4.0, Deutschlands Zukunft als Produktionsstandort sichern*; acatech online, 2013.
- [Kal-15] Kalelioğlu, F.; *A new way of teaching programming skills to K-12 students: Code.org*; in *Computers in Human Behavior*; 52, 2015:200–210.
- [Kei-07] Keil, R.; Schmidt, M.; *Material Efficiency in Higher Education - Requirements Regarding Teaching and their Practical Realisation in a New Simulation Game*; in Hilty, L. M.; Edelmann, X.; Ruf, A. (Hrsg.), *Recovery of materials and energy for resource efficiency: R'07 world congress, Switzerland, Davos, September 3 to 5, 2007*; Empa, Davos, 2007.
- [Kel-10] Kelton, W. D.; Sadowski, R. P.; Swets, N. B.; *Simulation with Arena*; McGraw-Hill International Editions; McGraw-Hill Higher Education, Boston; 5. Edn., 2010.
- [Kel-15] Kelton, W. D.; Sadowski, R. P.; Zupick, N. B.; *Simulation with Arena*; McGraw-Hill Education, New York; sixth Edn., 2015.
- [Ker-16] Kerwitz, M.; Clasen, M.; *Evaluation von Entwicklungsumgebungen für Multi-Agenten-Simulationen*; in Ruckelshausen, A.; Meyer-Aurich, A.; Rath, T.; Recke, G.; Theuvsen, B. (Hrsg.), *Intelligente Systeme - Stand der Technik und neue Möglichkeiten, Lecture Notes in Informatics (LNI)*; Gesellschaft für Informatik e.V., Bonn, 2016; 85–88.
- [Kim-09] Kim, J.; Lee, E.; Thomas, T.; Dombrowski, C.; *Storytelling in new media: The case of alternate reality games, 2001–2009*; in *First Monday: Peer-Reviewed Journal on the Internet*; 14(6), 2009.

- [Kol-71] Kolb, D. A.; Rubin, I. M.; McIntyre, J. M.; *Organizational Psychology: an experiential approach*; Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
- [Kol-84] Kolb, D. A.; *Experiential learning: experience as the source of learning and development*; Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- [Kor-11] Korn, O.; *Potenziale und Fallstricke bei der spielerischen Kontextualisierung von Lernangeboten*; in Metz, M.; Theis, F. (Hrsg.), *Digitale Lernwelt - Serious Games: Einsatz in der beruflichen Weiterbildung*; Bertelsmann, Bielefeld, 2011; 15–26.
- [Kro-09] Kropik, M.; *Produktionsleitsysteme in der Automobilfertigung*; VDI-Buch; Springer, Berlin; 1. Edn., 2009.
- [Krü-15] Krüger, J.; Nguyen, T. D.; *Automated vision-based live ergonomics analysis in assembly operations*; in *CIRP Annals - Manufacturing Technology*; 64(1), 2015:9–12.
- [Küh-06] Kühn, W.; *Digitale Fabrik: Fabriksimulation für Produktionsplaner*; Carl Hanser Verlag, München, Wien, 2006.
- [Kuh-08] Kuhn, A.; *Prozessmodelle, -ketten und -netze*; in Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H.; Furmans, K. (Hrsg.), *Handbuch Logistik*; Springer, Berlin, Heidelberg, 2008; 224–228.
- [Kum-04] Kummer, O.; Wienberg, F.; Duvigneau, M.; Schumacher, J.; Köhler, M.; Moldt, D.; Rölke, H.; Valk, R.; *An Extensible Editor and Simulation Engine for Petri Nets: RENEW*; in Cortadella, J.; Reisig, W. (Hrsg.), *Applications and Theory of Petri Nets 2004*; Springer, Berlin, Heidelberg, New York, 2004; 484–493.
- [Kur-15] Kurihara, A.; Sasaki, A.; Wakita, K.; Hosobe, H.; *A Programming Environment for Visual Block-Based Domain-Specific Languages*; in *Procedia Computer Science*; 62, 2015:287–296.
- [Lan-10] Lanza, G.; Ude, J.; *Multidimensional evaluation of value added networks*; in *CIRP Annals - Manufacturing Technology*; 59(1), 2010:489–492.
- [Law-15] Law, A. M.; *Simulation Modeling and Analysis*; McGraw-Hill, New York; 5th [international student] Edn., 2015.
- [lea-15] learn2work; *learn2work: Unternehmerisch Denken und praxisnah Lernen*; <http://www.learn2work.de/funktionsumfang/> ; letzte Aktualisierung in 2015; zuletzt geprüft am 05.03.2019, 2015.
- [Lia-02] Lian, Y.-H.; van Landeghem, H.; *An Application of Simulation and Value Stream Mapping in Lean Manufacturing*; in *SCS Europa BVBA*, 2002:300–307.
- [Lia-07] Lian, Y.-H.; van Landeghem, H.; *Analysing the effects of Lean manufacturing using a value stream mapping-based simulation generator*; in *International Journal of Production Research*; 45(13), 2007:3037–3058.

- [Löd-16] Lödding, H.; *Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration*; Springer, Berlin, Heidelberg; 3. Edn., 2016.
- [Mac-14] Macal, C.; North, M.; *Introductory Tutorial: Agent-Based Modeling and Simulation*; in Tolks, A.; Diallo, S.; Ryzhov, I.; Yilmaz, L.; Buckley, S.; Miller, J. (Hrsg.), *Proceedings of the 2014 Winter Simulation Conference: Exploring big data through simulation*; IEEE, 2014; 6–20.
- [Man-19] Manrique, V.; Rodrigo, I.; Garcia-Panella, O.; Labrador, E.; Marczewski, A.; Escribano, F.; Solarski, C.; Pagés, C.; de Polavieja, G. M.; *Gamification Design Online Course: Iversity*; <https://iversity.org/de/courses/gamification-design> ; zuletzt geprüft am 05.03.2019, 2019.
- [Mär-11] März, L.; Weigert, G.; *Simulationsgestützte Optimierung*; in März, L.; Krug, W.; Rose, O.; Weigert, G. (Hrsg.), *Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*; VDI-Buch; Springer, Berlin, Heidelberg, 2011; 3–12.
- [Mat-89] Mattern, F.; Mehl, H.; *Diskrete Simulation: Prinzipien und Probleme der Effizienzsteigerung durch Parallelisierung*; in *Informatik-Spektrum*; 12, 1989:198–210.
- [May-10] Mayer, G.; Pöge, C.; *Auf dem Weg zum Standard – Von der Idee zur Umsetzung des VDA Automotive Bauteinkastens*; in Zülch, G.; Stock, P. (Hrsg.), *Integrationsaspekte der Simulation: Technik, Organisation und Personal*; ASIM-Mitteilung; KIT Scientific Publishing, Karlsruhe, 2010; 29–36.
- [May-12] Mayer, G.; Spieckermann, S.; Wenzel, S.; *Steigerung der Produktivität in Simulationsstudien mit Assistenzwerkzeugen*; in *ZWF – Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Carl Hanser Verlag, München; 107(03), 2012:174–177.
- [May-15a] Mayer, G.; Mieschner, M.; *Geführte Assistenz für Ablaufsimulationsprojekte*; in Rabe, M.; Clausen, U. (Hrsg.), *Simulation in Production and Logistics 2015*; ASIM-Mitteilung; Fraunhofer Verlag, Stuttgart, 2015; 605–612.
- [May-15b] Mayer, T.; Pappert, F. S.; Rose, O.; *A Natural-language-based Simulation Modelling Approach*; in Rabe, M.; Clausen, U. (Hrsg.), *Simulation in Production and Logistics 2015*; ASIM-Mitteilung; Fraunhofer Verlag, Stuttgart, 2015; 661–670.
- [May-17] Mayer, G.; Mieschner, M.; *Industrie 4.0 -Chance oder Risiko für die Ablaufsimulation*; in Wenzel, S.; Peter, T. (Hrsg.), *Simulation in Produktion und Logistik 2017*; kassel university press, Kassel, 2017; 1–8.
- [McF-13] McFarland, R.; Reise, C.; Postawa, A. B.; Seliger, G.; *Learnstruments in value creation and learning centered work place design*; in Seliger, G. (Hrsg.), *Proceedings of the 11th Global Conference on Sustainable Manufacturing - Innovative Solutions*; Universitätsverlag der TU Berlin, Berlin, 2013; 624–629.

- [McG-11] McGonigal, J.; *Reality is broken: Why games make us better and how they can change the world*; Penguin Press, New York, 2011.
- [Men-18] Menn, J. P.; Muschard, B.; Schumacher, B.; Sieckmann, F.; Kohl, H.; Seliger, G.; *Learning-instruments: Learning-conducive artefacts to foster learning productivity in production engineering*; in *CIRP Annals - Manufacturing Technology*; 67(1), 2018:459–462.
- [Met-11] Metz, M.; Theis, F.; *Mit Serious Games zum Lernerfolg*; in Metz, M.; Theis, F. (Hrsg.), *Digitale Lernwelt - Serious Games: Einsatz in der beruflichen Weiterbildung*; Bertelsmann, Bielefeld, 2011; 63–68.
- [Mey-17] Meyer, J.-D.; Klußmann, J.; *Evakuierungssimulation von Kreuzfahrtschiffen*; in Wenzel, S.; Peter, T. (Hrsg.), *Simulation in Produktion und Logistik 2017*; kassel university press, Kassel, 2017; 335–344.
- [moo-19] moodle; *Leistungsstark. Sicher. Open-Source*; <https://moodle.org> ; zuletzt geprüft am 05.03.2019, 2019.
- [Mos-13] Mosblech, C.; Marrenbach, F.; Wang-Görgens, N.; *Simulation in der innerbetrieblichen Weiterbildung: Technische Integration von Simulationssoftware*; in *wt-online, Springer-VDI-Verlag, Düsseldorf*; 103(3), 2013:252–256.
- [Mou-14] Mourtzis, D.; Doukas, M.; Bernidaki, D.; *Simulation in Manufacturing: Review and Challenges*; in *Procedia CIRP, CIRP Sponsored DET 2014 Conference*; 25, 2014:213–229.
- [Mül-15] Müller, B. C.; Reise, C.; Seliger, G.; *Gamification in Factory Management Education – A Case Study with Lego Mindstorms*; in *Procedia CIRP, 12th Global Conference on Sustainable Manufacturing – Emerging Potentials*; 26, 2015:121–126.
- [Mül-16a] Müller, B. C.; Nguyen, T. D.; Dang, Q.-V.; Duc, B. M.; Seliger, G.; Krüger, J.; Kohl, H.; *Motion Tracking Applied in Assembly for Worker Training in different Locations*; in *Procedia CIRP, 23rd CIRP Conference on Life Cycle Engineering*; 48, 2016:460–465.
- [Mül-16b] Müller, B. C.; Reise, C.; Duc, B. M.; Seliger, G.; *Simulation-games for Learning Conducive Workplaces: A Case Study for Manual Assembly*; in *Procedia CIRP, 13th Global Conference on Sustainable Manufacturing – Decoupling Growth from Resource Use*; 40, 2016:353–358.
- [Mül-17a] Müller, B. C.; Kühn, M.; Schele, M.; Seliger, G.; *Mobile Lernumgebung zur simulativen Gestaltung von Produktionssystemen*; in Wenzel, S.; Peter, T. (Hrsg.), *Simulation in Produktion und Logistik 2017*; kassel university press, Kassel, 2017; 345–354.
- [Mül-17b] Müller, B. C.; Menn, J. P.; Seliger, G.; *Procedure for Experiential Learning to Conduct Material Flow Simulation Projects, Enabled by Learning Factories*; in *Procedia Manufacturing, 7th Conference on Learning Factories, CLF 2017*; 9, 2017:283–290.
- [Mun-17] Munkelt, T.; Behne, S.; Wacker, M.; Völker, S.; *A Model Factory in Augmented Reality as an Eye-Catcher at Exhibitions and Fairs*; in *SNE Simulation Notes Europe*; 27(1), 2017:1–8.

- [Mus-10] Mustafee, N.; Katsaliaki, K.; *The blood supply game*; in Johansson, B.; Jain, S.; Montoya-Torres, J.; Hagan, J.; Yücesan, E. (Hrsg.), *Proceedings of the 2010 Winter Simulation Conference*; IEEE, 2010; 327–338.
- [MyS-19a] MySQL; *MySQL 8.0 Reference Manual: What is MySQL?*; <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> ; zuletzt geprüft am 05.03.2019, 2019.
- [MyS-19b] MySQL; *MySQL Community Edition*; <https://www.mysql.com/de/products/community/> ; zuletzt geprüft am 05.03.2019, 2019.
- [Nas-14] Nassehi, A.; *Simulation of Manufacturing Systems*; in Laperrière, L.; Reinhart, G. (Hrsg.), *CIRP Encyclopedia of Production Engineering*; Springer, Heidelberg, New York, Dordrecht, London, 2014; 1120–1123.
- [Nie-13] Niegemann, H. M.; *Didaktische Medienproduktion: Instruktionsdesign*; in *Online-Zeitschrift für Wissenschaft und Praxis: Medienproduktion: Didaktische Medienproduktion*; 3, 2013:11–14.
- [Noc-91] Noche, B.; Wenzel, S.; *Marktspiegel Simulationstechnik in Produktion und Logistik*; Praxiswissen aktuell; TÜV Rheinland, Köln, 1991.
- [Noc-93] Noche, B.; Bernhard, W.; Krauth, J.; Meyer, R.; Wenzel, S.; *Simulationsinstrumente im Überblick*; in Kuhn, A.; Reinhard, A.; Wiendahl, H.-P. (Hrsg.), *Handbuch Simulationsanwendungen in Produktion und Logistik*; Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden, 1993; 267–308.
- [Nor-18] North, K.; Reinhardt, K.; Sieber-Suter, B.; *Kompetenzmanagement in der Praxis: Mitarbeiterkompetenzen systematisch identifizieren, nutzen und entwickeln. Mit vielen Praxisbeispielen*; Springer Gabler, Wiesbaden; 3., aktualisierte und erweiterte Edn., 2018.
- [Ola-14] Olaitan, O.; Geraghty, J.; Young, P.; Dagkakis, G.; Heavey, C.; Bayer, M.; Perrin, J.; Robin, S.; *Implementing ManPy, a Semantic-free Open-source Discrete Event Simulation Package, in a Job Shop*; in *Procedia CIRP, 8th International Conference on Digital Enterprise Technology - DET 2014 - "Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution*; 25, 2014:253–260.
- [Oua-15] Ouahbi, I.; Kaddari, F.; Darhmaoui, H.; Elachqar, A.; Lahmine, S.; *Learning Basic Programming Concepts by Creating Games with Scratch Programming Environment*; in *Procedia - Social Behavioral Sciences, WCES*; 191, 2015:1479–1482.
- [Pad-14] Padilla, J. J.; Diallo, S. Y.; Barraco, A.; Lynch, C. J.; Kavak, H.; *Cloud-based simulators: Making simulations accessible to non-experts and experts alike*; in Tolks, A.; Diallo, S.; Ryzhov, I.; Yilmaz, L.; Buckley, S.; Miller, J. (Hrsg.), *Proceedings of the 2014 Winter Simulation Conference: Exploring big data through simulation*; IEEE, 2014; 3630–3639.
- [Pad-15] Padilla, J. J.; Romero-Hall, E.; Diallo, S. Y.; Barraco, A.; Kavak, H.; Lynch, C. J.; Gore, R. J.; Sheth-Chandra, M.; *Modeling and Simulation As a Service (MSaaS) for Education*:

- Learning STEM Concepts Through Simulation Use and Building*; in *SummerSim 2015, Proceedings of the Conference on Summer Computer Simulation (SCSC)*, 2015:1–9.
- [Pad-16] Padilla, J. J.; Lynch, C. J.; Diallo, S. Y.; Gore, R. J.; Barraco, A.; Kavak, H.; Jenkins, B.; *Using simulation games for teaching and learning discrete-event simulation*; in Roeder, T. M.; Frazier, P. I.; Szechtman, R.; Zhou, E.; Huschka, T.; Chick, S. E. (Hrsg.), *Proceedings of the 2016 Winter Simulation Conference: Simulating complex service systems*; IEEE, 2016; 3375–3384.
- [Pal-17] Palacios Neffke, J. C.; *Contribution to the development of technology-enhanced education in manufacturing and energy generation*; Dissertation; Technische Universität Berlin; Berlin, 2017.
- [Pan-17] Pandey, A.; *A Brief History of Gamification*; in *XRDS*; 24(1), 2017:13.
- [Pas-10] Paschmann, W.; *Montage*; in Lechner, C.; Seume, J. (Hrsg.), *Stationäre Gasturbinen*; VDI-Buch; Springer, Berlin, 2010; 837–841.
- [Pet-62] Petri, C. A.; *Kommunikation mit Automaten*; Dissertation; Technische Hochschule Darmstadt; Darmstadt, 1962.
- [Poc-19] PocketGamer; *Kategorien im App Store nach Anzahl der Apps im Februar 2019*; <https://de.statista.com/statistik/daten/studie/217342/umfrage/beliebteste-kategorien-im-app-store-nach-anzahl-der-apps/> ; letzte Aktualisierung am 04.02.2019; zuletzt geprüft am 23.02.2019, 2019.
- [Poi-08] Poiger, M.; Reiner, G.; *Gestaltung und Bewertung von „Just-in-Sequence“ - Anlieferung in der Automobilindustrie*; in Engelhardt-Nowitzki, C.; Nowitzki, O.; Krenn, B. (Hrsg.), *Praktische Anwendung der Simulation im Materialflussmanagement: Erfolgsfaktoren und Implementierungsszenarien*; Gabler Edition Wissenschaft. Leobener Logistik Cases; Gabler, Wiesbaden, 2008; 133–143.
- [Poj-18] Pojahr, P.; Herte, M.; *Spielmechanik*; in Beil, B.; Hensel, T.; Rauscher, A. (Hrsg.), *Game Studies: Film, Fernsehen, Neue Medien*; Springer VS, Wiesbaden, 2018; 235–249.
- [Pos-11] Postawa, A. B.; Reise, C.; Seliger, G.; *Training on the Job in Remanufacturing supported by Information Technology Systems*; in Seliger, G. (Hrsg.), *Sustainable Manufacturing - Shaping Global Value Creation: 9th Global Conference on Sustainable Manufacturing*; Universitätsverlag der TU Berlin, Berlin, 2011; 337–342.
- [Pos-19] Positech Games; *Production line: Efficiency is everything*; <http://www.positech.co.uk/productionline/index.html> ; zuletzt geprüft am 05.03.2019, 2019.
- [Pot-13] Potente, T.; Varandani, R.; Prote, J.-P.; *Gamification in management decisions: judging global production networks in a cyber-physical way*; in *Advanced Materials Research*; 769, 2013:327–334.
- [Pre-01] Prensky, M.; *Digital Natives, Digital Immigrants*; in *On the Horizon*; 9(5), 2001:1–6.

- [Pri-16] Prinz, C.; Morlock, F.; Freith, S.; Kreggenfeld, N.; Kreimeier, D.; Kuhlenkötter, B.; *Learning Factory Modules for Smart Factories in Industrie 4.0*; in *Procedia CIRP, 6th CLF - 6th CIRP Conference on Learning Factories*; 54, 2016:113–118.
- [Pro-19] ProModel Corporation; *Process Simulator: Dynamic Process Improvement*; <https://www.promodel.com/products/ProcessSimulator> ; zuletzt geprüft am 05.03.2019, 2019.
- [Rab-08] Rabe, M.; Spieckermann, S.; Wenzel, S.; *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*; VDI-Buch; Springer, Berlin, Heidelberg, 2008.
- [Rei-14] Reise, C.; Müller, B. C.; Seliger, G.; *Resource Efficiency Learning Game – Electric Scooter Game*; in *Procedia CIRP, 21st CIRP Conference on Life Cycle Engineering*; 15, 2014:355–360.
- [Rei-16] Reise, C.; Phan, L.; *Sustainable Manufacturing in Vietnamese Engineering Education – Approaches from the Vietnamese-German University*; in *Procedia CIRP, 13th Global Conference on Sustainable Manufacturing – Decoupling Growth from Resource Use*; 40, 2016:341–346.
- [Rem-16] Remi-Omosowon, A.; Cant, R.; Langensiepen, C.; *Applying Gamification Principles to a Container Loading System in a Warehouse Environment*; in *SNE Simulation Notes Europe*; 26(2), 2016:99–104.
- [Res-09] Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; Kafai, Y.; *Scratch: Programming for All*; in *Communications of the ACM*; 52(11), 2009:60–67.
- [Ret-19] RethinkDB; *Rethink DB: Frequently asked questions*; <https://www.rethinkdb.com/faq/> ; zuletzt geprüft am 05.03.2019, 2019.
- [Rhe-04] Rheinberg, F.; *Motivation*; Urban-Taschenbücher; Kohlhammer, Stuttgart; 5., überarbeitete und erweiterte Edn., 2004.
- [Roe-15] Roeder, T. M.; Miyaoka, J.; *Using Simulation as a Teaching Tool in an Introductory Operations Management Course*; in Yilmaz, L.; Chan, W.; Moon, I.; Roeder, T.; Macal, C.; Rossetti, M. (Hrsg.), *Proceedings of the 2015 Winter Simulation Conference: Social and Behavioral Simulations*; IEEE, 2015; 3481–3489.
- [Ros-11] Rose, O.; März, L.; *Simulation*; in März, L.; Krug, W.; Rose, O.; Weigert, G. (Hrsg.), *Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*; VDI-Buch; Springer, Berlin, Heidelberg, 2011; 13–19.
- [Rud-16] Rudel, S.; *Prozess-Simulation in KMU: Entwicklung und Analyse eines Petrinetz-basierten Baukastensystems als Instrument zur Modellierung und Simulation von Produktions- und Logistikprozessen - speziell ausgelegt zur Unterstützung von kleinen und mittleren Unternehmen (KMU)*; Dissertation; Universität Duisburg-Essen; Essen, 2016.

- [Rud-17] Rudel, S.; *Prozess-Simulation in kleinen und mittleren Unternehmen mittels des Baukastensystems KMUSimMetall*; in Wenzel, S.; Peter, T. (Hrsg.), *Simulation in Produktion und Logistik 2017*; kassel university press, Kassel, 2017; 433–442.
- [Rya-00] Ryan, R. M.; Deci, E. L.; *Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions*; in *Contemporary educational psychology*; 25(1), 2000:54–67.
- [Sáe-16] Sáez-López, J.-M.; Román-González, M.; Vázquez-Cano, E.; *Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools*; in *Computers & Education*; 97, 2016:129–141.
- [San-16] Sanders, B.; Shuttleworth, D.; Deuro, J.; Padilla, J. J.; *Learning discrete event simulation design methodology via interactive and collaborative projects*; in *SpringSim-ANSS, Proceedings of the 49th Annual Simulation Symposium*, 2016:1–8.
- [Sch-88] Schmidt, B.; *Simulation von Produktionssystemen*; in Feldmann, K.; Schmidt, B. (Hrsg.), *Simulation in der Fertigungstechnik*; Springer, Berlin, 1988; 1–45.
- [Sch-06] Scholz-Reiter, B.; Höhns, H.; *Selbststeuerung logistischer Prozesse mit Agentensystemen*; in Schuh, G. (Hrsg.), *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte*; VDI-Buch; Springer, Berlin, Heidelberg, New York, 2006; 741–780.
- [Sch-08a] Schmidt, A.; Schneider, M.; *Lager- und Materialflussprozesse*; in Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H.; Furmans, K. (Hrsg.), *Handbuch Logistik*; Springer, Berlin, Heidelberg, 2008; 371–404.
- [Sch-08b] Schmidt, M.; Nyhuis, P.; Wiendahl, H.-P.; *Grundlagen der Produktionslogistik: Produktionsplanung- und -steuerung*; in Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H.; Furmans, K. (Hrsg.), *Handbuch Logistik*; Springer, Berlin, Heidelberg, 2008; 323–343.
- [Sch-09] Schulmeister, R.; *Grundlagen hypermedialer Lernsysteme: Theorie - Didaktik - Design*; Oldenbourg, München, Wien; 4., überarbeitete und aktualisierte Edn., 2009.
- [Sch-13] Schwarz, C.; Schachmanow, J.; Sauer, J.; Overmeyer, L.; Ullmann, G.; *Selbstgesteuerte Fahrerlose Transportsysteme*; in *Logistics Journal*; 2013(12), 2013.
- [Sch-14a] Scherfke, S.; *SimPy: Discrete-event simulation with SimPy*; *EuroPython*; <https://av.tib.eu/media/20032> ; zuletzt geprüft am 05.03.2019, 2014.
- [Sch-14b] Schmuck, T.; *Ein Beitrag zur effizienten Gestaltung globaler Produktions- und Logistiknetzwerke mittels Simulation*; Dissertation; Friedrich-Alexander-Universität Erlangen-Nürnberg; Erlangen-Nürnberg, 2014.
- [Sch-14c] Schönherr, O.; *Modellierung, Simulation und Transformation von diskreten Prozessen in der Produktion und Logistik auf der Basis von SysML*; Dissertation; Universität der Bundeswehr München; München, 2014.

- [Sch-16] Scholz, M.; Kreitlein, S.; Lehmann, C.; Böhner, J.; Franke, J.; Steinhilper, R.; *Integrating Intralogistics into Resource Efficiency Oriented Learning Factories*; in *Procedia CIRP, 6th CLF - 6th CIRP Conference on Learning Factories*; 54, 2016:239–244.
- [Sch-18] Schröter, F.; *Figur*; in Beil, B.; Hensel, T.; Rauscher, A. (Hrsg.), *Game Studies: Film, Fernsehen, Neue Medien*; Springer VS, Wiesbaden, 2018; 109–128.
- [Sch-19] Schumacher, B. C.; Balzar, F.; *Learnstrument zur Einarbeitung in ereignisdiskrete Simulationsinstrumente*; https://www.mf.tu-berlin.de/menue/forschung_und_industrie/kompetenzfelder/lernzeuge/learnstrument_zur_einarbeitung_in_ereignisdiskrete_simulationsinstrumente/ ; letzte Aktualisierung am 11.01.2019; zuletzt geprüft am 02.03.2019, 2019.
- [Scr-19] Scratch; *Über Scratch*; scratch.mit.edu ; zuletzt geprüft am 05.03.2019, 2019.
- [Sha-98] Shannon, R. E.; *Introduction to the art and science of simulation*; in Medeiros, D. J.; Watson, E. F.; Carson, J. S.; Manivannan, M. S. (Hrsg.), *Proceedings of the 1998 Winter Simulation Conference*; IEEE, 1998; 7–14.
- [Sie-14] Siemens PLM Software; *Tecnomatix Plant Simulation: Simulate, visualize, analyze and optimize production systems and logistics processes*; https://www.plm.automation.siemens.com/media/store/en_us/Tecnomatix%20Plant-%20Simulation_7541_tcm29-2062.pdf ; letzte Aktualisierung in 2014; zuletzt geprüft am 05.03.2019, 2014.
- [Sie-19] Siemens; *Tecnomatix Plant Simulation Version 13: Hilfe*; zuletzt geprüft am 05.03.2019, 2019.
- [Sie-20] Siemens; *Interval Extension - Combustion System Upgrades*; <https://new.siemens.com/global/en/products/energy/services/performance-enhancement/modernization-upgrades/gas-turbines/interval-extension-combustion-system-upgrade-sgt-5000f.html> ; zuletzt geprüft am 29.01.2020, 2020.
- [Sim-19a] SimPlan AG; *MobiSim - Forschungsprojekt zur Schaffung eines mobilen Wertstromsimulators für die Produktion*; <https://www.simplan.de/forschung/forschungsprojekt-mobisim/> ; zuletzt geprüft am 05.03.2019, 2019.
- [Sim-19b] SimPlan AG; *SimVSM: Die App zur Wertstromanalyse & Simulation*; <https://www.simplan.de/software/simvsm/> ; zuletzt geprüft am 05.03.2019, 2019.
- [Sni-13] Sniekers, D.; *A Generic Simulation-based Perioperative Decision Support Tool for Tactical Decisions*; Dissertation; University of Toronto; Toronto, 2013.
- [Sol-09] Solding, P.; Gullander, P.; *Concepts for simulation based Value Stream Mapping*; in Rossetti, M.; Hill, R.; Johansson, B.; Dunkin, A.; Ingalls, R. G. (Hrsg.), *Proceedings of the 2009 Winter Simulation Conference*; IEEE, 2009; 2231–2237.
- [Spa-13] Spath, D.; Ganschar, O.; Gerlach, S.; Hämmerle, M.; Krause, T.; Schlund, S.; *Produktionsarbeit der Zukunft - Industrie 4.0*; Fraunhofer-Verlag, Stuttgart, 2013.

- [SQL-19] SQLite; *About SQLite*; <https://www.sqlite.org/about.html> ; zuletzt geprüft am 05.03.2019, 2019.
- [Sta-16] Statistic Brain Research Institute; *Anzahl der weltweit insgesamt heruntergeladenen Apps nach App Stores im Jahr 2016 (in Milliarden)*; <https://de.statista.com/statistik/daten/studie/321649/umfrage/insgesamt-heruntergeladene-apps-pro-app-store-weltweit/> ; letzte Aktualisierung am 02.06.2016; zuletzt geprüft am 23.02.2019, 2016.
- [Ste-11] Stewart, B.; *Personality And Play Styles: A Unified Model*; http://www.gamasutra.com/view/feature/134842/personality_and_play_styles_a_.php ; letzte Aktualisierung am 01.09.2011; zuletzt geprüft am 05.03.2019, 2011.
- [Sto-17] Stoffel, T.; *Plant Simulation in Design and Optimization of Manufacturing Processes of Gas Turbine Burners: Presentation at the Plant Simulation Worldwide User Conference June 2017*; <https://community.plm.automation.siemens.com/t5/Plant-Simulation-Worldwide-User-Conference/Presentation-Plant-Simulation-in-Design-and-Optimization-of/gpm-p/426856> ; letzte Aktualisierung am 14.08.2017; zuletzt geprüft am 05.03.2019, 2017.
- [Str-10] Straßburger, S.; Bergmann, S.; *Modellgenerierung im Kontext der Digitalen Fabrik - Stand der Technik und Herausforderungen*; in Zülch, G.; Stock, P. (Hrsg.), *Integrationsaspekte der Simulation: Technik, Organisation und Personal*; ASIM-Mitteilung; KIT Scientific Publishing, Karlsruhe, 2010; 37–44.
- [Swa-15] Swain, J. J.; *Simulation Software Survey: Simulated worlds*; in *OR/MS Today*; Oktober 42(5), 2015:36–49.
- [Swa-17] Swain, J. J.; *Simulation Software Survey - Simulation: new and improved reality show*; in *OR/MS Today*; Oktober 44(5), 2017:38–49.
- [Tis-15] Tisch, M.; Ranz, F.; Abele, E.; Metternich, J.; Hummel, V.; *Learning Factory Morphology – Study Of Form And Structure Of An Innovative Learning Approach In The Manufacturing Domain*; in *TOJET: The Turkish Online Journal of Educational Technology - July 2015, Special Issue 2 for INTE 2015*; Special Issues 2, 2015:346–363.
- [Tob-11] Tobail, A.; Crowe, J.; Arisha, A.; *Learning by Gaming: Supply Chain Application*; in Jain, S.; Creasey, R. R.; Himmelspace, J.; White, K. P.; Fu, M. (Hrsg.), *Proceedings of the 2011 Winter Simulation Conference*; IEEE, 2011; 3940–3951.
- [Tsc-16] Tschöpe, S.; *Fabrikplanungsspezifische Modellierungssprache*; Dissertation; Gottfried Wilhelm Leibniz Universität Hannover; Hannover, 2016.
- [TZI-19] TZI Universität Bremen; *PlaSMA Multiagent Simulation*; <http://plasma.informatik.uni-bremen.de/> ; zuletzt geprüft am 16.01.2019, 2019.
- [Ull-12] Ullén, F.; de Manzano, Ö.; Almeida, R.; Magnusson, P. K.; Pedersen, N. L.; Nakamura, J.; Csíkszentmihályi, M.; Madison, G.; *Proneness for psychological flow in everyday life*:

- Associations with personality and intelligence*; in *Personality and Individual Differences*; 52(2), 2012:167–172.
- [Uni-19] Unity3D; *Unity 2017: Die weltweit führende Creation-Engine für Gaming*; <https://unity3d.com/de/unity> ; zuletzt geprüft am 05.03.2019, 2019.
- [Uyg-15] Uygun, Y.; Ringeln, M.; Straub, N.; *Pull-Prinzip*; in Dombrowski, U.; Mielke, T. (Hrsg.), *Ganzheitliche Produktionssysteme: Aktueller Stand und zukünftige Entwicklungen*; VDI-Buch; Springer Vieweg, Berlin, 2015; 110–128.
- [VDA-09] VDA; *Verband der Automobilindustrie Empfehlung: 3D Datenaustausch in der Fabrikplanung: Arbeitskreis "PLM"*; Verband der Automobilindustrie, Frankfurt, 2009.
- [VDA-4810] *Verband der Automobilindustrie Richtlinie 4810 Teil 1 Definition: Ausprägung der Logistiksimulationen*; VDA 4810 T1; Verband der Automobilindustrie, Berlin, März 2011.
- [VDA-4811] *Verband der Automobilindustrie Richtlinie 4811: Qualitätskriterien für Simulationsstudien der Ablaufsimulation*; VDA 4811; Verband der Automobilindustrie, Berlin, Mai 2013.
- [VDI-2689] *Verein Deutscher Ingenieure Richtlinie 2689 Entwurf: Leitfaden für Materialflussuntersuchungen*; VDI 2689 Entwurf; Beuth Verlag, Berlin, Mai 2010.
- [VDI-2710] *Verein Deutscher Ingenieure Richtlinie 2710 Blatt 3: Einsatzgebiete der Simulation für Fahrerlose Transportsysteme*; VDI 2710 Blatt 3; Beuth Verlag, Berlin, Mai 2014.
- [VDI-2870a] *Verein Deutscher Ingenieure Richtlinie 2870: Ganzheitliche Produktionssysteme: Grundlagen, Einführung und Bewertung*; VDI 2870 Blatt 1; Beuth Verlag, Berlin, Juli 2012.
- [VDI-2870b] *Verein Deutscher Ingenieure Richtlinie 2870: Ganzheitliche Produktionssysteme: Methodenkatalog*; VDI 2870 Blatt 2; Beuth Verlag, Berlin, Februar 2013.
- [VDI-3633a] *Verein Deutscher Ingenieure Richtlinie 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen: Grundlagen*; VDI 3633 Blatt 1; Beuth Verlag, Berlin, Dezember 2014.
- [VDI-3633b] *Verein Deutscher Ingenieure Richtlinie 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen: Begriffe*; VDI 3633 Blatt 0 Entwurf; Beuth Verlag, Berlin, Dezember 2013.
- [VDI-4008a] *Verein Deutscher Ingenieure Richtlinie 4008: Methoden der Zuverlässigkeit: Petri-Netze*; VDI 4008 Blatt 4; Beuth Verlag, Berlin, Juli 2008.
- [VDI-4008b] *Verein Deutscher Ingenieure Richtlinie 4008: Monte-Carlo-Simulation*; VDI 4008 Blatt 6; Beuth Verlag, Berlin, April 1999.
- [VDI-4465] *Verein Deutscher Ingenieure Richtlinie 4465: Modellierung und Simulation: Modellbildungsprozess*; VDI 4465 Blatt 1 Entwurf; Beuth Verlag, Berlin, Mai 2016.
- [VDI-4499a] *Verein Deutscher Ingenieure Richtlinie 4499: Digitale Fabrik: Grundlagen*; VDI 4499 Blatt 1; Beuth Verlag, Berlin, Februar 2008.

- [VDI-5200] Verein Deutscher Ingenieure Richtlinie 5200: *Fabrikplanung: Planungsvorgehen*; VDI 5200 Blatt 1; Beuth Verlag, Berlin, Februar 2011.
- [Vie-19] Vietnamese-German University; *Vietnamese-German Training Center for Lean and Green Production (VGTCLGP)*; <https://vgu.edu.vn/ca/vietnamese-german-training-center-for-lean-and-green-production-vgtclgp>; zuletzt geprüft am 05.03.2019, 2019.
- [War-10] Warden, T.; Porzel, R.; Gehrke, J. D.; Herzog, O.; Langer, H.; Malaka, R.; *Towards Ontology-based Multiagent Simulations: The PlaSMA Approach*; in Bargiela, A.; Azam Ali, S.; Crowley, D.; Kerckhoffs, E. J. (Hrsg.), *24th European Conference on Modelling and Simulation (ECMS 2010)*; European Council for Modelling and Simulation, 2010; 50–56.
- [Wei-15] Weintrop, D.; Wilensky, U.; *To block or not to block, that is the question: Students' perceptions of blocks-based programming*; in *Proceedings of the 14th International Conference on Interaction Design and Children*, 2015:199–208.
- [Wen-08a] Wenzel, S.; *Modellbildung und Simulation in der Logistik*; in Lehmann, A.; Krieger, T. (Hrsg.), *Modellbildung und Simulation, Operations Research - Methoden zur Gestaltung der Zukunft der Bundeswehr und zur Unterstützung von Einsätzen*; Institut für Technik intelligenter Systeme e.V. an der Universität der Bundeswehr, München, 2008; 68–84.
- [Wen-08b] Wenzel, S.; *Modellierung logistischer Systeme: Simulation Logistischer Systeme*; in Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H.; Furmans, K. (Hrsg.), *Handbuch Logistik*; Springer, Berlin, Heidelberg, 2008; 73–94.
- [Wen-08c] Wenzel, S.; Collisi-Böhmer, S.; Pitsch, H.; Rose, O.; Weiß, M.; *Qualitätskriterien für die Simulation in Produktion und Logistik: Planung und Durchführung von Simulationsstudien*; VDI-Buch; Springer, Berlin, Heidelberg, 2008.
- [Wie-87] Wieneke-Toutaoui, B.; *Rechnerunterstütztes Planungssystem zur Auslegung von Fertigungsanlagen*; Dissertation; Technische Universität Berlin; Berlin, 1987.
- [Wie-10] Wiedermann, A.; *Gasturbinentypen - eine Übersicht*; in Lechner, C.; Seume, J. (Hrsg.), *Stationäre Gasturbinen*; VDI-Buch; Springer, Berlin, 2010; 1–10.
- [Wie-14] Wiendahl, H.-P.; Reichardt, J.; Nyhuis, P.; *Systematik der Veränderungsfähigkeit*; in Wiendahl, H.-P.; Nyhuis, P.; Reichardt, J. (Hrsg.), *Handbuch Fabrikplanung: Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten*; Hanser, München, 2014; 117–150.
- [Wil-10] Wilson, A.; Moffat, D. C.; *Evaluating Scratch to introduce younger schoolchildren to programming*; in *22nd Annual Psychology of Programming Interest Group workshop*, 2010.
- [Wim-11] Wimpff, D.-P.; Seitz, S.; Ergin, T.; Korn, O.; Rommel, S.; Hermann, M.; Berkemeyer, D.; Iuliano, G.; Schuler, D.; Kless, D.; *Abschlussbericht: simKMU - Internet- und webbasierte Simulationsdienste: Forschungsergebnisse im Teilvorhaben "Grundlagen, Konzeption und Intralogistik" des Verbundprojekts simKMU*; Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), 2011.

- [Zac-15] Zachary, F.-W.; *Investigating the Design of Effective Gamification: Experiences for Mobile Applications and Devices*; Dissertation; Queensland University of Technology; Queensland, 2015.
- [Zac-18] Zachary, F.-W.; *A brief history of gamification*; <http://zacfitzwalter.com/gamification/history> ; letzte Aktualisierung in 2018; zuletzt geprüft am 05.03.2019, 2018.
- [Zic-11] Zichermann, G.; Cunningham, C.; *Gamification by design: Implementing game mechanics in web and mobile apps*; O'Reilly Media, Sebastopol; 1st Edn., 2011.
- [Zül-06] Zülch, G.; Stohwasser, S.; *ROFA-Plus - Ein simulationsunterstütztes Planspiel zur Reorganisation von Produktionssystemen*; in Engelhardt-Nowitzki, C. (Hrsg.), *Ausbildung in der Logistik*; DUV, Wiesbaden, 2006; 125–130.
- [Zwa-03] Zwahr, A.; *Brockhaus Universallexikon: Von A - Z in 26 Bänden*; Weltbild, Augsburg, 2003.
- [Zyd-05] Zyda, M.; *From visual simulation to virtual reality to games*; in *Computer*; 38(9), 2005:25–32.

Anhang

A Marktgängige Simulationsinstrumente

Die Tabelle A.1 zeigt die ereignisdiskreten Simulationsinstrumente nach SWAIN, deren Eigenschaften im Kapitel 2.3.2 ausgewertet sind [Swa-17].

Tabelle A.1: Liste marktgängiger ereignisdiskreter Simulationsinstrumente und deren Eigenschaften nach [Swa-17]

Software	Primary Markets for which the software is applied	Can the software utilize other software to perform specialized functions?	Can the software be controlled or run by an external program?	Graphical model construction (icon or drag-and-drop)	Model building using programming/ access to programmed modules
AnyLogic	Supply Chains, Logistics, Manufacturing, Healthcare, Pedestrian Flows, Traffic, Mining, Defense, Social Processes, Marketing.	●	●	●	●
CSIM20	system design and evaluation	●	●	○	●
Enterprise Dynamics	Warehouses/Distribution Centers, Airport, Harbors, Healthcare, Pharmaceuticals, FMCG	●	●	●	●
ExtendSim DE	Communication, Healthcare, Manufacturing, Security & Defense, Aerospace, Transportation, Sustainability, Logistics, H2O Mgmt.	●	●	●	●
FlexSim	Manufacturing, packaging, warehousing, material handling, supply chain, logistics, healthcare, factory, aerospace, mining.	●	●	●	●
Integrated Performance Modelling Environment (IP-ME)	Global defense, manufacturing, air traffic control	●	●	●	●
Process Simulator	DoD, Gov, Manufacturing, Pharma, Med Device, Supply Chain and Logistics, Warehouse and DC, Healthcare	●	●	●	●

Legende:

● – erfüllt

○ – nicht erfüllt

Fortsetzung von Tabelle A.1

Software	Primary Markets for which the software is applied	Can the software utilize other software to perform specialized functions?	Can the software be controlled or run by an external program?	Graphical model construction (icon or drag-and-drop)	Model building using programming/ access to programmed modules
SAS Simulation Studio	Manufacturing, banking, pharmaceutical, health care, energy, government agencies, retail, education, transportation, etc.	●	●	●	○
Simio Design Edition	Academic, Aerospace & Defense, Airports, Healthcare, Manufacturing, Mining, Military, Oil & Gas, Supply Chain, Transportation	●	●	●	●
SIMPROCESS	Military, government, commercial, education, international, foreign	●	●	●	●
SIMUL8 Professional	Manufacturing, Healthcare, Education, Engineering, Supply Chain, Logistics, Govt, BPMN, Lean, Automotive, Call Centers	●	●	●	●
Solver SDK Platform	Finance/investment, insurance, pharmaceuticals, oil and gas, MBA education	●	●	○	●
Tecnomatix Plant Simulation	Automotive OEM & Supplier, Aerospace & Defense, Consumer Products, Logistics, Electronics, Machinery, Healthcare, Consulting	●	●	●	●
WITNESS	Business Planning, Process Optimisation and Decision Making	●	●	●	●
Summe „erfüllt“:		14 von 14	14 von 14	12 von 14	13 von 14

Legende:

● – erfüllt

○ – nicht erfüllt

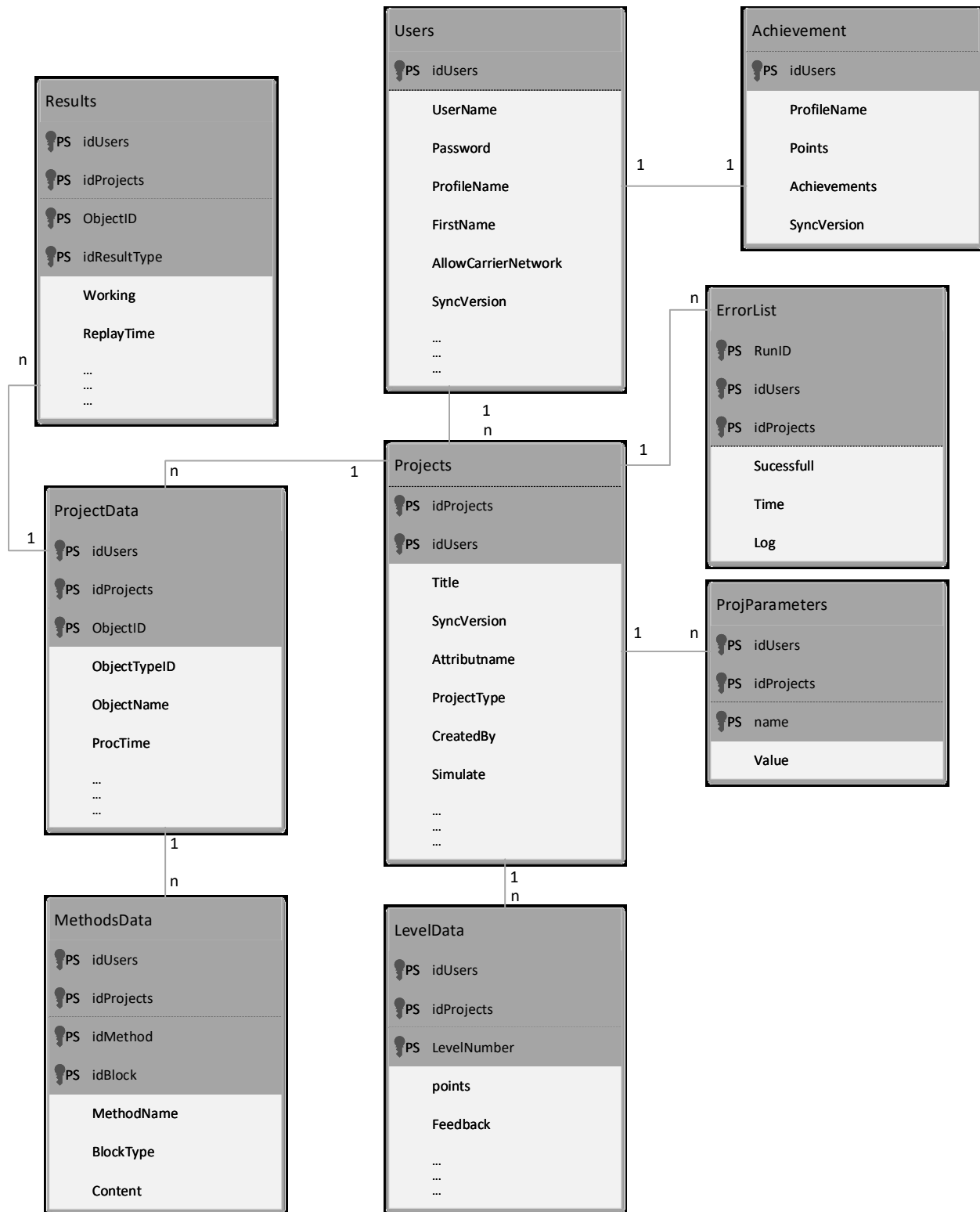
B Skripte und Tabellen Umsetzung Cloud-IT-Architektur

Das der Cloud-IT-Architektur zugrunde liegende Datenbankmodell ist in Abbildung B.1 abgedruckt. Für das vorgeschlagene Datenbankmodell sei angemerkt, dass die *idMethod* gleich der *ObjectID* ist. Nur im Fall von Methoden werden Daten in der Tabelle *MethodsData* gespeichert. Eine weitere Besonderheit stellen die Ergebnisse von Senken dar. Diese sind durch mehrere Zeilen in der Tabelle *Results* beschrieben. Realisiert ist dies durch die *idResultType*. Zwei Einträge, differenziert durch die *idResultType* sind jeder Senke zugeordnet.

Der Programmablauf der (teil-)automatischen Modellgenerierung im Simulationsinstrument Tecnomatix Plant Simulation ist in der Abbildung B.2 abgedruckt. Die Darstellungen der Programmablaufpläne in den Abbildungen B.2a und B.2b wurden mit der open-source Software *PapDesigner* realisiert [Fol-19]. Der Darstellung liegt die Norm DIN-66001 zugrunde [DIN-66001].

PSIMA_Controller übernimmt drei wichtige Funktionen. Dieses Programm prüft in regelmäßigen Abständen, ob in der Datenbank ein neues Projekt zur Simulation verfügbar ist. Wenn dem so ist, erzeugt es eine neue Instanz von PSIMA_Executer. Diese neue Instanz ist so gestaltet, dass sich diese nach Abschluss der Simulation automatisch beendet. Sollte PSIMA_Executer z.B. durch einen Syntaxfehler die Simulation mit einer Fehlermeldung unterbrechen, so wird diese Instanz durch PSIMA_Controller spätestens 30 Sekunden nach deren Start beendet. Fehlermeldungen können nämlich zur Folge haben, dass die Routine zum Schreiben der Ergebnisdaten in die Datenbank und Beenden der Instanz nicht ausgeführt werden kann. Hinsichtlich des Umgangs mit Modellfehlern lassen sich neben der oben beschriebenen Maßnahme weitere benennen:

- Methode *CheckLoop* in Instanz *PSIMA_Executer*: Nach automatischem Start der Simulation wird sekundlich (Realzeit) geprüft, ob sich die Simulationszeit im Vergleich zur vergangenen Sekunde (Realzeit) geändert hat. Trifft dies nicht zu, so wird die Instanz PSIMA_Executer beendet und eine Fehlermeldung in die Datenbank geschrieben. Diese Meldung kann der Benutzer in der App einsehen, um ggf. Änderungen am Modell vorzunehmen.
- Methode *setErrorHandler*: Diese Methode liefert der Softwarehersteller mit dem Simulationsinstrument Tecnomatix Plant Simulation mit. Eine Methode wird statt dem Methoden-Debugger ausgeführt, sobald ein Fehler in einer Methode auftritt. Die ausgeführte Methode stoppt die weitere Ausführung des Simulationslaufs und schreibt eine meist sehr präzise Fehlerbeschreibung in die Datenbank. Diese wird dem Lernenden in der App zugänglich gemacht.

**Legende:**

PS – Primärschlüssel/Fremdschlüssel

Abbildung B.1: Datenbankmodell der Cloud-IT-Architektur

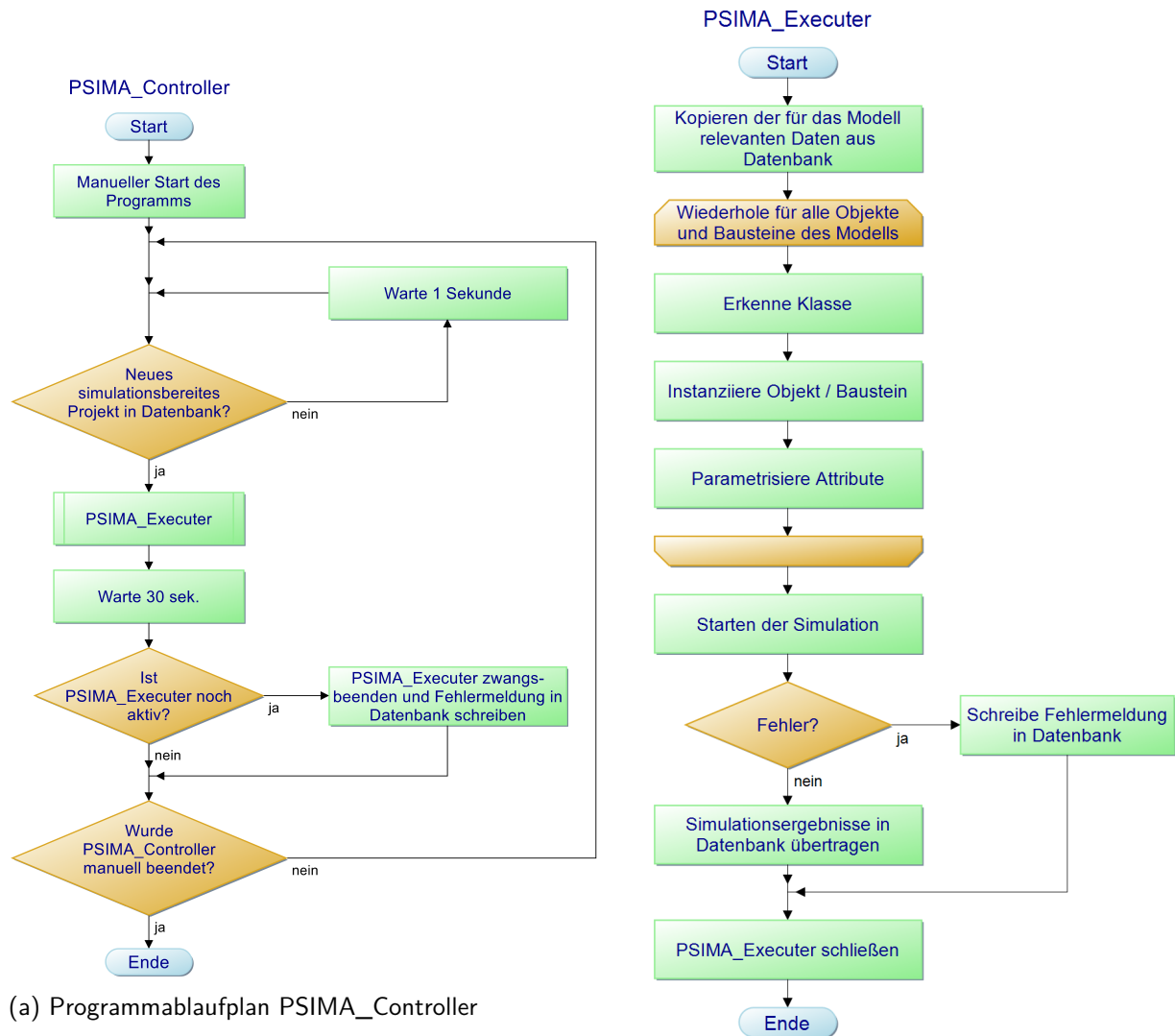


Abbildung B.2: Ablauf der (teil-)automatischen Modellgenerierung im Simulationsinstrument

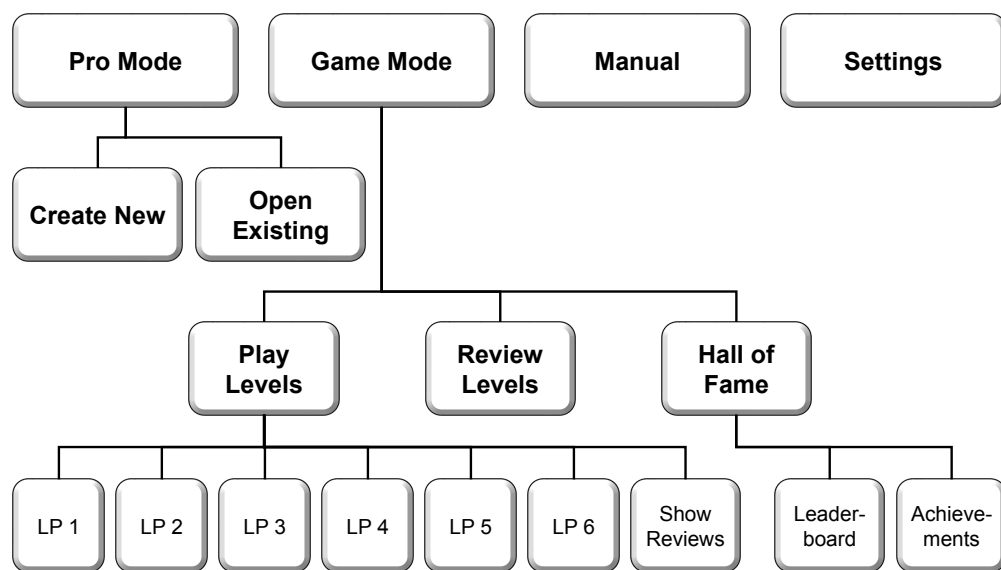
C Implementierung von PSIMA

Im Folgenden sind in Ergänzung zu Kapitel 5.1.2 die Menüführung, vertiefende Bildschirmfotos und die in die der App PSIMA implementierten Objekte und Bausteine beschrieben.

Die Menüführung der App ist in Abbildung C.1 gezeigt. In Ergänzung zu den in Abbildung 5.2 und 5.3 dargestellten Bildschirmfotos der App PSIMA sind in Abbildung C.2 vertiefende Bildschirmfotos hinsichtlich der Darstellung von Simulationsergebnisdaten (Abbildungen C.2a und C.2b) und Tutorials (Abbildungen C.2c und C.2d) dargestellt.

Abbildung C.2a zeigt exemplarisch den Ergebnisbericht für alle beweglichen Elemente eines Simulationsmodells. Im gezeigten Fall existierte im Modell nur ein bewegliches Element mit dem Namen *Entity*. Der Durchsatz sowie die durchschnittliche Durchlaufzeit sind ersichtlich. Abbildung C.2b zeigt eine Ressourcenstatistik am Beispiel einer *Parallelstation*. Konkret benannt ist in dieser Ansicht die Aufteilung des Status der Maschine in *arbeitend*, *wartend* und *blockiert*. Weiterhin sind die Anzahl beweglicher Elemente in der Parallelstation zum Ende der Simulation sowie die über die Laufzeit minimale, maximale sowie die kumulierte Anzahl der Ein- und Austritte von beweglichen Elementen ersichtlich. Um den Lernenden den Einstieg zu erleichtern, sind in der App Tutorials gestaltet worden. Eines enthält umfangreiche Erläuterungen über die Funktionsweisen der einzelnen Bausteine und Objekte, vgl. Abbildung C.2c. Ein zweites Tutorial ist zur Unterstützung der Einarbeitung in die VBPIs umgesetzt worden, vgl. Abbildung C.2d. Zudem findet der Lernende auch kurze Videos, die einen ersten Überblick über die Handhabung der App PSIMA geben.

In der Tabelle C.1 sind die in der App PSIMA implementierten Objekte und Bausteine kurz beschrieben [Sie-19]. In der Tabelle C.2 sind darüber hinaus die Symbole der Objekte/Bausteine ergänzt [Sie-19]. Hinsichtlich der Erweiterbarkeit dieser Modellwelt sei auf die Beschreibung der anwenderdefinierten Bausteine zur Multiagentensimulation im Kapitel 5.1.4 verwiesen.

**Legende:**

LP – Levelpack

Achievements – englisch für Abzeichen

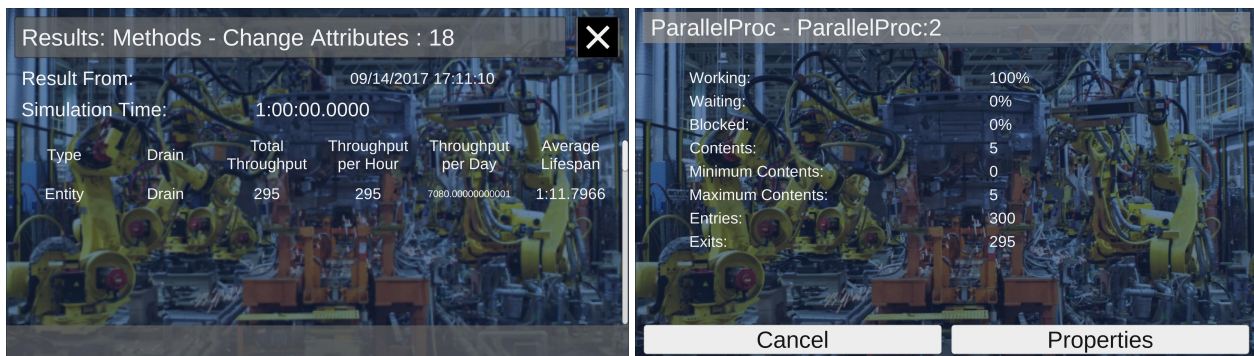


– Hierarchie der Menüpunkte/Wahloptionen für den Benutzer

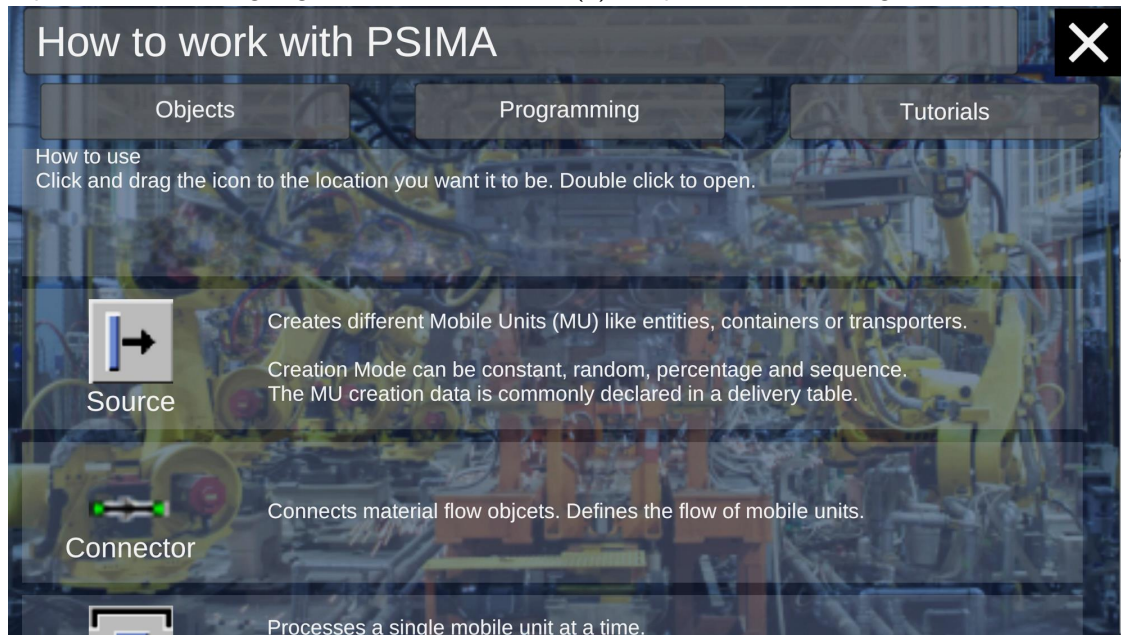


– Durch den Benutzer auszuwählender Menüpunkt

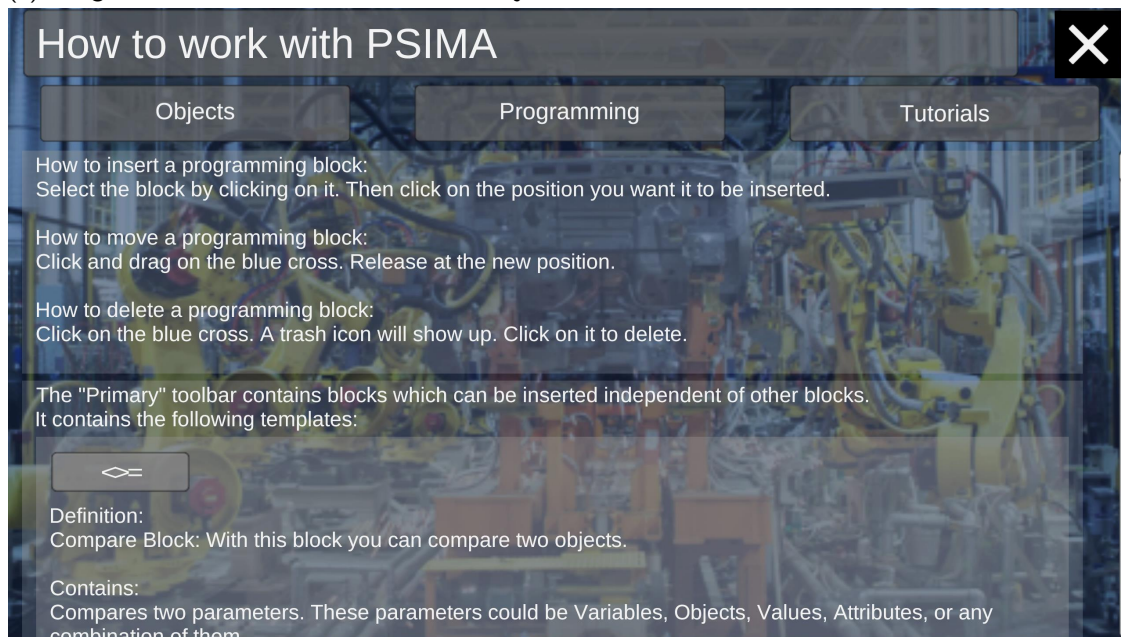
Abbildung C.1: Menüführung der App PSIMA



(a) Beispielhafte Darstellung Ergebnisse Simulationslauf (b) Beispielhafte Darstellung Ressourcenstatistik



(c) Integriertes Tutorial für Bausteine und Objekte



(d) Integriertes Tutorial für VBPIs

Abbildung C.2: Bildschirmfotos der prototypischen App: Vertiefende Darstellung

Tabelle C.1: In PSIMA implementierte Objekte und Bausteine: Kurzbeschreibung der Eigenschaften

Name [e/d]		Kurzbeschreibung der wichtigsten Eigenschaften nach [Sie-19]
Source Quelle	/	Die Quelle produziert bewegliche Elemente. Der Benutzer definiert hierzu Zwischenankunftszeiten. Bewegliche Elemente sind z.B. Werkstücke oder Transporter.
Connector Kante	/	Kanten verbinden Ausgänge mit den Eingängen von Nachfolgern. Es handelt sich um logische Verbinder. Die Simulationszeit wird bei einem Übergang von beweglichen Elementen nicht fortgeschrieben.
SingleProc Einzelstation	/	Die Einzelstation hat eine Kapazität von einem beweglichen Element. Die Zeit wird um die Bearbeitungszeit des jeweiligen Elements fortgeschrieben. Das ausgehende bewegliche Element ist das, welches auch in die Einzelstation eingegangen ist.
Drain Senke	/	Die Senke vernichtet die beweglichen Elemente.
ParallelProc Parallelstation	/	Die Parallelstation bildet die Logik mehrerer parallel angeordneter Einzelstationen ab. Benutzer können die Anzahl paralleler Stationen festlegen.
Assembly Montagestation	/	Die Montagestation bildet einen Lade- oder Montageprozess ab. Beim Laden werden bewegliche Elemente auf einen Werkstückträger umgesetzt. Beim Montieren verschmelzen bewegliche Elemente miteinander und ein neues Element anderer Bezeichnung wird abhängig von der Stückliste erzeugt.
Dismantle- Station Demontagestation	/	Bewegliche Elemente, die auf einen Werkstückträger geladen wurden, können in der Demontagestation wieder abgeladen werden. Bei der Demontage können neue bewegliche Elemente anderer Bezeichnung - abhängig von der Stückliste - erzeugt werden.
Store / Lager Buffer Puffer	/	Das Lager kann eine beliebige Anzahl beweglicher Elemente speichern. Ein Puffer kann temporär bewegliche Elemente speichern. Bei Störung der vor- oder nachgelagerten Stationen wirkt der Puffer entkoppelnd.
Line Förderstrecke	/	Die Förderstrecke dient der Modellierung von Fördersystemen. Bewegliche Elemente, die keinen eigenen Antrieb aufweisen wie Paletten, werden über eine vom Benutzer zu definierende Länge und Geschwindigkeit transportiert.
Track Weg	/	Der Weg eignet sich zur Darstellung von Transportstrecken. Bewegliche Elemente mit eigenem Antrieb wie Fahrzeuge, können auf dem Weg mit einer zu definierenden Geschwindigkeit fahren.
Method Methode	/	Die Methode dient der Erstellung individueller Steuerungen. Diese können vom Benutzer oder von Bausteinen und Objekten aufgerufen werden. Methoden werden in der Sprache SimTalk verfasst. SimTalk erlaubt den flexiblen Zugriff auf Attribute von Objekten und Bausteinen und bietet durch den Hersteller vorbereitete Funktionalitäten, die eine schnelle Erstellung von Steuerungen erlauben.
Table Tabelle	/	Die Tabelle ist als Liste mit mehreren Spalten definiert. Der Zugriff erfolgt über den System-Index, z.B. [1, 2], oder über einen benutzerdefinierten Index, z.B. ["Spalte eins", "Zeile zwei"]. Die Tabelle kann aus Zeilen/Spalten unterschiedlicher Datentypen bestehen.
Area		Eine Area ist ein anwenderdefinierter Baustein zur Abbildung eines Agentensystems. Das Durchlaufen einer Area erfordert eine vom Anwender zu definierende Zeit. Jede Area ermöglicht dem beweglichen Element einen Übergang in benachbarte Areas. Zu einem Zeitpunkt kann sich maximal ein bewegliches Element in einer Area befinden. Mit Areas lässt sich im Simulationsmodell ein Layout beschreiben. (Stand: partiell umgesetzt)
AGVLoading		AGVLoading ist ein anwenderdefinierter Baustein zur Abbildung eines Agentensystems. Er dient dem Beladen von fahrerlosen Transportfahrzeugen. (Stand: partiell umgesetzt)
AGVUnloading		Wie AGVLoading nur zum Entladen. (Stand: partiell umgesetzt)

Tabelle C.2: In PSIMA implementierte Objekte und Bausteine: Symbole

Name [e/d]	Symbol - eigene Darstellung nach [Sie-19]
Source Quelle	/ 
Connector Kante	/ 
SingleProc Einzelstation	/ 
Drain Senke	/ 
ParallelProc Parallelstation	/ 
Assembly Montagestation	/ 
Dismantle-Station Demontagestation	/ 
Store Lager	/ 
Buffer Puffer	/ 
Line Förderstrecke	/ 
Track Weg	/ 
Method Methode	/ 
Table Tabelle	/ 
Area	
AGVLoading	
AGVUnloading	

D Kurzbeschreibung der exemplarischen Level von PSIMA

Im Folgenden sind die im Kapitel 5.1.3 erwähnten Levelpacks zusammenfassend dargestellt. Der Aufbau der Levelpacks richtet sich nach den Ausführungen im Kapitel 4.1.2.1. Es wurden 37 Level exemplarisch konzipiert und im Simulationsinstrument getestet. Die IKS dienen der Einarbeitung des Lernenden in den grafisch-interaktiven Modellaufbau unter Nutzung von Bausteinen, Objekten und Programmiersprache zur Implementierung von Simulationsmodellen im Kontext von Materialflusssystemen. In der App PSIMA sind 29 Level in vier Levelpacks umgesetzt worden. In Tabelle D.1 sind die Level aus Levelpack 1 (Schwerpunkt Materialfluss) kurz zusammengefasst. Analog hierzu sind die Level mit dem Schwerpunkt auf den Informationsfluss aus Levelpack 2 in Tabelle D.2 beschrieben. Tabelle D.3 zeigt die Level unter Verwendung anwenderdefinierter Bausteine am Beispiel der Multiagentensimulation und Tabelle D.4 vermittelt einen zusammenfassenden Eindruck des Story-Modus aus Levelpack 5. Die Tabellen beinhalten neben der Nummer des Levels und dem Titel, das jeweilige Ziel inkl. einer Kurzbeschreibung. Wenn der Lernende zur Lösung einer Aufgabe bislang unbekannte Bausteine/Objekte der grafisch-interaktiven Modellierung oder Blöcke zur Programmierung nutzen muss, so sind die jeweiligen Titel einmalig in der nachfolgenden Spalte benannt. Die Begriffe der Bausteine/Objekte beziehen sich auf die Tabelle C.1. Die Begriffe der Blöcke finden sich in Tabelle 5.5 wieder. Schließlich ist der Stand der Umsetzung in PSIMA angegeben.

Um ein Level erfolgreich abzuschließen muss der Lernende ein Simulationsmodell gemäß einer Aufgabenstellung errichten, einen Simulationslauf starten und die Ergebnisse einsehen. Während die Level der ersten Levelpacks als unabhängige Aufgaben gestaltet sind, zieht sich eine Geschichte durch die Level aus Levelpack 5. Es liegt ein Beispiel eines fiktiven Automobilzulieferers zugrunde. An diesem Beispiel wird die Breite der Anwendbarkeit der ereignisdiskreten Simulation aufgezeigt. Die Aufgabenstellungen sind in den Bereichen Getriebebau, Wartezeit in der Kantine zur Mittagszeit, Insourcing der Montage von Rückspiegeln, Bewertung einer „Build-to-Order“-Bereitstellung von Sitzen und des Warenausgangs entwickelt worden.

Bei der tabellarischen Darstellung im Folgenden handelt es sich um eine Zusammenfassung der entwickelten Level. Die detaillierte Beschreibung jedes Levels ist online auf www.lernzeuge.de abrufbar [Sch-19]. Es sei angemerkt, dass die detaillierten Aufgabenstellungen informell und das textuelle Feedback bewusst lobend formuliert sind. Dies soll dazu beitragen, dass der Studierende den Umgang mit PSIMA als angenehm wahrnimmt.

Tabelle D.1: PSIMA IKS Levelpack 1 - Materialfluss nach [Sch-19]

Level	Titel	Ziel/Kurzbeschreibung	Neue Bausteine/Blöcke	Objekte/	Umsetzung
1-1	Single Process	Instanziierung von Bausteinen/Objekten durch Ziehen und Ablegen, Verbinden von Quelle, Einzelstation und Senke durch Kanten, Starten eines Simulationslaufs.	Quelle, Einzelstation, Senke, Kante		umgesetzt
1-2	Single Process - Serial	Ändern von Attributen von Bausteinen/Objekten: Umbenennung, Manipulation der Bearbeitungszeit.			umgesetzt
1-3	Parallel Process Station	Funktionen und Attribute des Bausteins Parallelstation kennenlernen.	Parallelstation		umgesetzt
1-4	Methods - Change Attributes Assembly Station	Manipulation von Attributen von Objekten/Bausteinen durch Methode.	Main, Set, Object, Prop, Value		umgesetzt
1-5	Assembly Station	Funktionen und Attribute des Bausteins Montagestation kennenlernen.	Montagestation		umgesetzt
1-6	Dismantle Station	Funktionen und Attribute des Bausteins Demontagestation kennenlernen. Wissen über Kanten (Vorgänger, Nachfolger) erweitern.	Demontagestation		umgesetzt
1-7	Buffer	Funktionen und Attribute des Puffers kennenlernen. Stochastische Zeit zur Bearbeitung an einer Maschine implementieren.	Puffer		umgesetzt
1-8	Line	Funktionen und Attribute des Bausteins Förderstrecke kennenlernen.	Förderstrecke		umgesetzt
1-9	Store	Funktionen und Attribute des Bausteins Lager kennenlernen.	Lager		umgesetzt
1-10	Method	Nutzung der Methode zur Manipulation von Attributen von Objekten und Bausteinen.			umgesetzt
1-11	Complex production line	Aufbau eines Simulationsmodells unter Nutzung der zuvor eingeführten Objekte und Bausteine.			umgesetzt

Tabelle D.2: PSIMA IKS Levelpack 2 - Informationsfluss nach [Sch-19]

Level	Titel	Ziel/Kurzbeschreibung	Neue Objekte/ Bausteine/Blöcke	Umsetzung
2-1	Designators	Adressieren von Objekten und Bausteinen mit den anonymen Bezeichnern „?“ und „@“.		umgesetzt
2-2	Random Numbers - Introduction	Einführung in If-Bedingung sowie die Nutzung von benutzerdefinierten Variablen. Zuweisen und Vergleichen von Werten.	If, If-Else, Var, Var- Ref, Com	umgesetzt
2-3	Random Numbers - Revisited	Vertiefende Aufgabe zur Nutzung der If-Bedingung auf Basis von Level 2-2.		umgesetzt
2-4	Random Numbers - Inspect Selection	Einführung in Inspect-Anweisung.	Inspect	umgesetzt
2-5	Waituntil	Einführung in Waituntil-Anweisung zur Unterbrechung der Ausführung einer Methode unter Nutzung des Blocks Free für freies Programmieren.	Free	umgesetzt
2-6	Stopuntil	Einführung in Stopuntil-Anweisung zur Unterbrechung der Ausführung einer Methode unter Nutzung des Blocks Free für freies Programmieren.		umgesetzt
2-7	Failures	Verfügbarkeit einer Maschine festlegen durch Manipulation der Attribute mittels Programmierung mit dem Block Free.		umgesetzt
2-8	Transporter	Funktionen und Attribute des Bausteins Weg und des beweglichen Elements Transporter kennenlernen. Manipulation von Attributen (sog. Sensoren) durch Programmierung mit dem Block Free. Erstmals mit einer Schleife arbeiten. Quellcode nur partiell in Aufgabenstellung abgedruckt.	Weg, For	umgesetzt
2-9	Delivery Table	Einführung in die Arbeit mit Tabellen. Zugriff auf Tabellen durch Programmierung. Nur textuelle Beschreibung, kein Quellcode in Aufgabenstellung abgedruckt.	Tabelle	umgesetzt
2-10	Delivery Table - Part 2	Vertiefende Arbeit mit Tabellen. Nur textuelle Beschreibung, kein Quellcode in Aufgabenstellung abgedruckt.		umgesetzt
2-11	The Airport	Aufbau eines Simulationsmodells am Beispiel einer Beschreibung von Abläufen an einem Flughafen unter Nutzung der zuvor eingeführten Objekte und Bausteine. Simulationsmodell schon teilweise vorbereitet.		umgesetzt

Tabelle D.3: PSIMA IKS Levelpack 4 - Anwenderdefinierte Bausteine nach [Sch-19]

Level	Titel	Ziel/Kurzbeschreibung	Neue Bausteine/Blöcke	Objekte/ Umsetzung
4-1	Working with Libraries - Multi-agent	Einführung in eine anwenderdefinierte Bausteinbibliothek zur Modellierung und Simulation von Agenten.	Area	partiell umgesetzt
4-2	Multiagent - colliding flows	Vertiefende Arbeit mit dem Baustein zur Modellierung und Simulation von Agenten am Beispiel sich kreuzender Materialflüsse.		partiell umgesetzt
4-3	Multiagent - layout planning	Vertiefende Arbeit mit dem Baustein zur Modellierung und Simulation von Agenten. Engpassermittlung.		partiell umgesetzt
4-4	Multiagent - AGV planning routing	Einführung in zwei anwenderdefinierte Bausteine zur Be- und Entladung von fahrerlosen Transportfahrzeugen.	AGVLoading, AGVUnloading	partiell umgesetzt
4-5	Multiagent - AGV planning	Vertiefende Arbeit mit der Bausteinbibliothek zur Modellierung und Simulation von Agenten. Ermittlung der Anzahl erforderlicher fahrerloser Transportfahrzeuge.		partiell umgesetzt

Tabelle D.4: PSIMA IKS Levelpack 5 - Story Modus nach [Sch-19]

Level	Titel	Ziel/Kurzbeschreibung	Umsetzung
5-1	Introduction Goods Arrival	- Einführung in den Story-Modus. Simulationsstudie (Parametervariation) ist am Beispiel des SCM im Automobilbau durchzuführen. Es gilt, den Engpass im Materialflusssystem zu identifizieren.	umgesetzt
5-2	Multi Order Picking	Simulationsstudie ist am Beispiel des Getriebebaus durchzuführen. Es gilt die Anzahl montierter Getriebe zu erhöhen.	umgesetzt
5-3	Second Source	Sensitivitätsanalyse zur Produktivitätssteigerung der Montagelinie für Getriebe ist durchzuführen. Besonderes Augenmerk ist auf das Adressieren von Attributen von Bausteinen und Objekten durch Methoden gelegt.	umgesetzt
5-4	Vendor Managed Inventory	Simulationsstudie ist am Beispiel des Inhouse-Consultings durchzuführen. Hierbei ist von einem lieferantengesteuerten Bestand an Zukaufteilen für Getriebe auszugehen. Programmierkenntnisse sollen weiter entwickelt werden.	umgesetzt
5-5	Tugger Train Milkrun	- Simulationsstudie ist am Beispiel einer Umplanung von einzelnen fahrerlosen Transportfahrzeugen hin zu einem Routenzug in der Produktion von Getrieben durchzuführen. Der Schwerpunkt liegt auf der Vertiefung von Programmierkenntnissen.	partiell umgesetzt
5-6	The Canteen	Simulationsstudie am Beispiel der Kantine zur Mittagszeit: Umplanung des Layouts, um mehr Mitarbeiter verpflegen zu können. Ziel: Programmierkenntnisse vertiefen.	umgesetzt
5-7	Mirrors	Simulationsstudie am Beispiel eines Szenarios zum „Insourcing“ der Montage von Rückspiegeln.	partiell umgesetzt
5-8	Seat Assembly	Simulationsstudie am Beispiel von Sitzen. Die Vorteilhaftigkeit eines „Build-to-Order“-Ansatzes für die Bereitstellung von Sitzen am Band soll durch den Lernenden bewertet werden.	partiell umgesetzt
5-9	Crossdocking serial	- Simulationsstudie am Beispiel des Warenausgangs des Automobilzulieferers.	umgesetzt
5-10	Crossdocking parallel	- Simulationsstudie am Beispiel des Warenausgangs des Automobilzulieferers.	umgesetzt

E Abzeichen in PSIMA

In der Tabelle E.1 ist eine Übersicht über die Abzeichen abgedruckt, die in PSIMA vorgesehen bzw. implementiert worden sind. Die Liste zeigt neben den Grafiken auch die Kriterien für die Verleihung des Abzeichens sowie den Stand der Umsetzung. Die Abbildung E.1 zeigt die Abzeichen in der App PSIMA, wie sie dem Lernenden in der *Hall of Fame* präsentiert werden. Schwarz abgebildet sind die noch nicht durch den Lernenden freigeschalteten Abzeichen. Die Verwendung des Spielelements Abzeichen ist in den Kapiteln 4.1.2.1, 5.1.2 und 5.1.3 ausführlich beschrieben.

Abzeichen machen in PSIMA über den Abschluss eines Levels hinaus Verdienste des Spielers für den Spieler und Mitspieler sichtbar. Abzeichen tragen zu einer hohen Lehr- und Lernproduktivität bei.

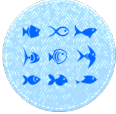
Tabelle E.1: Übersicht über die Abzeichen in PSIMA nach [Sch-19]

Grafik	Titel	Aufgabe/Kriterium	Umsetzung
	Curious	Take a look at the Hall of Fame.	umgesetzt
	Editor	Submit a level for Levelpack 4.	vorbereitet
	Employee Of The Year	Solve level 6 in Levelpack 3.	umgesetzt
	Finally	Finish a level on the fifth submit.	partiell umgesetzt
	First	Be the first to submit a valid solution to one level (in Levelpack 3).	vorbereitet
	First Day On The Job	Solve the first level in Levelpack 3.	umgesetzt
	Imperator	Be the first to complete Levelpack 3.	vorbereitet
	I've Got The Plan	Submit a solution after 10 minutes or earlier (in Levelpack 3 - submitted level must also be successfully completed).	vorbereitet

Fortsetzung von Tabelle E.1

Grafik	Titel	Aufgabe/Kriterium	Umsetzung
	Let's Get Started!	Solve the first level.	umgesetzt
	Rank 1 - Bronze	Finish Levelpack 1.	umgesetzt
	Rank 2 - Silver	Finish Levelpack 2.	umgesetzt
	Rank 3 - Gold	Finish Levelpack 3.	umgesetzt
	No Vacation	Submit solutions on five consecutive days (in Levelpack 3).	vorbereitet
	Perfectionist	Score 10 reviewer-points on a single level (in Levelpack 3).	vorbereitet
	Persistent	Open the app 10 times.	umgesetzt
	Reviewer	Submit your first review.	vorbereitet
	Room For Improvement	Score more reviewer-points at the next try on the same level (in Levelpack 3).	vorbereitet
	Start Over	Start a level from scratch by clicking the „Revert“ button (in Levelpack 3).	partiell umgesetzt
	Supervisor	Submit your 10th review.	vorbereitet
	Surgeon	Solve a level without reverting to the starting configuration (in Levelpack 3).	partiell umgesetzt

Fortsetzung von Tabelle E.1

Grafik	Titel	Aufgabe/Kriterium	Umsetzung
	Swarm Intelligence	All participants complete Levelpack 3.	umgesetzt
	Going The Whole Way	Complete one level of Levelpack 4.	vorbereitet
	United We Learn	All participants complete level 1 in Levelpack 3.	umgesetzt
	Veni Vidi Vici	Complete a level on the first submit (in Levelpack 3).	vorbereitet
	You Can Do Better	Score 5 or more reviewer-points.	partiell umgesetzt

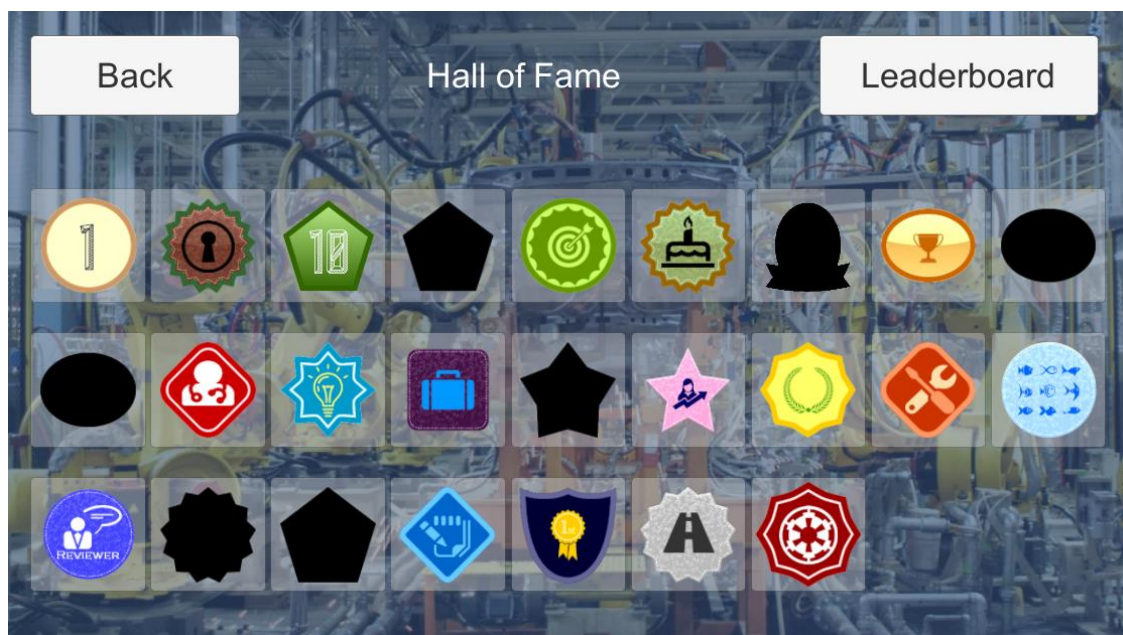


Abbildung E.1: Abzeichen in PSIMA

F Skript Umsetzung Multiagentensimulation

Der Quellcode F.1 dokumentiert die Entscheidungsfindung der Agenten in der anwenderdefinierten Bausteinbibliothek. Der Quellcode kommt im Simulationsinstrument Tecnomatix Plant Simulation zur Anwendung.

Quellcode F.1 Skript zur Entscheidungsfindung der Agenten

```
is
surrounding : table[object,object,object];
surroundingfree : table[object,object];
route : table[object];
i,j,k,l,m : integer;
nextdestinationaccordingtoPS : object;
randomnumber : integer;
blockingliste : table;

do
surrounding.create;
surroundingfree.create;
route.create;
surrounding[2,1] := current.exittop.succ;
surrounding[2,2] := ?;
surrounding[3,2] := current.exitrigh.succ;
surrounding[2,3] := current.exitbottom.succ;
surrounding[1,2] := current.exitleft.succ;
k := 1;
l := 1;
@.automatischezielfindung := false;
@.schliessedialog(true);
@.automatischezielfindung := true;
@.schliessedialog(true);
@.holeroute.kopiereInTabellenspalte(route,1);
for i := 1 to route.ydim loop
if route[1,i] = ? then
exitloop;
end;
next;
nextDestinationAccordingToPS := route[1,i+1];
if nextdestinationaccordingtoPS = void then debug; end;
if nextDestinationAccordingToPS.voll then
for i := 1 to surrounding.xdim loop
for j := 1 to surrounding.ydim loop
if surrounding[i,j] /= void then
if surrounding[i,j].voll = false then
if surrounding[i,j].class /= .materialfluss.senke then
surroundingfree[1,k] := surrounding[i,j];
k := k+1;
```

Fortsetzung von Quellcode F.1

```
else
end;
else
if surrounding[i,j] /= ? then
surroundingfree[2,1] := surrounding[i,j];
if surroundingfree[2,1].class = .materialfluss.senke then debug; end;
l := l + 1;
end;
end;
end;
next;
next;
if k = 1 then
?.fwblocklist(blockingliste);
if blockingliste.ydim > 0 then
for i := 1 to blockingliste.ydim loop
if blockingliste[1,1] = surroundingfree[2,i] then
surroundingfree.cutrow(m);
l := l - 1;
end;
next;
end;
randomnumber := z_uniform(1,1,1);
if surroundingfree[2,randomnumber].class = .materialfluss.senke then debug; end;
@.transfer(surroundingfree[2,randomnumber]);
else
randomnumber := z_uniform(1,1,k);
if surroundingfree[1,randomnumber].class = .materialfluss.senke then debug; end;
if @.transfer(surroundingfree[1,randomnumber]) = false then
end;
end;
else
@.transfer(nextdestinationaccordingtoPS);
end;
end;
```

G Skripte, Beispiel und Tabellen Umsetzung VBPIs

Neben der grafischen Oberfläche der VBPIs ist ein Programm zur Übersetzung der Sprachen zwischen PSIMA und Tecnomatix Plant Simulation erforderlich. Anforderungen an das im Folgenden **Compiler** genannte Programm sind, dass es über die Datenbank speicherbar ist, möglichst wenig Speicherplatz beansprucht und die Syntax sowie Semantik korrekt übermittelt. Im Datenbankmodell ist zum Zweck der Speicherung eine Tabelle mit dem Namen *MethodsData* verfügbar, vgl. Abbildung B.1 im Anhang B. Die Spalten *idMethod*, *idUser*, *idProjects* und *MethodName* dienen der eindeutigen Zuordnung des Blocks zur entsprechenden Methode. Besteht eine Methode aus mehreren Blöcken, so wird eine neue Zeile in der Tabelle erzeugt. Diese unterscheidet sich durch den Parameter *idBlock* eindeutig. Weiterhin ist der jeweilige Block-Typ im späteren Verlauf für den Compiler von Relevanz. Tabelle 5.5 zeigt für die in PSIMA prototypisch implementierten Blöcke neben einer Kurzbeschreibung das Kürzel, das zur Unterscheidung im Compiler genutzt wird.

Wie zuvor beschrieben können dem Nutzer bereitgestellte Blöcke ausschließlich zwischen dem *is* und *do*, oder dem *do* und *end*; eingefügt werden. Ausgenommen hiervon ist der Block *Main*. Dieser stellt die Grund-Struktur automatisch bereit. Er besteht aus vier Teilen:

Übergabe-Parameter: Werden Methoden aufgerufen, an die Parameter übergeben werden, so muss dies in der aufgerufenen Methode parametrisiert werden. Anders ausgedrückt müssen für übergebene Parameter in der Methode die Anzahl sowie der jeweilige Datentyp spezifiziert sein.

Rückgabe-Datentyp: Analog zum Übergabe-Parameter kann ein Rückgabe-Datentyp definiert werden. Es wird davon ausgegangen, dass es nur einen Parameter zur Rückgabe gibt.

Is: Bereich, der zur Deklaration von Variablen vorgesehen ist.

Do: Bereich, der für das Programm vorgesehen ist.

Die generische Struktur des Blocks *Main* ist wie folgt definiert:

Main – < Übergabe-Parameter > – < Rückgabe-Datentyp > – < Is > – < Do >

Während die Übergabe-Parameter direkt in diesen Block *Main* geschrieben werden, sind entsprechende Blöcke im Bereich des *Is* und *Do* durch die *idBlock* referenziert. Ein Beispiel kann so aussehen:

Main – *x, y : real* – *boolean* – *B3* – *B5*

Alle weiteren Blöcke sind durch folgende Syntax definiert:

$\langle \text{Block-Typ} \rangle - \langle \text{Wert oder Bx} \rangle - \langle \text{Wert oder By} \rangle - \dots$
 wobei der Block-Typ eines der in Tabelle 5.5 dargestellten Kürzel wie *If*,
 ein durch Benutzer eingegebener oder ausgewählter alphanumerischer Wert und
 Bx, By, ... eine Referenz auf die *idBlock* x, y, ... darstellen. Ein konkretes Beispiel lautet:
 $\text{Set} - B1 - B4$

Da jeder Block eine andere Struktur hinsichtlich der Anzahl und Art der erforderlichen Argumente hat, liegt dem Compiler die generische Struktur jedes Blocks zugrunde. Diese ist in der Tabelle G.2 partiell abgedruckt. Anhand dieser Tabelle weiß der Compiler, wie viele Argumente welcher Art an welcher Stelle zu erwarten sind.

Im Folgenden ist anhand eines konkreten **Minimal-Beispiels** die Arbeitsweise des Compilers erläutert. Der Compiler hat die Aufgabe, die VBPIs der App PSIMA in Programmiersprache zur Implementierung von Simulationsmodellen zu übersetzen. Die dem Beispiel zugrundeliegende Programmiersprache zur Implementierung von Simulationsmodellen ist *SimTalk* des objektorientierten Simulationsinstruments Tecnomatix Plant Simulation. Die Erklärung bezieht sich auf die Ausführungen im Kapitel 5.1.6. Ausgangspunkt ist das in der Abbildung G.1 gezeigte Beispiel. Dieser Code ist - wie aus dem Bild ersichtlich - in der App PSIMA erstellt worden. Die der Methode zugeordneten Einträge in der Datenbank sind in der Tabelle G.1 extrahiert.

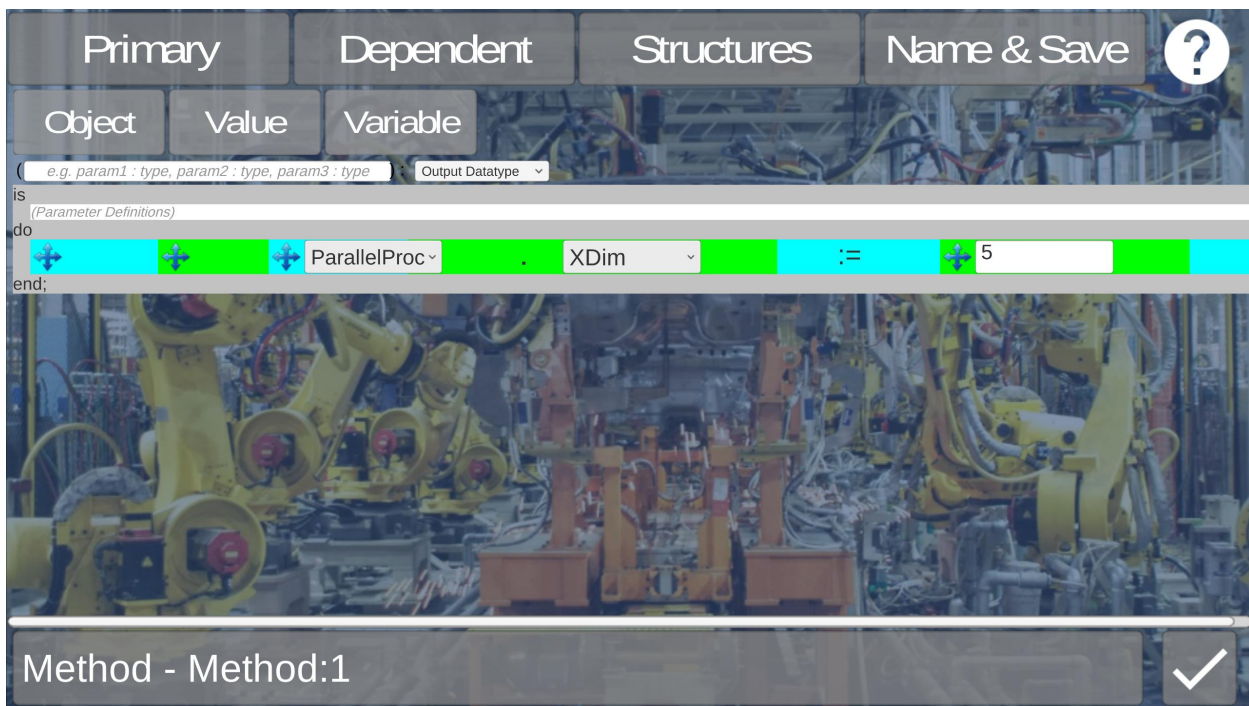


Abbildung G.1: VBPIs Minimalbeispiel Compiler: Block-Editor in der App PSIMA

Im ersten Schritt wird der Block *Main* von einem Algorithmus eingelesen. Da dieser Block wie zuvor beschrieben immer aus vier Teilen besteht, findet sich diese Logik durch vier trennende Bindestriche „-“

Tabelle G.1: VBPIs Minimalbeispiel Compiler: Auszug aus der Datenbank (Tabelle *MethodsData*) für das in Abbildung G.1 gezeigte Programm

BlockType	idBlock	Content
Prop	1	Prop-B2-XDim
Ref	2	Ref-ParallelProc
Set	3	Set-B1-B4
Value	4	Value-5
Main	5	Main- -Output Datatype- -B3

in der Tabelle wieder. Klar wird, dass in diesem Beispiel weder Parameter an die Methode übergeben werden noch diese einen Parameter zurückgibt. Auch eine Variable wird nicht deklariert. Der Block *B3* lässt sich dem *Do*-Teil des Programms zuordnen. Es handelt sich um den Block mit der *idBlock* 3. Dieser wiederum ist vom Typ *Set* und repräsentiert also einen Block zur Zuweisung von Werten, vgl. Tabelle 5.5. Folglich muss dieser Zuweisung eine linke und rechte Seite, getrennt durch $:=$, zugrunde liegen. Tabelle G.2 zeigt diese generische Struktur für jeden Block. Die linke Seite verweist gemäß der Tabelle G.1 auf die *idBlock* 1, die rechte auf die *idBlock* 4. *idBlock* 1 ist vom Typ *Prop* und soll den Zugriff auf Attribute erlauben. Um herauszufinden, von welchem Objekt das Attribut gewählt wird, ist Block 2, der auf ein Element *ParallelProc* referenziert, verschachtelt. Somit ist nun klar, dass das Attribut *XDim* (=Dimension/Anzahl Plätze) des Elements *ParallelProc* vom Nutzer adressiert wurde. Die rechte Seite der Zuweisung referenziert auf die *idBlock* 4. Dieser Block ist vom Typen *Value* und beinhaltet den Wert 5. Der beschriebene Prozess wird in der App sowie umgekehrt im Simulationsinstrument Tecnomatix Plant Simulation ausgeführt.

Kurz zusammengefasst werden ausgehend von der in der App implementierten VBPIs hierarchische Einträge in die Datenbank übernommen. Diese Daten werden an das Simulationsinstrument übergeben, welches die Einträge in der Tabelle in *SimTalk* übersetzt. Die beschriebene Systematik ließe sich auch auf weitere objektorientierte Programmiersprachen zur Implementierung von Simulationsmodellen übertragen.

Die Übersetzung gestaltet sich aus Sicht des Simulationsinstruments gemäß des in Tabelle G.1 eingeführten Beispiels wie folgt. Bei dem ersten Block in der Liste handelt es sich um einen vom Typen *Prop*. Aus der Tabelle G.2 ist ersichtlich, dass *Prop* (Zugriff auf Attribut eines Objekts) aus drei Elementen besteht: *#BBBB* (=Block), *„.“* (=Punkt im Format String) und *#FFFF* (=vom Benutzer eingegebener oder ausgewählter alphanumerischer Wert). Entsprechend ist in der Spalte „Content“, vgl. Tabelle G.1, der Block mit der *id* 2 referenziert. Die zweite Zelle liefert einen *„.“*, der nachfolgend an den Rückgabewert des Inhalts von *idBlock* 2 gesetzt wird. Der dritte Wert *#FFFF* wird mit dem in der Datenbank mitgelieferten String „XDim“ ersetzt. Dieses Verfahren wird nun Zeile für Zeile so ausgeführt. Dabei ergeben sich die folgenden Zwischenschritte:

- B1 entspricht {B2}.XDim
- B2 entspricht ParallelProc
- B3 entspricht {B1}:= {B4};
- B4 entspricht 5
- B5 entspricht
Is
Do {B3}
end;

Miteinander kombiniert ergibt sich die im Quellcode G.1 gezeigte Lösung.

Quellcode G.1 VBPIs Minimalbeispiel Compiler: SimTalk

is

do

ParallelProc.Xdim := 5;

end;

Tabelle G.2: Ausschnitt der generischen Struktur implementierter Blöcke zur Übersetzung der VBPIs in die Sprache SimTalk

If	IfElse	While	For	Prop	Compare	Var	Ref	Value	Com	Set	Call	Return	VarRef
If	If	While	For	#BBBB	#BBBB	#FFFF	#FFFF	#FFFF	-	#BBBB	#BBBB	return	#FFFF
#BBBB	#BBBB	#BBBB	#BBBB	.	#FFFF	:			#CCCC	:=	;	#BBBB	
Then	Then	loop	to	#FFFF	#BBBB	#FFFF			#BRKL	#BBBB		;	
#BRKL	#BRKL	#BRKL	#BBBB			;				;	#BRKL		
#BBBB	#BBBB	#BBBB	loop			#BRKL							
#BRKL	#BRKL	#BRKL	#BRKL										
end;	Else	end;	#BBBB										
#BRKL	#BRKL	#BRKL	#BRKL										
	#BBBB		next;										
	#BRKL		#BRKL										
	end;												
	#BRKL												

Legende:

- #BBBB – An dieser Stelle steht ein anderer Block.
 - #BRKL – „Breakline“ = Zeilenumbruch.
 - #FFFF – Freitext, der vom Benutzer eingegeben oder ausgewählt wurde. Dieser ist direkt in das Programm zu übernehmen.
 - #CCCC – Freitext, der auskommentiert ist (hat keine Relevanz für das Programm).
- Schlüsselworte – z.B. If oder While ohne #: Schlüsselworte, die im Programm stehen.

H Skripte und Tabellen Umsetzung Animation

Die Abbildung H.1 zeigt Bildschirmfotos der Tabelle zur Animation. Diese Tabellen wurden durch das Simulationsinstrument generiert. Sie listen die für die Animation relevanten Ereignisse des ereignisdiskreten Simulationsinstruments an einem konkreten Beispiel namens *PaintShop*, vgl. die Ausführungen im Kapitel 5.1.7.

Der Quellcode H.1 dokumentiert das Skript zum Import von Daten aus der Datenbank in die Game-Engine Unity3D. Der Quellcode H.2 dokumentiert das Vorgehen zur Positionierung stationärer Grafiken in Unity3D. Schließlich zeigt der Quellcode H.3 das Skript zur Animation der temporären physischen Objekte. Abbildung H.2 zeigt exemplarisch das Ergebnis der Umsetzung der Animation. Ersichtlich ist eine stereoskopische dreidimensionale Animation. Diese kann auf handelsüblichen Smartphones und einer kostengünstigen Brille zur stereoskopischen Projektion für den Lernenden immersiv verfügbar gemacht werden.

idUsers	idProjects	ID	Class	Type	Name	MuNumber	Xpos	Ypos	Zpos	RealTime	Length	Orientation	Time diff
2	1	1	Source	Machine	Source_parts	null	0	0	0	0	0	0	0
2	1	2	Buffer	Machine	Partsbuffer	null	4	0	0	0	0	0	0
2	1	3	Assembly	Machine	Hanging	null	8	0	0	0	0	0	0
2	1	4	SingleProc	Machine	Pretreatment	null	12	0	0	0	0	0	0
2	1	5	SingleProc	Machine	Painting	null	16	0	0	0	0	0	0
2	1	6	Buffer	Machine	Drying	null	20	0	0	0	0	0	0
2	1	7	DismantleS...	Machine	Take_down	null	24	0	0	0	0	0	0
2	1	8	Source	Machine	Source_han...	null	28	0	6	0	0	180	0
2	1	9	Buffer	Machine	Hangingbu...	null	8	0	6	0	0	270	0
2	1	10	Store	Machine	Store	null	28	0	0	0	0	0	0
2	1	11	Conveyor	Line	Meet point	null	24	0	6	0	0	0	0

(a) Position und Orientierung der stationären Bausteine und permanenten Objekte

idUsers	idProjects	ID	Class	Type	Name	MuNumber	Xpos	Ypos	Zpos	RealTime	Length	Orientation	Time diff
2	1	12	Entity	MU	Entity	311513	0	0	0	0	0	0	0
2	1	13	Entity	MU	Entity	311513	0	0	0	60	0	0	60
2	1	14	Entity	MU	Entity	311513	4	0	0	62	0	0	2
2	1	15	Entity	MU	Entity	311513	4	0	0	62	0	0	0
2	1	16	Entity	MU	Entity	311513	8	0	0	64	0	0	2
2	1	17	Entity	MU	Entity	311513	8	0	0	124	0	0	60
2	1	18	Entity	MU	Entity	311513	12	0	0	126	0	0	2
2	1	19	Entity	MU	Entity	311513	12	0	0	186	0	0	60
2	1	20	Entity	MU	Entity	311513	16	0	0	188	0	0	2
2	1	21	Entity	MU	Entity	311513	16	0	0	248	0	0	60
2	1	22	Entity	MU	Entity	311513	20	0	0	250	0	0	2
2	1	23	Entity	MU	Entity	311513	20	0	0	250	0	0	0
2	1	24	Entity	MU	Entity	311513	24	0	0	252	0	0	2

(b) Positionen eines exemplarischen temporären physischen Objekts über die Simulationszeit

Abbildung H.1: Bildschirmfoto der Tabelle zur Animation generiert durch das Simulationsinstrument

Quellcode H.1 Skript zum Import von Daten zur Animation in Unity3D

```
SQLControl Manager = (SQLControl)GameObject.Find("SQLManager").  
    GetComponent("SQLControl");  
string DataType = "3DData";  
int projectId = 1;  
string Condition = "MuNumber = \"null\"";  
  
ArrayList Data = Manager.getProjectData(DataType, projectId, Condition);
```

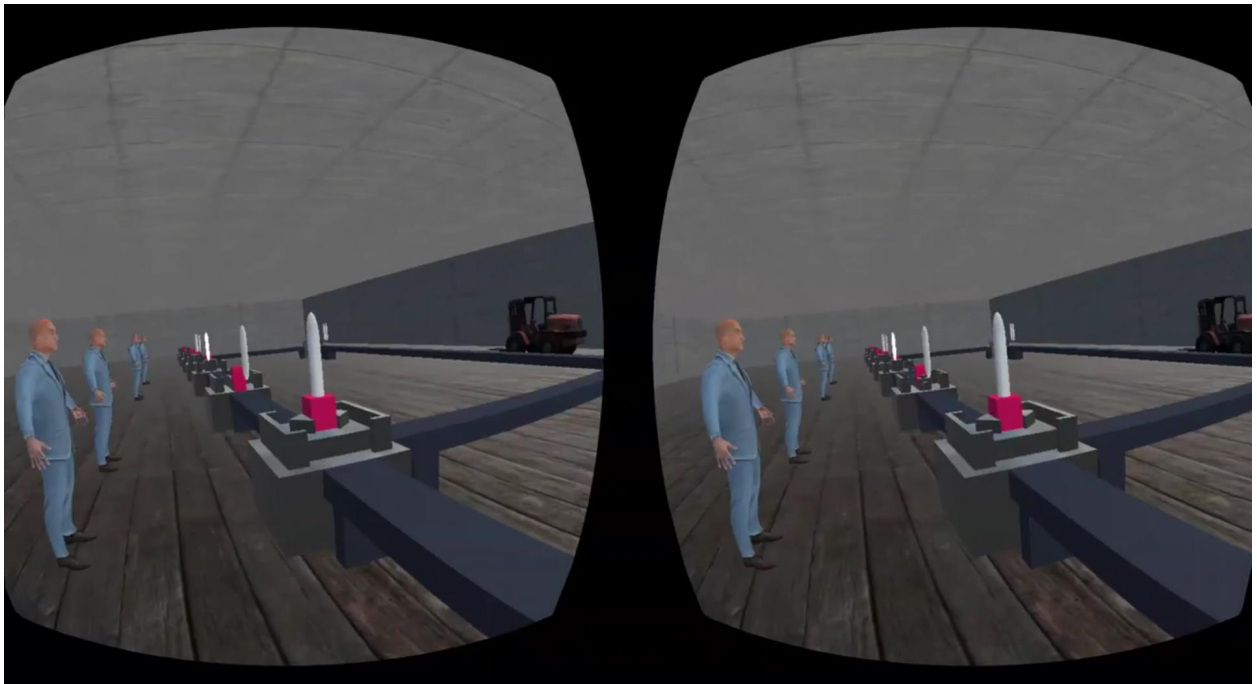


Abbildung H.2: Bildschirmfoto der stereoskopischen dreidimensionalen Animation: In blauer Farbe sind Fördereinrichtungen, in grauer Farbe Bearbeitungsstationen, in roter Farbe Werkstückträger und in weißer Farbe Produkte gezeigt

Quellcode H.2 Skript zur Positionierung stationärer Grafiken in Unity3D

```

for (int i = 1; i < Data.Count; i++)
{
    ArrayList temp = (ArrayList)Data[i];
    long xPos  = (long)temp[7];
    long yPos  = (long)temp[8];
    long zPos  = (long)temp[9];
    string Class = (string)temp[3];
    long Orientation = (long)temp[12];
    if (Class == "Source")
    {
        GameObject Source = GameObject.Instantiate(Resources.Load<GameObject>("Source"))
            as GameObject;
        Source.transform.position = new Vector3(xPos * 10, yPos * 10, zPos * 10);
        Source.transform.eulerAngles = new Vector3(0, Orientation, 0);
    }
    else if (Class == "Buffer")
    {
        GameObject Buffer = GameObject.Instantiate(Resources.Load<GameObject>("Buffer Part"))
            as GameObject;
        Buffer.transform.position = new Vector3(xPos * 10, yPos * 10, zPos * 10);
        Buffer.transform.eulerAngles = new Vector3(0, Orientation, 0);
    }
    else if (Class == "Assembly")
    {
        GameObject Assembly = GameObject.Instantiate(Resources.Load<GameObject>("Assembly"))
            as GameObject;
        Assembly.transform.position = new Vector3(xPos * 10, yPos * 10, zPos * 10);
        Assembly.transform.eulerAngles = new Vector3(0, Orientation, 0);
    }
    else if (Class == "SingleProc")
    {
        GameObject SingleProc = GameObject.Instantiate(Resources.Load<GameObject>("SingleProc"))
            as GameObject;
        SingleProc.transform.position = new Vector3(xPos * 10, yPos * 10, zPos * 10);
        SingleProc.transform.eulerAngles = new Vector3(0, Orientation, 0);
    }
    else if (Class == "DismantleStation")
    {
        GameObject DismantleStation = GameObject.Instantiate(Resources.Load<GameObject>
            ("DismantleStation")) as GameObject;
        DismantleStation.transform.position = new Vector3(xPos * 10, yPos * 10, zPos * 10);
        DismantleStation.transform.eulerAngles = new Vector3(0, Orientation, 0);
    }
    else if (Class == "Store")
    {
        GameObject Store = GameObject.Instantiate(Resources.Load<GameObject>("Store"))
            as GameObject;
        Store.transform.position = new Vector3(xPos * 10, yPos * 10, zPos * 10);
        Store.transform.eulerAngles = new Vector3(0, Orientation, 0);
    }
}

```

Quellcode H.3 Skript zur Animation

```
for (int i = 1; i < Data.Count; i++)
{
    ArrayList temp = (ArrayList)Data[i];
    long xPos = (long)temp[7];
    long yPos = (long)temp[8];
    long zPos = (long)temp[9];
    long zeit = (long)temp[13];
    float time = zeit/GlobalVariable;
    iTween.MoveTo(gameObject, iTween.Hash("position", new Vector3
        (xPos * 10, yPos + 5, zPos * 10),
        "time", time, "easetype", iTween.EaseType.linear));
    yield return new WaitForSeconds(time);
}
```

I Ablauf der Montage in der Lernfabrik

Die Tabelle I.1 detailliert die Arbeitsabläufe je OP in der Lernfabrik an der VGU. Die Tabelle I.2 fasst den Ablauf der Erprobung des Vorgehens zum EEDS kurz zusammen. Es sei auf die Ausführungen im Kapitel 5.2.1 verwiesen. Der Studierende lernt gemeinsam mit anderen Studierenden im Sinne einer Kleingruppe von ca. fünf Personen. Studierendengruppen konkurrieren gegeneinander.

Tabelle I.1: Montageablauf in der Lernfabrik der VGU

OP	Arbeitsinhalte
OP 1	- Der Werker entscheidet, welche Variante montiert wird. - Entsprechend wird Rahmen 1 oder 2 auf den Werkstückträger aufgelegt. - Je nach Rahmen-Typ werden zwei oder vier Rollenhalter auf den Rahmen gelegt. - Weitergabe des Werkstückträgers an OP 2.
OP 2	- Werkstückträger wird von OP 1 in Empfang genommen. - Ansetzen von vier Muttern und Festschrauben eines Rollenhalters mittels Elektroschrauber. - Ein Rad an Rollenhalter mit Achse befestigen. - Ansetzen einer Mutter an Achse und mittels Elektroschrauber festschrauben. - Weitergabe des Werkstückträgers an OP 3.
OP 3	- Werkstückträger wird von OP 2 in Empfang genommen. - Ansetzen von vier Muttern und Festschrauben eines Rollenhalters mittels Elektroschrauber. - Ein Rad an Rollenhalter durch Einschieben einer Achse befestigen. - Ansetzen einer Mutter an Achse und mittels Elektroschrauber festschrauben. - Weitergabe des Werkstückträgers an OP 4.
OP 4	- Werkstückträger wird von OP 3 in Empfang genommen. - Ansetzen von vier Muttern und Festschrauben mittels Elektroschrauber (optional). - Ein Rad an Rollenhalter durch Einschieben einer Achse befestigen. - Ansetzen einer Mutter an Achse und mittels Elektroschrauber festschrauben. - Weitergabe des Werkstückträgers an OP 5.
OP 5	- Werkstückträger wird von OP 4 in Empfang genommen. - Ansetzen von vier Muttern und Festschrauben mittels Elektroschrauber (optional). - Ein Rad an Rollenhalter durch Einschieben einer Achse befestigen. - Ansetzen einer Mutter an Achse und mittels Elektroschrauber festschrauben. - Weitergabe des Werkstückträgers an OP 6.
OP 6	- Werkstückträger wird von OP 5 in Empfang genommen. - Ansetzen eines Etiketts mit zwei Schrauben und Befestigung mittels Elektroschrauber. - Weitergabe des Werkstückträgers an OP 7.
OP 7	- Werkstückträger wird von OP 6 in Empfang genommen. - Fertig montierter Trolley wird in einen Karton gelegt, der mit Klebeband verschlossen wird. - Karton wird in einer Lagerfläche für Fertigerzeugnisse abgelegt.

Tabelle I.2: Übersicht über den Ablauf der Übung in der Lernfabrik der VGU

Tag	Ablauf
1	Zufällig ausgewählte Studierende betreiben die Lernfabrik für 45 Minuten. Alle weiteren Studierenden sind für die Datenaufnahme durch Messung von Zeiten und Beobachtung zuständig.
2	Studierende dürfen die Fabrik nicht physisch betreiben. Studierende bilden ein valides und verifiziertes Simulationsmodell basierend auf den zuvor aufgenommenen Daten und führen Simulationsexperimente im Hinblick auf Profitabilitätssteigerungen durch.
3	Gleicher Inhalt wie an Tag zwei.
4	Jede Gruppe Studierender hat in der Lernfabrik eine Stunde Zeit: 30 Minuten für den Umbau der Fabrik gemäß der an den Tagen zwei und drei ermittelten Verbesserungen und 30 Minuten für den Betrieb der verbesserten Fabrik.
5	Präsentation der Ergebnisse im Hinblick auf simulativ ermittelte Verbesserungen, deren Umsetzung in der Lernfabrik und einen Abgleich zwischen simulativ erwartetem und aus dem Betrieb der Lernfabrik ermitteltem Ergebnis.

J Veröffentlichungen des Autors und Beiträge von Studierenden

In regelmäßigen Abständen sind während der Erarbeitung der Dissertation Teile vorveröffentlicht worden. Die im Folgenden benannten Publikationen (Auszug) sind in die vorliegende Promotionsschrift eingeflossen.

- [Sch-19]: Schumacher, B. C.; Balzar, F.; *Learnstrument zur Einarbeitung in ereignisdiskrete Simulationsinstrumente*; https://www.mf.tu-berlin.de/menue/forschung_und_industrie/kompetenzfelder/lernzeuge/learnstrument_zur_einarbeitung_in_ereignisdiskrete_simulationsinstrumente/ ; letzte Aktualisierung am 11.01.2019; zuletzt geprüft am 02.03.2019, 2019.
- [Mül-17a]: Müller, B. C.; Kühn, M.; Schele, M.; Seliger, G.; *Mobile Lernumgebung zur simulativen Gestaltung von Produktionssystemen*; in Wenzel, S.; Peter, T. (Hrsg.), *Simulation in Produktion und Logistik 2017*; kassel university press, Kassel, 2017; 345–354.
- [Mül-17b]: Müller, B. C.; Menn, J. P.; Seliger, G.; *Procedure for Experiential Learning to Conduct Material Flow Simulation Projects, Enabled by Learning Factories*; in *Procedia Manufacturing, 7th Conference on Learning Factories, CLF 2017*; 9, 2017:283–290.
- [Mül-15]: Müller, B. C.; Reise, C.; Seliger, G.; *Gamification in Factory Management Education – A Case Study with Lego Mindstorms*; in *Procedia CIRP, 12th Global Conference on Sustainable Manufacturing – Emerging Potentials*; 26, 2015:121–126.

Im Rahmen integrierter Forschungs- und Lehraktivitäten entstanden am Fachgebiet Montagetechnik und Fabrikbetrieb der TU Berlin studentische Arbeiten. Die Ergebnisse sind partiell in die vorliegende Arbeit eingeflossen. Der Autor der Dissertation war sowohl für die Definition der Aufgabenstellungen als auch die inhaltliche und wissenschaftliche Betreuung der Studierenden verantwortlich. Ein herzliches Dankeschön für das zum großen Teil sehr hohe Engagement gilt den im Folgenden benannten Studierenden.

- Al-Eryani, Shehabaddin: *PSIMA*. Projektarbeit, Sommersemester 2017.
- Balzar, Felix: *Mobile Lernumgebung zur Vermittlung der Anwendung von Ablaufsimulatoren*. Masterarbeit, 2017.
- Nguyen, Truong Trung: *Educational concept for the mediation of knowledge on material flow simulation*. Masterarbeit, 2016.
- Ho Phuong, Thao: *Learning environment for material flow simulation with Lego Mindstorms*. Masterarbeit, 2016.
- von Freyend, Maximilian John; Heinen, René; Sathyanarayana, Amogha: *Immersive 3D-Environment for mobile material flow simulation*. Projektarbeit, Sommersemester 2016.
- 30 Studierende des internationalen Studienprogramms Global Production Engineering and Management an der VGU: *Exercise Learning factory*. Projektarbeiten, Sommersemester, 2016.
- Sun, Xiaoyuan; Karandikar, Mrugank; Gosavi, Chaitanya; Sriraman, Sairam; Chougule, Koushal: *Cyber-physical learning environment for material flow simulation*. Projektarbeit, Sommersemester 2015.
- Al-Matari, Mohammed; Crevar, Danijel; Kühn, Markus; Schlüter, Marian: *Weiterentwicklung einer Simulations-App*. Projektarbeit, Wintersemester 2013.
- Ryter, Thomas; Fongern, Conrad; Alborz, Shahriar: *Entwicklung einer App für ein Android System zum Datenaustausch mit Simulationssoftware*. Projektarbeit, Sommersemester 2013.

Glossar

Abstraktion

Abstraktion bezeichnet ein „Verfahren zur Reduzierung der Komplexität eines Problems durch Trennung der für bestimmte Aspekte der Problemlösung wichtigen und unwichtigen Details des Problems“ [VDI-3633b, S.3]. Es lassen sich zwei Verfahren unterscheiden [VDI-3633b, S.3]:

Reduktion: Verzicht auf die Abbildung unwichtiger Details.

Idealisierung: Vereinfachung von realen Gegebenheiten.

Abzeichen

In →PSIMA machen Abzeichen über den Abschluss eines →Levels hinaus Verdienste des Spielers für den Spieler und Mitspieler sichtbar. Bei offenkundigen Abzeichen sind dem Spieler die Kriterien zur Verleihung bekannt. Im Gegensatz hierzu stehen die nicht offenkundigen Abzeichen. Ein Abzeichen ist ein →Spielement.

Agent

Ein Agent ist ein →Programm [Sch-06, S.752]. Ein Agent repräsentiert eine Umgebung, eine Organisation oder Gegenstände [TZI-19]. Agenten treffen Entscheidungen, handeln zielorientiert und in Eigeninitiative. Ein Agent kann seine Umgebung wahrnehmen und mit anderen Agenten kommunizieren. Agenten verhalten sich interaktiv und kooperativ zu anderen Agenten [Bra-18, S.132f.]. Siehe →Multiagentensimulation.

Alternate Reality Game

Alternate Reality Games sind eine Untergruppe von →Serious games. Das Ziel von Alternate Reality Games besteht darin, eine →Geschichte zu erzeugen, die in das reale Leben eindringt [Kim-09]. Die Grenzen zwischen Realität und →Geschichte können dabei miteinander verschmelzen, sodass Spieler vergessen können, dass sie spielen. Das Vorgehen zum →EEDS ist als Alternate Reality Game in dieser Arbeit gestaltet.

Animation

Der Begriff Animation beschreibt die „Erzeugung und Präsentation von Bildfolgen, in denen Änderungen [...] einen visuellen Effekt bedingen“ [VDI-3633b, S.3]. Beispiele für solch einen visuellen Effekt sind über die Zeit variierende Positionen sowie Änderungen von Formen und Farben [Gut-17, S.25] [VDI-3633b, S.21]. Die Animation dient der →Verifikation von →Simulationsmodellen. Die Animation fördert das Verständnis des Lernenden über das Modellverhalten.

Area

Englisch für Bereich. Eine Area ist ein anwenderdefinierter →Baustein zur Abbildung eines Multiagentensystems. Mobile Materialflussbausteine durchqueren eine Area in einer vom Anwender zu definierenden Zeit. Jede Area ermöglicht dem mobilen Materialflussbaustein einen Übergang in benachbarte Areas. Zu einem Zeitpunkt kann sich maximal ein mobiler Materialflussbaustein in einer Area befinden. Areas können nach Belieben vom Benutzer miteinander verbunden werden. So lässt sich im →Simulationsmodell ein Layout beschreiben.

Baustein

Ein Baustein ist ein Modellelement „mit einer eigenen internen Ablauflogik“ [VDI-3633a, S.26f.]. Bausteine sind funktionell abgrenzbare Elemente mit definiertem Inhalt und Umfang, die sich unter Beachtung von Regeln kombinieren lassen [Rud-16].

Ein vorkonfigurierter Baustein kann „durch Parametereingaben den konkreten Gegebenheiten des realen Systems angepasst werden“ [VDI-3633b, S.21].

Ein anwenderdefinierter Baustein ist ein vom „Anwender selbst erstellter Baustein“ [VDI-3633b, S.3].

Bausteinkasten

Sammlung von →Bausteinen [Rud-16].

Synonym: Baukastensystem.

Bausteinkonzept

Bei einem Bausteinkonzept handelt es sich um ein anwendungsbezogenes →Modellierungskonzept, da die →Beschreibungsmittel sich begrifflich an den abzubildenden Systemen der Anwendung orientieren [Wen-08b]. →Bausteine sind für einen bestimmten Anwendungsbereich topologisch, organisatorisch und/oder informatorisch zweckmäßig vordefiniert und aggregiert. Der Anwender kann diese parametrisieren.

Beschreibungsmittel

Beschreibungsmittel sind Werkzeuge für die Modellierung [Wie-87, S.31].

Block	Im Sinne der →VBPIS und der →visuellen blockbasierten Programmiersprache stellen Blöcke Programmelemente wie Funktionen oder Variablen dar [Kur-15].
Block-Bibliothek	→Blöcke werden dem Lernenden in einer Block-Bibliothek bereitgestellt. Diese →Blöcke lassen sich im →Block-Editor →instanzieren.
Block-Editor	→Blöcke werden im Block-Editor durch Lernende →instanziiert. →Programme lassen sich so erstellen. In →PSIMA ist ein Block-Editor am Beispiel einer →Programmiersprache zur Implementierung von Simulationsmodellen eines beispielhaften →Simulationsinstruments umgesetzt worden.
Cloud-IT-Architektur	Der Begriff Cloud-IT-Architektur beschreibt den →skalierbaren Aufbau der mobilen App →PSIMA, eines Webservers, einer Datenbank und eines →Simulationsinstruments. Die Anzahl mobiler Apps sowie die Anzahl der →Simulationsinstrumente sind nahezu beliebig veränderbar.
Compiler	Ein Compiler bezeichnet ein „Übersetzungsprogramm, das eine Programmiersprache A [...] in eine andere Sprache B [...] übersetzt“ [VDI-3633b, S.5]. Der in der Arbeit entwickelte Compiler übersetzt die →VBPIS in die →Programmiersprache zur Implementierung von Simulationsmodellen <i>SimTalk</i> des →Simulationsinstruments Tecnomatix Plant Simulation.
Cyber-Physical Production System Kurz: CPPS	CPPS beschreiben die Anwendung von Cyber-Physical Systems „in der produzierenden Industrie und somit die Befähigung zur durchgängigen Betrachtung von Produkt, Produktionsmittel und Produktionssystem unter Berücksichtigung sich ändernder und geänderter Prozesse“ [Kag-13, S.84]. Intelligente Produkte und Werkstückträger können mit Maschinen und Benutzern in ihrer Umgebung kommunizieren (siehe →Multiagentensimulation) [Bis-15, S.235].

Ereignisdiskrete Simulation

Kurzwort für zeitdiskret-ereignisorientierte →Simulation. Der Zeitfortschritt wird durch Ereignisse, Aktivitäten und Prozesse bestimmt [Wen-08b, S.73ff.]. Ein Ereignis ist grundsätzlich als atomar anzusehen und somit nicht weiter zerlegbar. Es verbraucht keine →Simulationszeit. Ein über das Ereignis hinaus andauernder Zustand behält im →Simulationsmodell bis zum nächsten Ereignis seine Gültigkeit.

Zusammengefasst lässt sich diese Prozedur als ein Springen von einem Ereigniszeitpunkt zum nächsten [Nas-14] beschreiben, wobei sich jeweils der Systemzustand ändert [Ros-11, S.16].

Erfahrung

Unter der Erfahrung ist „eine durch Praxis erworbene Fähigkeit sicherer Orientierung in Handlungs- und Sachzusammenhängen“ zu verstehen [Zwa-03].

**Erfahrungsbasiertes
Erlernen der Durchführung
von Simulationsstudien**

Kurz: EEDS

Das Vorgehen zum EEDS dient der Einarbeitung von Lernenden in die Durchführung von →Simulationsstudien. Eine →Lernfabrik ermöglicht es den Lernenden eine →Simulationsstudie praxisnah durchzuführen. Eine →Simulationsstudie umfasst die Datenaufnahme, →Abstraktion, den Modellaufbau, das Experimentieren mit dem digitalen →Modell sowie die Ergebnisanalyse. Hiervon abgeleitete Verbesserungen werden in der →Lernfabrik umgesetzt und überprüft. Lernende erhalten folglich eine Rückmeldung über die Wirksamkeit der ermittelten Verbesserungen in der →Lernfabrik. Der Aufbau von →Erfahrung in der Durchführung von →Simulationsstudien steht im Vordergrund.

Flow

Flow beschreibt das menschliche Empfinden von Konzentration, geringer Selbsterfahrung und Genuss. Dieser Zustand tritt typischerweise während herausfordernden Aktivitäten ein, die im Hinblick auf den Schwierigkeitsgrad an die Fähigkeiten der individuellen Person angepasst sind [Csi-14b] [Ull-12].

Die Flow-Theorie liegt den →IKS zugrunde.

Game-Engine

Game-Engines sind Software-Pakete, die die Erstellung digitaler Spiele ermöglichen. Es handelt sich um „lizenzierbare Softwarepakete, die die Infrastruktur des Spiels bereitstellen und der Echtzeitberechnung von digitalen 3-D-Umgebungen dienen“ [Sch-18, S.119] [Gre-09, S.3ff.]. →PSIMA ist in einer Game-Engine entwickelt worden.

Gamification	Gamification beschreibt die Anwendung von Spiel-Design-Elementen (siehe →Spielelement) in einem nicht-spielerischen Kontext [Det-11].
Geschichte	Eine Geschichte ist eine Erzählung. Ein →Levelpack in →PSIMA ist im sog. Story-Modus gestaltet. Maßnahmen zur Verbesserung des →Materialflusses werden durch eine Geschichte eingegrenzt. Geschichten kommen auch zum Einsatz, um Lernende in die Durchführung von →Simulationsstudien einzuarbeiten (siehe →Alternate Reality Game). Eine Geschichte ist ein →Spielelement.
Inkrementell komplizierter werdende Simulationsmodelle Kurz: IKS	Bei IKS handelt es sich um kurze, abgeschlossene Lerneinheiten, die induktiv aufeinander aufbauen. In schrittweise komplizierter werden den →Leveln werden dem Lernenden immer neue Aufgaben gestellt, zu deren Lösung er bislang unbekannte →Objekte und →Bausteine nutzen muss. Im Sinne des →Flows ist durch die IKS der Einklang von Herausforderungen in Form von zu lösenden Aufgaben mit den individuellen Fähigkeiten des Spielers sichergestellt. Durch IKS werden Lernende in den grafisch-interaktiven Modellaufbau unter Nutzung von →Bausteinen, →Objekten und →Programmiersprache zur Implementierung von Simulationsmodellen im Anwendungsbereich der →Materialflusssystemen eingearbeitet.
Instanz des Simulationsinstruments	Bei einer Instanz des →Simulationsinstruments handelt es sich um eine Kopie des →Simulationsinstruments im Hauptspeicher des Computers [Com-03]. Das in dieser Arbeit exemplarisch genutzte →Simulationsinstrument Tecnomatix Plant Simulation erlaubt den parallelen Betrieb mehrerer Instanzen.
Instanziierung/Instanz	In objektorientierten →Simulationswerkzeugen ist ein grafisch-interaktiver Modellaufbau üblich. Vordefinierte Klassenobjekte und -bausteine werden instanziiert und die Instanzen dann parametrisiert. Klassenobjekte und -bausteine sind mit den Instanzen durch die sogenannte Vererbung verbunden. Vererbt werden beispielsweise Werte von Attributen. In →PSIMA können die Lernenden sowohl →Objekte und →Bausteine in der →Modellierungsumgebung im Sinne der grafisch-interaktiven Modellierung als auch →Blöcke im →Block-Editor zur Erstellung eines →Programms instanziiieren.
Instruktionsdesign	Das Instruktionsdesign ist eine Disziplin in der erforscht wird, wie „Lernangebote bzw. →Lernumgebungen auf der Grundlage empirisch fundierter Theorien und Befunde systematisch konzipiert werden sollen, wenn bestimmte Bildungsziele zu erreichen sind“ [Nie-13].

Kompetenz

Kompetenz basiert auf →Wissen, das angewendet wird [Nor-18, S.42]. Kompetenz konkretisiert sich im Handlungsbezug und wird am erzielten Ergebnis der Handlung sicht- und messbar [Nor-18, S.41].

In der Arbeit steht der Aufbau von Kompetenzen zur Nutzung von →Simulationsinstrumenten zur Lösung unterschiedlicher Problemstellungen im Bereich der →Materialflusssysteme im Vordergrund. Zunächst baut der Lernende →Wissen über die Anwendung eines →Simulationsinstruments durch die Nutzung von →PSIMA auf. Im weiteren Verlauf des →Lernens und Spielens steht der Aufbau von Kompetenz im Vordergrund. Im Rahmen einer →Geschichte erzählte Problemstellungen sind durch den Lernenden zu lösen, indem er zunächst →Sensitivitätsanalysen (siehe →IKS) und später →Simulationsstudien (siehe →EEDS) durchführt.

Leaderboard

Ein Leaderboard stellt den Rang eines Spielers in einer Gruppe von Spielern dar. Das Leaderboard ist von den Mitgliedern einer Gruppe jederzeit einsehbar. In →PSIMA beeinflusst die individuelle →Punktzahl den Rang des Studierenden. Ein Leaderboard ist ein →Spielement.

Lernen

Unter Lernen wird ein Prozess verstanden, bei dem →Wissen durch Transformation von →Erfahrungen generiert wird [Kol-84, S.38].

Lernfabrik

Im engeren Sinne wird unter einer Lernfabrik eine →Lernumgebung verstanden, die authentische Prozesse, mehrere Stationen, technische als auch organisatorische Aspekte vereint [Abe-14] [Abe-15]. Der Aufbau einer Lernfabrik ist veränderlich und einer realen Wertschöpfungskette ähnlich. In der Lernfabrik wird ein physisches Produkt hergestellt. Ein veränderlicher Aufbau und ein didaktischer Ansatz kennzeichnen eine Lernfabrik. Lernende können Entscheidungen vor Ort treffen.

Lernumgebung

Die in dieser Arbeit entwickelte Lernumgebung entsteht durch die Anwendung von →PSIMA zur schrittweisen Einarbeitung in den grafisch-interaktiven Modellaufbau unter Nutzung von →Bausteinen und →Objekten (→IKS) zur Abbildung von →Materialflusssystemen, der →VBPIS zur Einarbeitung in die →Syntax und →Semantik von →Programmiersprache zur Implementierung von Simulationsmodellen sowie durch die Anwendung des Vorgehens zum →EEDS unter Nutzung einer →Lernfabrik zur Einarbeitung in die Durchführung von →Simulationsstudien.

Level	Level repräsentieren kurze abgeschlossene Lerneinheiten. Ein Level ist ein →Spielement. Level unterstützen die Strukturierung der →IKS in der App →PSIMA. Ein Level umfasst eine durch den Lernenden zu lösende Aufgabe.
Levelpack	→Level sind thematisch in Levelpacks zusammengefasst. In der Arbeit sind sechs Levelpacks nach Lerninhalten zur Abbildung von →Materialflusssystemen und nach didaktischen Gesichtspunkten konzipiert worden.
Materialflusssystem	„Die Anordnung von mindestens zwei Gegenständen des Materialflusses, die im Rahmen eines Transformationsprozesses eine Veränderung des Systemzustandes von Gütern hinsichtlich Zeit, Ort, Menge, Zusammensetzung und Qualität ermöglichen, wird als Materialflusssystem bezeichnet“. Die Gegenstände des Materialflusses umfassen das Personal, die Informationen, die Produktionsmittel, die Materialflussmittel, die Informationsflussmittel, die Energie und die Infrastruktur [Sch-08a].
Modell	Ein Modell ist eine →abstrahierte Abbildung eines Systems [Küh-06, S.28]. Ein Modell stellt eine „vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System“ dar [VDI-3633b, S.11].
Modellierungsumgebung	In der Modellierungsumgebung baut der Lernende grafisch-interaktiv →Simulationsmodelle auf. →Bausteine und →Objekte werden in der Modellierungsumgebung durch Benutzer →instanziiert und Attribute parametrisiert. In der App →PSIMA ist eine vereinfachte Modellierungsumgebung für den Lernenden bereitgestellt. Die vereinfachte Modellierungsumgebung ist hinsichtlich der grafischen Gestaltung an die Modellierungsumgebung eines exemplarischen →Simulationsinstruments angelehnt.
Modellierungskonzept	Ein Modellierungskonzept bezeichnet den Aufbau und die Verwaltung des →Simulationsmodells [Wen-08b, S.73ff.].
Modellwelt	Unter der Modellwelt wird die „Summe der zur Verfügung stehenden Modellelemente eines →Simulators“ verstanden [VDI-3633b, S.12]. Siehe →Baustein und →Objekt.

Multiagentensimulation	Die Multiagentensimulation ist ein Spezialfall der →ereignisdiskreten Simulation. Das Systemverhalten wird maßgeblich durch die Interaktion von →Agenten mit ihrer Umgebung und weiteren →Agenten bestimmt [Law-15, S.694]. Die Multiagentensimulation bildet ein Multiagentensystem ab. Synonym: →agentenbasierte →Simulation.
Objekt	Objekte sind Modellelemente ohne eigene interne Ablauflogik [VDI-3633a, S.26f.] [VDI-3633b, S.13]. Es werden physische Objekte und Informationsobjekte unterschieden, die je in temporärer oder permanenter Ausprägung auftreten können.
Peer-Review	Studierende, die viele →Punkte gesammelt haben, dürfen weniger →erfahrene Lernende evaluieren. Reviewende Lernende erhalten zur Belohnung →Punkte und →Abzeichen. Der Peer-Review ist ein →Spielement.
Production Simulation Application Kurz: PSIMA	Mobile und →gamifizierte App, die →VBPIS und →IKS in einer ähnlich zum →Simulationsinstrument gestalteten →Modellierungsumgebung integriert.
Programm	„Ein Programm ist eine Abfolge von Daten und Befehlen an einen Prozessor, um diese Daten in andere Daten umzuwandeln. Es wird in einer formal definierten Sprache verfasst, der Programmiersprache“ [Hen-07, S.24].
Programmiersprache zur Implementierung von Simulationsmodellen	In dieser Arbeit ist unter einer Programmiersprache zur Implementierung von Simulationsmodellen eine auf die Problemstellungen der →Simulation angepasste Programmiersprache zu verstehen, die in Kombination mit weiteren Modellbeschreibungsarten in →Simulationsinstrumenten zum Aufbau und für die Verwaltung von →Simulationsmodellen angeboten wird.
Punkt	Punkte dokumentieren sichtbar für eine Gruppe an Spielern den Fortschritt des einzelnen Spielers. Punkte werden für die Interaktion mit →PSIMA vergeben. Die Anzahl der Punkte, die ein Spieler sammelt, beeinflussen seinen Rang im →Leaderboard. Ein Punkt ist ein →Spielement.
Semantik	Die Semantik bezeichnet die Bedeutung einzelner Sprachelemente der →Syntaxdefinition [Hen-07, S.24]. Sprachelemente können Wörter, Phrasen oder Symbole umfassen [Hit-08, S.13].

Sensitivitätsanalyse	Bei einer Sensitivitätsanalyse handelt es sich um eine „Untersuchung des Systemverhaltens bei Änderung einer einzelnen Einflussgröße, während alle anderen konstant bleiben“ [VDI-3633b, S.16].
Serious game	Spiel mit explizitem und sorgfältig durchdachtem Bildungsziel [Abt-70].
Simulation	Unter Simulation versteht man das „Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren →Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind; insbesondere werden die Prozesse über die Zeit entwickelt“ [VDI-3633a, S.3].
Simulationsexperte	Der Simulationsexperte ist in der Lage, „die Aufbaustruktur und das Verhalten technischer Systeme, die Prozessabläufe sowie Organisations- und Steuerstrategien im Medium Software zu formulieren und auf diese Weise Werkzeuge für die Aufgabenlösung in Form problemspezifischer →Simulationsmodelle zur Verfügung zu stellen“ [VDI-3633a, S.10].
Simulationsinstrument	→Simulationswerkzeug, das sich an Endanwender richtet [Wen-08b, S.82]. Endanwender sind im Sinne dieser Arbeit Planer von →Materialflusssystemen und Studierende der Produktionstechnik. Die Mehrheit der in dieser Arbeit untersuchten →Simulationsinstrumente ermöglichen eine grafisch-interaktive Modellbeschreibung unter Verwendung von →Bausteinen/→Objekten. Die Nutzung von →Programmiersprache zur Implementierung von Simulationsmodellen ist bei nahezu allen untersuchten →Simulationsinstrumenten möglich. Andere Software kann durch Schnittstellen →Simulationsinstrumente steuern. Das in der Arbeit beispielhaft gewählte →Simulationsinstrument ist Tecnomatix Plant Simulation. Synonym: Simulator.
Simulationslauf	„Nachbildung des Verhaltens eines Systems mit einem spezifizierten ablauffähigen →Modell“ über einen bestimmten →Simulationszeitraum (siehe →Simulationszeit) [VDI-3633b, S.17].
Simulationszeit	„Modellgröße, die die im realen System voranschreitende Zeit im →Modell abbildet“ [VDI-3633b, S.18].

Simulationszeitraum	„Durch ein Beginn- und Enddatum begrenzter Zeitraum, in dem die →Simulation stattfindet“ [VDI-3633b, S.18].
Simulationsmethode	Die Simulationsmethode beschreibt das Zeitverhalten eines →Simulationsmodells [Wen-08b, S.73ff.].
Simulationsmodell	Ein Simulationsmodell ist eine „Abbildung eines realen oder geplanten Systems zum Zweck der →Simulation“ [VDI-3633b, S.17].
Simulationsstudie	Eine Simulationsstudie bezeichnet ein „Projekt zur simulationsgestützten Untersuchung eines Systems“ [VDI-3633b, S.18]. Eine Simulationsstudie besteht aus der Vorbereitungs-, Ausführungs- und Bewertungsphase. Lernende können mit dem Vorgehen zum →EEDS in die Durchführung von Simulationsstudien eingearbeitet werden.
Simulationswerkzeug	Simulationswerkzeuge werden dazu verwendet, die →Simulationszeit inklusive der Zustandsänderungen fortzuschreiben sowie den Aufbau und die Verwaltung eines →Simulationsmodells (→Modellierungskonzept) zu unterstützen [Wen-08b, S.73ff.]. Ein Simulationswerkzeug ermöglicht eine softwaretechnische Nachbildung eines Systems in einem →Modell. Simulationswerkzeuge bestehen in der Regel aus einem →Simulatorkern, einer Datenverwaltung, einer Benutzungsoberfläche (=GUI) und verfügen über Schnittstellen zu externer Software [VDI-3633a, S.14f.].
Simulator	Siehe →Simulationsinstrument.
Simulatorkern	Der Simulatorkern verknüpft und koordiniert die Prozesse einzelner Komponenten des →Simulationswerkzeugs. Er kann als die zentrale Ablaufsteuerung verstanden werden und manipuliert die Modelldaten [VDI-3633a, S.14f.]. Anwender können in der Regel nicht auf den Simulatorkern zugreifen.

Skalierbar

Der in der vorliegenden Arbeit beschriebene Aufbau der →Cloud-IT-Architektur ist skalierbar. Die Anzahl mobiler Apps sowie die Anzahl der →Simulationsinstrumente sind nahezu beliebig veränderbar. Auf Anwenderseite kann sich durch eine Vielzahl gleichzeitig mit der App interagierenden Personen eine Warteschlange nicht begonnener Simulationsaufträge vor dem →Simulationsinstrument bilden. Um einen Engpass bei der Abarbeitung der Simulationsaufträge zu vermeiden, lassen sich weitere →Simulationsinstrumente zuschalten. Die →Simulationsinstrumente werden auf Computern betrieben, die räumlich verteilt angeordnet sein können. Die →(teil-)automatische Modellgenerierung ermöglicht die Nutzung einer nahezu beliebigen Anzahl an →Simulationsinstrumenten. Durch diese skalierbare Architektur kann eine kurze Zeit bis zur Bereitstellung von Simulationsergebnisdaten auch bei hohem Simulationsaufkommen gewährleistet werden.

Spielelement

Kurzwort für Spiel-Design-Element. Mit Spielelementen werden Spiele gestaltet. Spielelemente kommen in →Serious games und der →Gamification zur Anwendung. In dieser Arbeit gestaltete Spielelemente sind →Abzeichen, →Geschichten, ein →Leaderboard, →Level, →Peer-Reviews, →Punkte und →Sterne.

Spielmechanik

Eine Spielmechanik bezeichnet das Zusammenspiel von →Spielelementen. Es handelt sich um einen Fachbegriff aus der Welt der Spieleentwickler.

Stern

Gelbe Sterne symbolisieren in →PSIMA den erfolgreichen Abschluss eines →Levels. Sterne stellen den Fortschritt des Lernenden für ihn offensichtlich dar. Ein Stern ist ein →Spielelement.

Syntax

Die Syntax beschreibt den formalen Aufbau von Zeichenketten nach gewissen Regeln (Grammatik) [VDI-3633b, S.19] [Hen-07, S.24].

(Teil-)automatische Modellgenerierung

Bei der (teil-)automatischen Modellgenerierung werden →Simulationsmodelle durch Algorithmen mit Daten einer externen Quelle generiert [Str-10]. Im →Simulationsinstrument, das Bestandteil der →Cloud-IT-Architektur ist, ist ein parametrischer Ansatz der (teil-)automatischen Modellgenerierung implementiert worden. →Simulationsmodelle werden auf Basis existierender →Bausteine/→Objekte erstellt [Eck-02] [Ber-14] [Str-10].

Validierung

Die Validierung bezeichnet die Prüfung auf hinreichende Übereinstimmung zwischen dem →Modell und der Realität [VDI-3633a, S.37]. Laut der Richtlinie VDI 3633 ist sicherzustellen, dass das →Modell das Verhalten des →Materialflusssystems genau genug und fehlerfrei widerspiegelt.

Verifikation

Die Verifikation bezeichnet einen Prozess, bei dem die Korrektheit von →Programmen oder Programmteilen formal nachgewiesen wird [VDI-3633b, S.21]. Eine Verifikation kann durch Syntax-Checks oder Konsistenz-Checks erfolgen [VDI-3633b, S.21]. Die →VBPIS erleichtert durch visuell vorgegebene Strukturen und Regeln die Erstellung verifizierter →Programme. Im Kontext der →Simulationsmodelle eignet sich die →Animation zur Verifikation.

Visuelle blockbasierte Programmiersprache

Programmieranfänger kombinieren durch Ziehen und Ablegen (→Instanziierung) vorbereitete →Blöcke und erhalten somit ausführbare →Programme. Die →Blöcke weisen visuelle Merkmale auf. So weiß der Benutzer intuitiv anhand von Formen und/oder Farben, wann ein →Block zu verwenden ist.

Visuelle blockbasierte Programmiersprache zur Implementierung von Simulationsmodellen

Kurz: VBPIS

Die VBPIS kombiniert die grafischen Eigenschaften der →visuellen blockbasierten Programmiersprache mit den Funktionen von →Programmiersprachen zur Implementierung von Simulationsmodellen. Die VBPIS macht →Programmiersprache zur Implementierung von Simulationsmodellen für Anfänger in der Programmierung zugänglich. Die VBPIS unterstützt den Lernenden in der Erstellung →verifizierter →Programme in →Simulationsinstrumenten. Die VBPIS ist prototypisch in →PSIMA integriert. Ein →Compiler übersetzt die VBPIS in eine →Programmiersprache zur Implementierung von Simulationsmodellen.

Wissen

„Wissen [...] basiert auf Informationen als Daten“, wobei Daten „in einem Bedeutungskontext stehen und als Grundlage von Entscheidungen und Handlungen dienen“ [Nor-18, S.41].

Mit →PSIMA wird zu Beginn des Lernverlaufs Wissen über →Bausteine und →Objekte, die →Syntax einer →Programmiersprache zur Implementierung von Simulationsmodellen und den grafisch-interaktiven Modellaufbau vermittelt.