

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für
Betriebswissenschaften und Montagetechnik

**Modellbasierte Entwicklung von Mensch-Maschine-
Schnittstellen im Maschinen- und Anlagenbau**

Christoph Julius Richter

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

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Michael Zäh

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Gunther Reinhart
2. Prof. Dr. phil. Klaus Bengler

Die Dissertation wurde am 19.03.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 01.10.2018 angenommen.

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) der Technischen Universität München und an der Fraunhofer Projektgruppe für Ressourceneffiziente Mechatronische Verarbeitungsmaschinen (RMV) sowie als Abteilungsleiter an der Fraunhofer-Einrichtung für Gießerei-, Composite- und Verarbeitungstechnik (IGCV).

Ein spezieller Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. Gunther Reinhart, Leiter des IGCV und des *iwb*, für die Förderung und Unterstützung meiner Arbeit sowie die Möglichkeit zur beruflichen und persönlichen Weiterentwicklung während meiner Promotionszeit. Weiterhin danke ich Herrn Prof. Dr.-Ing. Michael Zäh, Leiter des *iwb*, für die Übernahme des Vorsitzes der Prüfungskommission und die wertvollen Hinweise zu meinem Promotionsvorhaben. Sehr gefreut hat mich ferner die Übernahme des Koreferats durch Herrn Prof. Dr. phil. Klaus Bengler, Leiter des Lehrstuhls für Ergonomie an der Technischen Universität München. Ich bin ihm sehr dankbar für die Durchsicht meiner Arbeit und die angeregte Diskussion.

Außerdem bedanke ich mich recht herzlich bei allen Kolleginnen und Kollegen, insbesondere der Abteilung Anlagen- und Steuerungstechnik. Neben der fachlichen Unterstützung hat vor allem das kollegiale und motivierende Arbeitsumfeld dazu beigetragen, dass ich die Arbeit erfolgreich abschließen konnte. Mein spezieller Dank gilt dabei Herrn Dr.-Ing. Julian Backhaus, Herrn Dr.-Ing. Stefan Braunreuther und Herrn Dr.-Ing. Peter Stich für die kritische Durchsicht meiner Arbeit sowie die wertvollen Anregungen zu deren Verbesserung. Bedanken möchte ich mich weiterhin bei allen Studierenden, die mich bei meiner Promotionsarbeit unterstützt haben.

Ein großer Dank gilt darüber hinaus meiner Familie und meinen Freunden, die mir bei der Erstellung meiner Arbeit geholfen haben sowie mich generell in allen Lebenslagen unterstützen. Der größte und herzlichste Dank gilt schließlich meiner Frau Sabrina. Sie hat mir in den letzten Jahren immer den Rücken freigehalten, mich stets aufs Neue motiviert und mit ihrem Know-how im Graphikdesign auch einen wertvollen Beitrag zu den Abbildungen in dieser Arbeit geleistet.

München, im November 2018

Christoph Richter

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abkürzungsverzeichnis	V
1 Einleitung	1
1.1 Ausgangssituation und Motivation	1
1.2 Zielsetzung der Arbeit	4
1.3 Forschungsmethodisches Vorgehen	5
1.4 Aufbau der Arbeit	6
2 Grundlagen	9
2.1 Mechatronische Systeme und Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau	9
2.2 Interaktionstechnologien für Maschinen und Anlagen	12
2.2.1 Etablierte Interaktionstechnologien und aktuelle Trends	13
2.2.2 Potenziale zukünftiger Interaktionstechnologien	16
2.3 Softwareergonomie	19
2.3.1 Definition und Ziele der Softwareergonomie	19
2.3.2 Zentrale Qualitätsmerkmale ergonomischer Software	21
2.3.3 Relevante Normen und Richtlinien	23
2.4 Modellbildung und Simulation	26
2.4.1 Beschreibung des System-, Modell- und Simulationsbegriffs	26
2.4.2 Modellbildung bei mechatronischen Systemen und Mensch- Maschine-Schnittstellen	28
2.4.3 Simulation von mechatronischen Systemen und Mensch- Maschine-Schnittstellen	33
3 Stand der Wissenschaft und Technik	37
3.1 Überblick über die Entwicklung mechatronischer Systeme	37
3.2 Überblick über die Entwicklung von Mensch-Maschine-Schnittstellen	40
3.3 Entwicklungsansätze für mechatronische Systeme	42
3.3.1 Generelle Vorgehensmodelle und Entwicklungsmethoden	43

3.3.2	Spezielle Ansätze zur modellbasierten Entwicklung	45
3.3.3	Analyse der Ansätze hinsichtlich der Berücksichtigung von Mensch-Maschine-Schnittstellen	53
3.4	Entwicklungsansätze für Mensch-Maschine-Schnittstellen	54
3.4.1	Generelle Vorgehensmodelle und Entwicklungsmethoden	55
3.4.2	Spezielle Ansätze zur modellbasierten Entwicklung	62
3.4.3	Analyse der Ansätze hinsichtlich einer Übertragung auf mechatronische Entwicklungsprozesse	70
3.5	Identifikation des Handlungsbedarfs	72
4	Anforderungsanalyse	73
4.1	Anforderungen an die Entwicklungsmethodik	73
4.1.1	Allgemeine Anforderungen	73
4.1.2	Spezielle Anforderungen im Kontext der Mechatronik	75
4.1.3	Spezielle Anforderungen im Kontext der Mensch-Maschine-Schnittstelle	76
4.2	Auswahl einer mechatronischen Entwicklungsmethodik als Forschungsgrundlage	77
5	Modellbasierte Entwicklungsmethodik für Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau	79
5.1	Gesamtüberblick über die Entwicklungsmethodik	79
5.2	Aktivität I: HMI-Modellierung	82
5.2.1	Erweiterte Systemmodellierungstechnik zur Abbildung von HMIs	82
5.2.2	Prozessablauf für den Einsatz der Modellierungstechnik während der Konzeption von mechatronischen Systemen	92
5.3	Aktivität II: Technologieauswahl	100
5.3.1	Gegenüberstellung von Auswahlfaktoren und Interaktionstechnologien in einer Einflussmatrix	100
5.3.2	Prozessablauf für den Einsatz der Einflussmatrix bei der Technologieauswahl	111
5.4	Aktivität III: HMI-Implementierung	115

5.4.1	Transformation der Modelle von Mensch-Maschine-Schnittstellen	115
5.4.2	Prozessablauf für den Einsatz der Modelltransformation während der Implementierung von Mensch-Maschine-Schnittstellen	120
5.5	Aktivität IV: HMI-Qualitätssicherung	124
5.5.1	Eingesetzte Simulationstechniken	125
5.5.2	Prozessablauf für den Einsatz der Simulationstechniken	130
5.6	Zusammenfassung	134
6	Anwendung und Bewertung der modellbasierten Entwicklungsmethodik	135
6.1	Vorgehensweise bei der Anwendung und Bewertung	135
6.2	Softwareumsetzung der technischen Lösungsbausteine	136
6.2.1	Umsetzung der Modellierungstechnik, der Modelltransformation und der Simulationstechniken	136
6.2.2	Umsetzung der Einflussmatrix	140
6.3	Anwendung an einem industrienahen Fallbeispiel	141
6.3.1	Einführung des Fallbeispiels	141
6.3.2	HMI-Modellierung	142
6.3.3	Auswahl von Interaktionstechnologien	147
6.3.4	HMI-Implementierung	149
6.3.5	HMI-Qualitätssicherung	154
6.4	Technische Bewertung	158
6.4.1	Bewertung anhand der formulierten Anforderungen	158
6.4.2	Nutzenpotenziale, Aufwände und Einschränkungen	161
6.5	Wirtschaftliche Bewertung	163
6.6	Zusammenfassung	167
7	Schlussbetrachtung	169
8	Literaturverzeichnis	173

Anhang	205
A1 Detailbetrachtung des IMoMeSA-Ansatzes von HACKENBERG ET AL.	205
A2 Metamodell der Modellierungstechnik	218
A3 Auswahlfaktoren und Klassifikationskriterien	222
A4 Verzeichnis verwendeter Software	228
A5 Verzeichnis betreuter Studienarbeiten	229

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
AR	Augmented Reality
AQUIMO	Adaptierbares Modellierungswerkzeug und Qualifizierungsprogramm für den Aufbau firmenspezifischer mechatronischer Engineeringprozesse
AutomationML	Automation Markup Language
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CAVE	Cave Automatic Virtual Environment
CONSENS	Conceptual Design Specification Technique for the Engineering of Complex Systems
CS	C Sharp (C#)
CSA	Canadian Standards Association
DIN	Deutsches Institut für Normung
DISL	Dialog and Interaction Specification Language
E-CAD	Elektronisches CAD
EN	Europäische Norm
ERP	Enterprise Resource Planning
FEM	Finite Elemente Methode
GUI	Graphical User Interface
HMI	Human Machine Interface
HTML	Hypertext Markup Language

HÜMNOS	Herstellerübergreifende Module für den nutzerorientierten Einsatz der offenen Steuerungsarchitektur
IFIP	International Federation for Information Processing
IMoMeSA	Integrierte modellbasierte Entwicklung mechatronischer Systeme im Maschinen- und Anlagenbau
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
Lean UX	Lean User Experience
M-CAD	Mechanisches CAD
MACON	Manufacturing Conception
MARIA	Model-based Language for Interactive Applications
MEPROMA	Mechatronisches Engineering zur effizienten Produktentwicklung im Maschinen- und Anlagenbau
MES	Manufacturing Execution System
MVVM	Model View ViewModel
MVP	Minimum Viable Product
NC	Numerical Control
OSACA	Open System Architecture for Controls within Automation Systems
PEBeMA	Phasenübergreifende Entwicklung von Benutzerschnittstellen im Maschinen- und Anlagenbau
PLC	Programmable Logic Controller
PLM	Produktlebenszyklusmanagement
ProMiS	Projektmanagement für interdisziplinäre Systementwicklungen
RC	Robot Control

SCADA	Supervisory Control and Data Acquisition
SPS	Speicherprogrammierbare Steuerung
STEM	Spatio-Temporal Engineering Models
SysLM	Systems Lifecycle Management
SysML	Systems Modeling Language
SYSMOD	Systems Modeling Toolbox
UIML	User Interface Markup Language
UML	Unified Modeling Language
UMLi	Unified Modeling Language for Interactive Applications
UseDDL	Useware Data Description Language
UseML	Useware Markup Language
UsiXML	User Interface Extensible Markup Language
VDE	Verband der Elektrotechnik Elektronik Informationstechnik
VDI	Verein Deutscher Ingenieure
VDMA	Verband Deutscher Maschinen- und Anlagenbau
VIBN	Virtuelle Inbetriebnahme
VoiceXML	Voice Extensible Markup Language
VR	Virtual Reality
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
XT	Extreme Tailoring

1 Einleitung

Um ein grundsätzliches Verständnis für das in der vorliegenden Arbeit adressierte Themenfeld zu schaffen, werden in Abschnitt 1.1 zunächst die aktuelle Situation im Maschinen- und Anlagenbau sowie die daraus resultierende Motivation bezüglich Mensch-Maschine-Schnittstellen (Human Machine Interfaces / HMIs) und deren Entwicklung dargestellt. Auf dieser Basis wird in Abschnitt 1.2 die konkrete Zielsetzung beschrieben, um HMIs im Maschinen- und Anlagenbau zukünftig geeignet entwickeln zu können. Eine Darstellung des forschungsmethodischen Vorgehens sowie des Aufbaus der Arbeit in den Abschnitten 1.3 und 1.4 runden das Kapitel ab.

1.1 Ausgangssituation und Motivation

Bedingt durch die derzeitigen Megatrends im Maschinen- und Anlagenbau, wie eine Dynamisierung der Produktlebenszyklen verbunden mit einer Zunahme kundenindividueller Produkte (ABELE & REINHART 2011, S. 15; VOGEL-HEUSER 2013, S. 1), müssen Unternehmen mittlerweile auch in Bereichen innovativ sein, die sich nicht ihrer unmittelbaren Kernkompetenz zuordnen lassen (KRAUSE ET AL. 2007). Ein Beispiel mit großem Innovationspotenzial bilden die Mensch-Maschine-Schnittstellen von Maschinen und Anlagen, die bei geeigneter Fokussierung zu einem entscheidenden Wettbewerbsfaktor werden können (BRECHER ET AL. 2010, S. 35 f.; PEISSNER & HIPP 2013, S. 4; ACATECH 2016, S. 10). Diese Einschätzung wird u. a. durch eine unter Mitgliedsunternehmen des VDI durchgeführte Umfrage bestätigt, in der die Mensch-Maschine-Kommunikation nach der Energie- und Ressourceneffizienz als zweitwichtigster Impulsgeber für die Automatisierungstechnik in den kommenden Jahren angesehen wird (vgl. Abbildung 1-1).

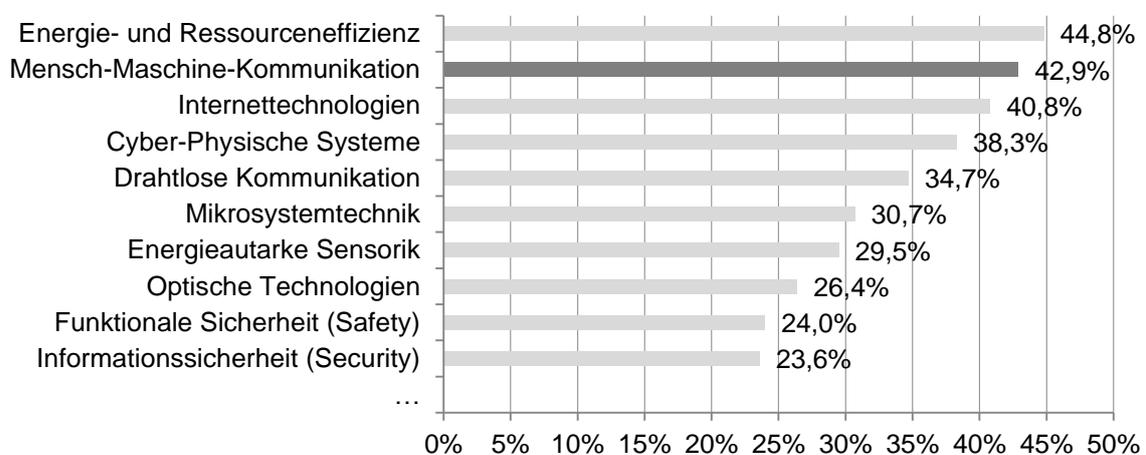


Abbildung 1-1: Wichtigste Impulsgeber für die Automatisierungstechnik bis zum Jahr 2020 (in Anlehnung an VDI/VDE 2013, S. 15)

Eine erste Begründung für diese Prognose lässt sich dabei aus den aktuellen Entwicklungen im Bereich der Konsumgüter ableiten. Hier haben ansprechende Mensch-Maschine-Schnittstellen bereits heute einen großen Anteil am Erfolg von Produkten, wie bspw. Smartphones oder Tablets (RICHTER & FLÜCKIGER 2013, S. 10 f.). Aufgrund der steigenden Bedeutung und Verfügbarkeit innovativer Technologien und einfach anzuwendender Software in diesem Bereich wird mittlerweile auch von den Kunden des Maschinen- und Anlagenbaus eine intuitive Interaktion mit produktionstechnischen Systemen bei gleichzeitig ansprechendem Design erwartet (PEISSNER & HIPPEL 2013, S. 6). In diesem Kontext spielt auch der Kulturwandel im Umgang mit Mensch-Maschine-Schnittstellen eine wichtige Rolle. Maschinenbenutzer sind heute in der Regel nicht mehr bereit, sich über umfassende Dokumentationen in die Funktionalität eines technischen Systems einzuarbeiten. Vielmehr wird erwartet, dass sich die einzelnen Interaktionsmöglichkeiten intuitiv über die Mensch-Maschine-Schnittstelle selbst erklären (ACATECH 2016, S. 25).

Allerdings lässt sich die steigende Bedeutung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau nicht allein mit den aktuellen Entwicklungen im Bereich der Konsumgüter begründen. Vielmehr wird eine intuitive Interaktion mit produktionstechnischen Systemen auch zunehmend erforderlich. Die Funktionsumfänge von Maschinen und Anlagen nehmen seit Jahren kontinuierlich zu, was meist zu einem Anstieg der Komplexität hinsichtlich der Interaktion mit diesen Systemen führt (ZÜHLKE 2012, S. 65). Damit Maschinenbenutzer trotz dieser Komplexität effizient mit Maschinen und Anlagen umgehen können, müssen HMIs anhand softwareergonomischer Prinzipien gestaltet sein, um eine hohe Gebrauchstauglichkeit (engl. Usability) zu gewährleisten. Dies ermöglicht es Benutzern, die vorhandenen Arbeitsaufgaben bestmöglich durchzuführen, dabei die Anzahl an Fehlbedienungen zu minimieren und somit einen wichtigen Beitrag zur Produktivität einer Maschine bzw. Anlage zu leisten (VDI/VDE 2015, S. 11).

Trotz dieser Erfordernisse kommt der Entwicklung von Software und insbesondere HMIs in weiten Teilen der Produktionstechnik aktuell nur eine untergeordnete Rolle zu, sodass Mensch-Maschine-Schnittstellen meist erst am Ende des gesamten Entwicklungsprozesses umgesetzt werden (VERL ET AL. 2009, S. 32; HENSEL 2011, S. 3; BACKHAUS ET AL. 2016, S. 38). Aufgrund des hier vorherrschenden Zeit- und Kostendrucks werden oft nur die notwendigsten Arbeiten durchgeführt (HOLM ET AL. 2013, S. 262 f.). Innovative Technologien oder die Ausrichtung des HMIs an den Bedürfnissen der Benutzer können hingegen nicht mehr berücksichtigt werden (BERG ET AL. 2017, S. 41). Ferner führt die späte Integration von Software und speziell HMIs in ein mechatronisches System zu großem Abstimmungsaufwand

zwischen den beteiligten Disziplinen (SPATH & DANGELMAIER 2016, S. 4). Dies kann in Verzögerungen im Entwicklungsprozess oder einer mangelnden HMI-Funktionalität resultieren, die letztlich einen großen Einfluss auf die Wahrnehmung der Gesamtqualität einer Maschine bzw. Anlage hat (VDI/VDE 2015, S. 11).

Um generell den Problemstellungen einer zu späten Realisierung von Software und der mangelnden Abstimmung der Fachdisziplinen zu begegnen, wurden im wissenschaftlichen Umfeld in den letzten Jahren verschiedene Vorgehensmodelle zur Entwicklung mechatronischer Systeme erarbeitet, die auch vermehrt im Maschinen- und Anlagenbau eingesetzt werden (HACKEL 2011). Diese Entwicklungsansätze zielen auf eine frühzeitige und kontinuierliche Einbindung aller Gewerke und somit eine Parallelisierung von Arbeitsschritten ab (WUCHERER 2004, S. 179), wofür insbesondere der Einsatz der Modellbildung und Simulation vorgeschlagen wird (VDI 2206, S. 4). Allerdings fokussieren aktuelle Ansätze in diesem Kontext primär die Integration der Disziplinen Mechanik, Elektrik und Steuerungstechnik. Mensch-Maschine-Schnittstellen werden bisher nicht oder nur als marginaler Bestandteil berücksichtigt, weswegen die zuvor aufgezeigten Probleme hinsichtlich der HMI-Entwicklung nach wie vor bestehen (RICHTER & REINHART 2014, S. 537).

Während integrative Herangehensweisen zur Konzeption und Umsetzung von Mensch-Maschine-Schnittstellen im Kontext mechatronischer Systeme fehlen, gibt es vor allem in der Informatik bereits eine Vielzahl an Ansätzen, die eine zielgerichtete und nutzerzentrierte HMI-Entwicklung als eigenständigen und in sich abgeschlossenen Prozess ermöglichen (RICHTER & FLÜCKIGER 2013, S. 16 f.). Die mangelnde Fokussierung auf den Maschinen- und Anlagenbau sowie die fehlende Integration in einen ganzheitlichen mechatronischen Entwicklungsprozess führen jedoch dazu, dass diese Ansätze im Hinblick auf produktionstechnische Systeme nicht eingesetzt werden können (RICHTER & REINHART 2014, S. 538).

Aus diesen Aspekten resultiert die generelle Problemstellung, dass eine geeignete Herangehensweise für die Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau derzeit fehlt. Ein neuartiger Ansatz kann dabei allerdings auf den bisherigen Erkenntnissen zur Entwicklung mechatronischer Systeme sowie den Vorarbeiten aus dem Bereich der HMI-Entwicklung aufbauen und diese bisher losgelöst voneinander betrachteten Themenfelder zu einem ganzheitlichen Entwicklungsansatz verbinden. Damit kann aus Entwicklungssicht der gestiegenen Bedeutung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau angemessen begegnet werden.

1.2 Zielsetzung der Arbeit

Vor diesem Hintergrund liegt die Zielsetzung der vorliegenden Arbeit im Entwurf einer Entwicklungsmethodik, welche eine integrative Konzeption und Umsetzung von Mensch-Maschine-Schnittstellen im Kontext mechatronischer Systeme des Maschinen- und Anlagenbaus ermöglicht. Durch eine frühe und kontinuierliche Berücksichtigung von HMIs im gesamten Entwicklungsprozess soll die Möglichkeit geschaffen werden, Mensch-Maschine-Schnittstellen an den Bedürfnissen der späteren Benutzer auszurichten und innovative Interaktionstechnologien einzusetzen. Dazu soll die zu erarbeitende Methodik auf den im aktuellen Stand der Wissenschaft und Technik verfügbaren Ansätzen zur Entwicklung von mechatronischen Systemen und Mensch-Maschine-Schnittstellen aufbauen, diese entsprechend zusammenführen und erweitern. Dabei ist es insbesondere wichtig, den spezifischen Anforderungen des Maschinen- und Anlagenbaus an die Entwicklung von HMIs gerecht zu werden.

Zur Erreichung dieser Zielsetzung soll im Rahmen dieser Arbeit speziell der Einsatz der Modellbildung und Simulation fokussiert werden, da sich diese Konzepte zur frühzeitigen Integration sowie kontinuierlichen Synchronisation der an einem Entwicklungsprozess beteiligten Disziplinen anbieten (VDI 2206, S. 4; EIGNER 2013, S. 91) und zu einer effizienten Entwicklung einen wichtigen Beitrag leisten (ROßMANN & HEINZE 2010, S. 1). Sowohl im Kontext mechatronischer Systeme als auch im HMI-Bereich haben sich jeweils eigene Konzepte etabliert, die im Rahmen dieser Arbeit genutzt, erweitert und aufeinander abgestimmt werden sollen. Neben dem Einsatz der Modelle zur interdisziplinären Abstimmung sollen diese so gestaltet werden, dass sie explizit eine Berücksichtigung der Benutzer unterstützen und zu diesem Zweck bspw. eine frühzeitige Absicherung der Funktionalität sowie der Gebrauchstauglichkeit mittels Simulation gestatten.

Schließlich soll die zu erarbeitende Entwicklungsmethodik speziell den Einsatz innovativer Interaktionstechnologien im Maschinen- und Anlagenbau ermöglichen. Dazu soll im Rahmen dieser Arbeit einerseits eine Möglichkeit geschaffen werden, Entwickler bei der Auswahl einer geeigneten Technologie für eine konkrete Maschine bzw. Anlage zu unterstützen. Andererseits soll die Entwicklungsmethodik so gestaltet werden, dass nicht a priori nur ein ausgewähltes Spektrum an Technologien, wie bspw. graphische HMIs, betrachtet wird. Insbesondere in einer frühen Entwicklungsphase wird vielmehr eine funktionale und lösungsneutrale Betrachtung von Mensch-Maschine-Schnittstellen angestrebt, anhand derer geeignete Interaktionstechnologien identifiziert werden können.

1.3 Forschungsmethodisches Vorgehen

Um die eben beschriebene Zielsetzung erreichen zu können, ist ein forschungsmethodisches Vorgehen notwendig, welches den Forschungsprozess strukturiert. Die Wahl einer geeigneten Vorgehensweise hängt allerdings von der jeweiligen Wissenschaftskategorie ab, weshalb die vorliegende Problemstellung zunächst in der Wissenschaftssystematik nach ULRICH & HILL (1976, S. 305) eingeordnet wird. Diese Systematik sieht auf oberster Ebene eine Unterteilung in Formal- und Realwissenschaften vor. Während Formalwissenschaften, wie die Mathematik oder die Logik, auf die Konstruktion von Sprachen bzw. Zeichensystemen ausgelegt sind, adressieren Realwissenschaften „die Beschreibung, Erklärung und Gestaltung empirisch (sinnlich) wahrnehmbarer Wirklichkeitsausschnitte“ (ULRICH & HILL 1976, S. 305). Realwissenschaften lassen sich dabei weiterführend in Grundlagen- und Handlungswissenschaften unterteilen. Die Naturwissenschaften zählen zur ersten Kategorie, die sich grundsätzlich mit der Erklärung von Wirklichkeitsausschnitten beschäftigt. Demgegenüber steht bei Handlungswissenschaften, zu denen u. a. die Sozialwissenschaften zählen, die Analyse menschlicher Handlungsalternativen zur Gestaltung sozialer und technischer Systeme im Vordergrund. Die Ingenieurwissenschaften lassen sich nicht unmittelbar einer dieser Kategorien zuordnen und bewegen sich im Spannungsfeld zwischen den Grundlagen- und Handlungswissenschaften. Aufgrund der konkreten Zielsetzung der vorliegenden Arbeit, einen Prozess zur Entwicklung von Mensch-Maschine-Schnittstellen zu gestalten, kann diese allerdings eindeutig den Handlungswissenschaften zugeordnet werden.

Aus dieser Einordnung lässt sich das forschungsmethodische Vorgehen für diese Arbeit ableiten, das auf der Strategie angewandter Forschung nach ULRICH ET AL. (1984, S. 192 f.) basiert. Demzufolge beginnt der Erkenntnisgewinn mit der Identifikation und Formulierung von Problemen (vgl. auch POPPER 1962), wie sie für die Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau in Abschnitt 1.1 formuliert wurden. Für die identifizierte Problemstellung werden daraufhin problemrelevante Theorien, Hypothesen und Verfahren recherchiert und interpretiert. Darüber hinaus wird der relevante Anwendungszusammenhang erfasst und untersucht, bevor die eigentliche wissenschaftliche Lösung durch das Ableiten von Gestaltungsregeln und -modellen erarbeitet wird. Im Zusammenhang der Anwendung werden diese Regeln und Modelle überprüft, bevor abschließend eine Beratung der Praxis erfolgt. In diesem Kontext ist zu erwähnen, dass durch die Überprüfung die Richtigkeit einer Lösung zwar bestätigt, aber nicht formal bewiesen werden kann. Es ist die Aufgabe des wissenschaftlichen Diskurses, eine erarbeitete Lösung kritisch zu hinterfragen und ggf. zu widerlegen (POPPER 1962).

1.4 Aufbau der Arbeit

Aus dem eben eingeführten forschungsmethodischen Vorgehen resultiert auch der Aufbau der vorliegenden Arbeit, welcher in Abbildung 1-2 dargestellt ist. Auf Basis der zuvor beschriebenen Motivation und Zielsetzung werden in Kapitel 2 die zum Verständnis dieser Arbeit notwendigen Grundlagen erläutert. Neben einer allgemeinen Begriffsklärung und Einordnung von mechatronischen Systemen und Mensch-Maschine-Schnittstellen werden aktuelle HMI-Technologien vorgestellt, bevor die Themenfelder der Softwareergonomie sowie der Modellbildung und Simulation betrachtet werden.

Kapitel 3 widmet sich anschließend der Vorstellung und Analyse der relevanten Ansätze des Stands der Wissenschaft und Technik, welche in die hier erarbeitete Entwicklungsmethodik für Mensch-Maschine-Schnittstellen eingeflossen sind. Aufbauend auf einem Überblick über die Entwicklungsprozesse von mechatronischen Systemen und Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau werden in beiden Bereichen ausgewählte Vorgehensmodelle und Entwicklungsmethoden vorgestellt. Dabei werden speziell modellbasierte Ansätze fokussiert, die für die vorliegende Arbeit von zentraler Bedeutung sind. Die präsentierten Ansätze werden jeweils hinsichtlich ihrer Potenziale für die adressierte Zielsetzung analysiert, bevor der Handlungsbedarf für die vorliegende Arbeit abgeleitet wird.

Der aufgezeigte Handlungsbedarf wird in Kapitel 4 aufgegriffen, um konkrete Anforderungen an die Entwicklungsmethodik abzuleiten. Neben deren Klassifikation und Beschreibung wird auf Basis der formulierten Anforderungen ein konkreter Entwicklungsansatz für mechatronische Systeme aus dem Stand der Wissenschaft und Technik ausgewählt, der im weiteren Verlauf als Forschungsgrundlage für die vorliegende Arbeit verwendet wird.

Im anschließenden Kapitel 5 wird die Methodik zur modellbasierten Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau beschrieben. Ausgehend von einem Überblick werden die einzelnen Lösungsbausteine im Detail vorgestellt. Neben einer Modellierungstechnik zur Abbildung von HMIs als Bestandteile mechatronischer Systeme umfassen diese Bausteine auch eine Methode für die Technologieauswahl sowie Möglichkeiten für eine Modelltransformation und Qualitätssicherung mittels Simulation. Bei jedem der Bausteine wird darauf eingegangen, wie dieser im Verlauf eines mechatronischen Entwicklungsprozesses einzusetzen ist und welche Wechselwirkungen mit weiteren Disziplinen bestehen.

Kapitel 6 widmet sich daraufhin der Anwendung und Bewertung der modellbasierten Entwicklungsmethodik. Dabei wird zunächst die Vorgehensweise dargestellt, bevor die prototypische Softwareumsetzung der einzelnen Lösungsbausteine thematisiert wird. Diese wiederum bildet die Basis für die Anwendung der Entwicklungsmethodik an einem repräsentativen Fallbeispiel, die im weiteren Verlauf des Kapitels vorgestellt wird. Die dabei gewonnenen Erkenntnisse fließen schließlich in eine technische und wirtschaftliche Bewertung der erarbeiteten Entwicklungsmethodik ein.

Im abschließenden Kapitel 7 werden die Ergebnisse der vorliegenden Arbeit zusammengefasst. Darüber hinaus wird ein Ausblick auf weiterführenden Forschungsbedarf im Zusammenhang mit der Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau gegeben.

1 Einleitung	<ul style="list-style-type: none"> ▪ Ausgangssituation und Motivation ▪ Zielsetzung der Arbeit ▪ Aufbau der Arbeit
2 Grundlagen	<ul style="list-style-type: none"> ▪ Mechatronische Systeme und HMIs ▪ Interaktionstechnologien ▪ Softwareergonomie ▪ Modellbildung und Simulation
3 Stand der Wissenschaft und Technik	<ul style="list-style-type: none"> ▪ Überblick über die Entwicklung von mechatronischen Systemen und HMIs ▪ Relevante Vorgehensmodelle und Entwicklungsmethoden
4 Anforderungsanalyse	<ul style="list-style-type: none"> ▪ Anforderungsklassifikation und -beschreibung ▪ Auswahl einer Entwicklungsmethodik als Forschungsgrundlage
5 Entwicklungsmethodik	<ul style="list-style-type: none"> ▪ Gesamtüberblick der modellbasierten Entwicklungsmethodik ▪ Detailbetrachtung der einzelnen Lösungsbausteine
6 Anwendung und Bewertung	<ul style="list-style-type: none"> ▪ Vorgehensweise ▪ Softwareumsetzung ▪ Anwendung an einem Fallbeispiel ▪ Techn. und wirtschaftl. Bewertung
7 Schlussbetrachtung	<ul style="list-style-type: none"> ▪ Zusammenfassung ▪ Ausblick

Abbildung 1-2: Aufbau der Arbeit

2 Grundlagen

Im folgenden Kapitel werden die wesentlichen Grundlagen zum Verständnis der in dieser Arbeit adressierten Themenfelder geschaffen. Dazu werden zunächst in Abschnitt 2.1 mechatronische Systeme und Mensch-Maschine-Schnittstellen definiert sowie in Relation zueinander gesetzt. Auf dieser Basis wird beschrieben, welche Technologien sich bereits für die Interaktion mit Maschinen und Anlagen etabliert haben sowie welche Trends und zukünftigen Entwicklungen in diesem Bereich zu erwarten sind (Abschnitt 2.2). Weiterhin erfolgt in Abschnitt 2.3 eine Einführung in das Themenfeld der Softwareergonomie, bevor in Abschnitt 2.4 die wesentlichen Grundlagen zur Modellbildung und Simulation von mechatronischen Systemen und Mensch-Maschine-Schnittstellen dargestellt werden.

2.1 Mechatronische Systeme und Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau

Wie bereits im ersten Kapitel dieser Arbeit dargelegt, setzen sich die Produkte des Maschinen- und Anlagenbaus mittlerweile in der Regel aus mechanischen, elektronischen und informationstechnischen Bestandteilen zusammen und werden deswegen als mechatronische Systeme bezeichnet (GRABOWSKI & GEIGER 1997, S. 187). Der Begriff *Mechatronik* wurde in den 70er-Jahren des 20. Jahrhunderts geprägt und setzte sich ursprünglich aus den Begriffen Mechanik und Elektronik zusammen (ISERMANN 2008, S. 3). Bereits in dieser Zeit sollte mit dem Kunstwort Mechatronik das interdisziplinäre Zusammenwirken der Ingenieurwissenschaften des Maschinenbaus und der Elektrotechnik ausgedrückt werden (JANSCHKE 2010, S. 2; RODDECK 2016, S. 1). Mit dem Aufkommen der Mikroelektronik kam schließlich die Informationstechnik als dritter integraler Bestandteil der Mechatronik hinzu (SCHWEITZER 1989). Nach heutigem Verständnis bezeichnet Mechatronik „(...) das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik bei dem Entwurf und bei der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung“ (VDI 2206, S. 14).

Insbesondere beim Entwurf von Maschinen und Anlagen ist aufgrund der vorherrschenden Komplexität ein Denken in Systemen (vgl. Abschnitt 2.4.1) mittlerweile unabdingbar (GAUSEMEIER ET AL. 2013, S. 10). Gemäß der VDI 2206 setzen sich *mechatronische Systeme* im Wesentlichen aus einem Grundsystem, Sensoren, Aktoren und einer Informationsverarbeitung zusammen (VDI 2206, S. 14 f.). Das Grundsystem besteht im Maschinen- und Anlagenbau zumeist aus einer

Kombination von mechanischen, elektromechanischen, pneumatischen und hydraulischen Komponenten. Messbare physikalische Werte dieses Grundsystems werden von Sensoren erfasst und der informationsverarbeitenden Einheit als Eingangsgrößen bereitgestellt. Dort werden die notwendigen Eingriffe bestimmt, um das Grundsystem in gewünschter Weise zu beeinflussen. Die Umsetzung dieser Eingriffe erfolgt über Aktoren, welche die rechnerinterpretierbaren Signale in physikalische Eingangsgrößen für das Grundsystem umwandeln. Neben diesen notwendigen Einheiten existieren zusätzliche, optionale Systembestandteile, die im Produktivbetrieb mechatronischer Systeme jedoch oftmals eine zentrale Rolle einnehmen. Neben der Umgebung des mechatronischen Systems, die bspw. Einflüsse auf ermittelte Sensorwerte haben kann, ist auch eine Kopplung der Informationsverarbeitung zu weiteren IT-Komponenten über Kommunikationssysteme möglich, um z. B. spezielle Aufgaben gemeinsam mit weiteren mechatronischen Systemen im Fabrikumfeld umzusetzen (SCHNELL 2003, S. 1). Einen letzten wichtigen Baustein im Kontext mechatronischer Systeme bildet schließlich die Schnittstelle des mechatronischen Systems zum Menschen, die im Rahmen der vorliegenden Arbeit als Mensch-Maschine-Schnittstelle bezeichnet wird. Abbildung 2-1 gibt zusammenfassend einen Überblick über den Aufbau und die Einheiten mechatronischer Systeme, wobei die Mensch-Maschine-Schnittstelle aufgrund ihrer zentralen Rolle in dieser Arbeit besonders hervorgehoben ist. (VDI 2206, S. 14 f.)

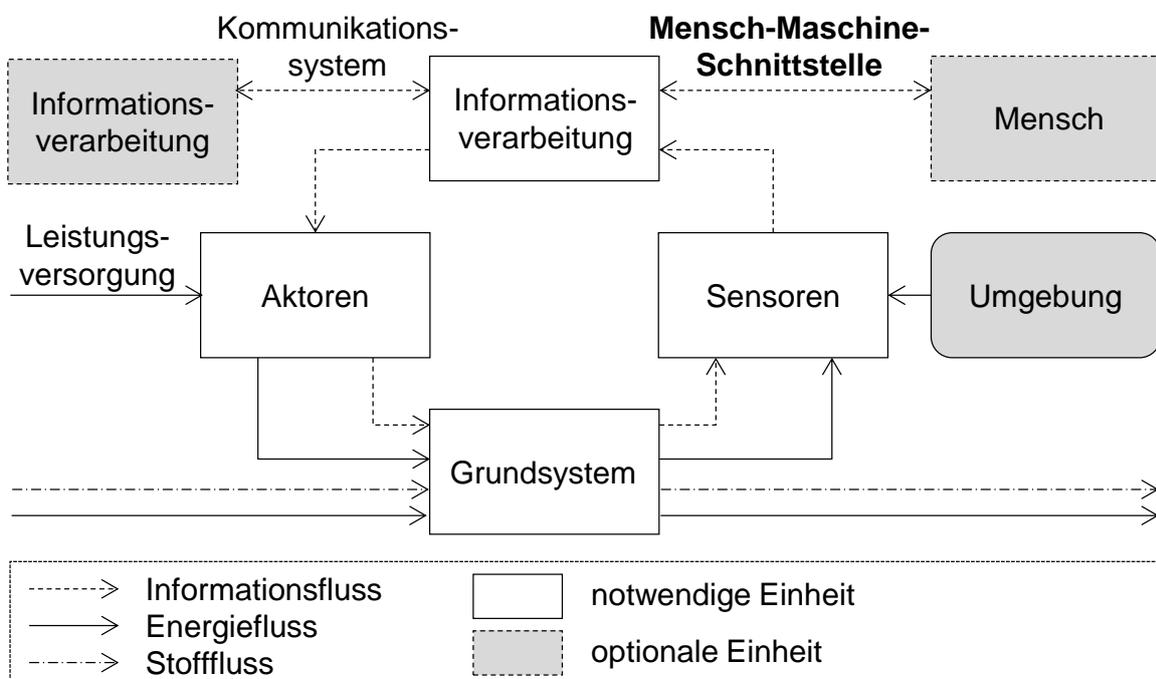


Abbildung 2-1: Aufbau mechatronischer Systeme (nach VDI 2206, S. 14)

Grundsätzlich werden der Mensch und somit auch die *Mensch-Maschine-Schnittstelle* in der VDI 2206 zwar als optionale Einheiten angesehen, jedoch ist die Interaktion eines Benutzers bei nahezu allen Maschinen und Anlagen explizit vorgesehen (GORECKY ET AL. 2017, S. 217). Typische Benutzungsaufgaben sind bspw. das Einstellen von Steuerungs- und Regelungsparametern, das Überwachen des Systemzustands sowie das Eingreifen im Fehlerfall (SIEMON 2001, S. 44 f.). Allgemein definiert die DIN EN ISO 9241-110 eine Mensch-Maschine-Schnittstelle als „(...) alle Bestandteile eines interaktiven Systems, die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Arbeitsaufgabe mit dem interaktiven System zu erledigen“ (DIN EN ISO 9241-110, S. 7). Als interaktive Systeme werden dabei beliebige Systeme angesehen, die aus einer Kombination von Hard- und Software resultieren.

Da durch die Ergebnisse dieser Arbeit speziell der Entwicklungsprozess von *Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau* unterstützt werden soll, wird die obige Definition im Folgenden hinsichtlich einer Anwendung im Kontext produktionstechnischer mechatronischer Systeme konkretisiert. An die Stelle allgemeiner interaktiver Systeme treten hier Maschinen und Anlagen, die von einem Benutzer überwacht und bedient werden müssen. Vor diesem Hintergrund definiert bspw. die VDI/VDE 3850 eine Mensch-Maschine-Schnittstelle als „(...) alle Komponenten einer technischen Anlage, die der Interaktion mit Nutzern dienen und von diesen wahrnehmbar sind“ (VDI/VDE 3850 – BLATT 1, S. 2 f.). Diese Komponenten lassen sich nach LANGMANN (2017) weiterhin in Anzeige- und Bedienelemente unterteilen. Während über Anzeigeelemente (z. B. Bildschirme) der Zustand einer Anlage visualisiert wird, ermöglichen Bedienelemente (z. B. Schalter) den Eingriff von Benutzern in den Prozess der Anlage. In beiden Fällen erfolgt die Kommunikation über Variablen, die mit der Steuerung der Maschine bzw. Anlage ausgetauscht werden. Die Steuerungseinheit ist in diesem Kontext dafür verantwortlich, die Benutzereingaben an die Aktorik des Systems weiterzuleiten bzw. den über Sensorwerte ermittelten Systemzustand an den Anzeigekomponenten darzustellen (LANGMANN 2017, S. 422 f.).

In Ergänzung zu der eben aufgeführten Definition, in der die Mensch-Maschine-Schnittstelle primär anhand ihrer Aufgabe charakterisiert wird, erfolgt eine Einordnung der in dieser Arbeit betrachteten Mensch-Maschine-Schnittstellen im Kontext der *Automatisierungspyramide* gemäß DIN EN 62264-1. In Anlehnung an die Beschreibungen von COPA-DATA (2017) und SAUTER (2010, S. 3586) werden HMIs im Rahmen dieser Arbeit der Steuerungsebene zugeordnet, um zum Ausdruck zu bringen, dass die Überwachung und Bedienung einer einzelnen

Maschine oder Anlage im Fokus steht (vgl. Abbildung 2-2). Damit unterscheiden sich die hier betrachteten HMIs insbesondere von SCADA-Systemen (Supervisory Control and Data Acquisition), die auf Prozessleitebene Anwendung finden, um bspw. maschinenübergreifend Betriebsdaten zu erfassen, auszuwerten und zu archivieren (COPA-DATA 2017; THIEL ET AL. 2008, S. 48). Es sei abschließend angemerkt, dass neben der eben dargelegten Einordnung von Mensch-Maschine-Schnittstellen in der Automatisierungspyramide auch Darstellungen existieren, in denen der Begriff SCADA übergreifend für alle Bedien- und Überwachungssysteme im Fabrikumfeld verwendet wird, wie z. B. DANEELS & SALTER (1999). Aufgrund der unterschiedlichen Aufgaben und auch Anforderungen (z. B. hinsichtlich einzusetzender Technologien) auf den einzelnen Ebenen der Automatisierungspyramide liegt der Fokus in dieser Arbeit jedoch auf Mensch-Maschine-Schnittstellen für Maschinen und Anlagen auf Steuerungsebene.

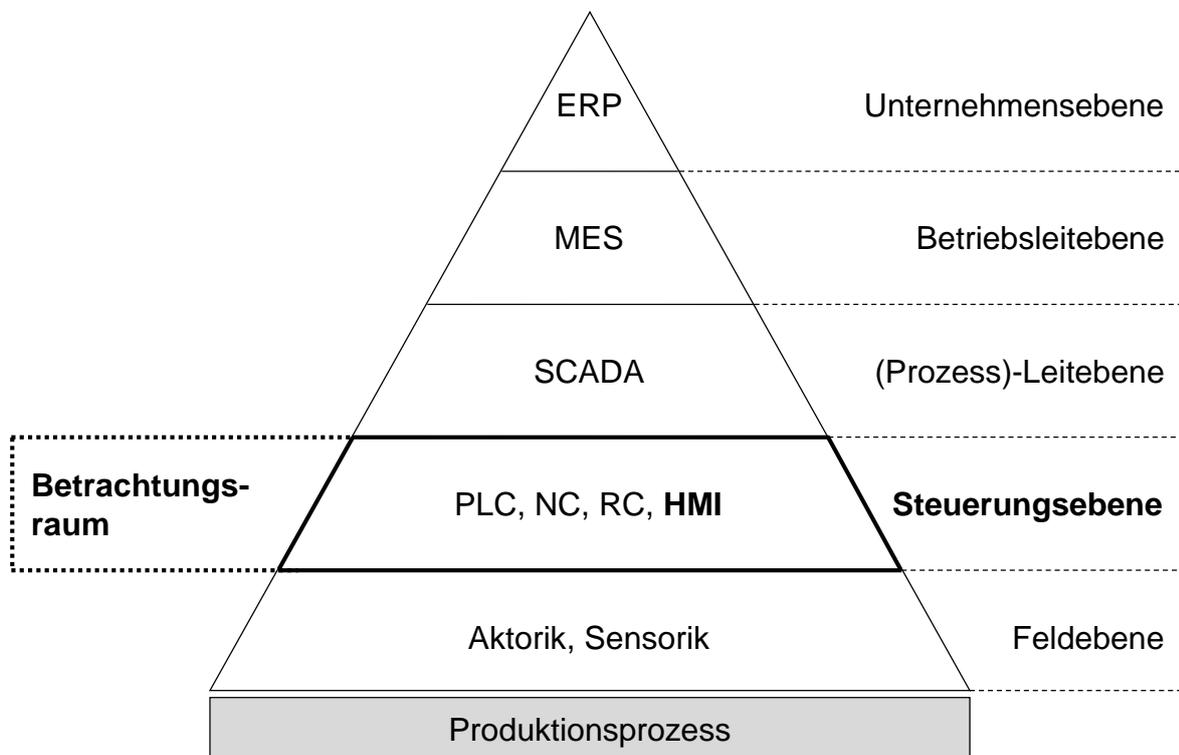


Abbildung 2-2: Betrachtungsraum in der Automatisierungspyramide (in Anlehnung an DIN EN 62264-1, COPA-DATA 2017 und SAUTER 2010, S. 3586)

2.2 Interaktionstechnologien für Maschinen und Anlagen

Um die im letzten Abschnitt beschriebenen Mensch-Maschine-Schnittstellen zu realisieren, existieren vielfältige Technologien zur Interaktion mit Maschinen und Anlagen. Seit dem Beginn der Forschung in diesem Themenbereich in den frühen

1940er-Jahren haben unterschiedlichste Konzepte den Umgang von Menschen mit technischen Systemen geprägt. TIMPE ET AL. (2002) bieten einen detaillierten Überblick über die Historie der Mensch-Maschine-Interaktion und gehen dabei auch auf die sich verändernde Rolle des Menschen vom reinen Regler bis zum interaktiven Problemlöser ein. Auch KOLSTER (2014, S. 5-12) zeigt am Beispiel von Werkzeugmaschinen auf, wie sich durch den fortschreitenden Einzug der Informationstechnik die Maschinenbedienung in der Produktionstechnik verändert hat. Aktuell existiert bereits ein breites Spektrum an potenziell einsetzbaren Technologien, welches sich zukünftig noch deutlich erweitern wird (ZÜHLKE 2012, S. 229). Vor diesem Hintergrund wird in Abschnitt 2.2.1 zunächst ein Überblick hinsichtlich relevanter Technologien gegeben, die sich in der Produktionstechnik entweder bereits etabliert haben oder gerade in die industrielle Praxis vordringen. Darauf aufbauend werden in Abschnitt 2.2.2 weiterführende Konzepte vorgestellt, die zukünftig in den Fokus des Maschinen- und Anlagenbaus rücken können. Dabei werden in den nachfolgenden Abschnitten allerdings lediglich die wesentlichen Charakteristika dieser Technologien vorgestellt, die zum Verständnis dieser Arbeit ausreichen. Für eine detaillierte Veranschaulichung, wie bspw. der physikalischen Wirkprinzipien, kann auf eine Vielzahl an verfügbarer Literatur, wie BAUMANN & LANZ (2013) oder JACKO (2012), zurückgegriffen werden.

2.2.1 Etablierte Interaktionstechnologien und aktuelle Trends

Obwohl sich die Rolle des Menschen bei der Interaktion mit technischen Systemen durch den fortschreitenden Einzug der Informationstechnik gewandelt hat (TIMPE ET AL. 2002), haben *einfache Bedien- und Anzeigeelemente*, wie Knöpfe, Handräder oder Lampen (vgl. Abbildung 2-3, links), dennoch nach wie vor ihre Berechtigung (BAUMANN & LANZ 2013, S. 35-37). Durch den meist unidirektionalen Datenfluss kann eine schnelle und zielgerichtete Informationsübermittlung erreicht werden. Der begrenzte Funktionsumfang dieser Elemente führt allerdings dazu, dass zur vollständigen Bedienung einer Maschine in der Regel eine Vielzahl an Komponenten notwendig ist, was eine intuitive Anwendung erschwert (SCHOLZ 2013). Vor diesem Hintergrund werden einfache Bedien- und Anzeigeelemente heute meist nur für spezielle Aufgaben eingesetzt, wie bspw. zur Realisierung eines Not-Aus oder zur Visualisierung des Maschinenzustands (KNOLL 2014).

Durch die Entwicklungen im Bereich der Mikroprozessortechnik und der Bildschirmtechnologie haben sich seit Beginn der 1980er-Jahre *stationäre und mobile HMI-Panels* (vgl. Abbildung 2-3, rechts) im Maschinen- und Anlagenbau

etabliert (KOLSTER 2014, S. 9 f.). Ein zentrales Merkmal dieser Technologien ist die graphische Informationsvisualisierung über Bildschirme, die entweder fest in den zugehörigen Schaltschränken verbaut werden oder in mobilen Handbediengeräten integriert sind (SHARMA 2011, S. 47 ff.). Ein weiteres Charakteristikum ist eine eigene integrierte Informationsverarbeitung, welche die darzustellenden Bildschirm-inhalte berechnet, die Benutzereingaben verarbeitet und die Kommunikation mit der Maschinensteuerung bewerkstelligt (SHARMA 2011, S. 47). Eingabeseitig kommen vereinzelt Tasten und Knöpfe zum Einsatz, ein Großteil der Benutzerinteraktion wird allerdings bereits über Touchscreens realisiert, wobei sich bislang vor allem die Single-Touch-Technologie im industriellen Umfeld etablieren konnte (KOLSTER 2014, S. 10). Durch eine berührungsempfindliche bzw. positionsermittelnde Oberfläche wird die Funktionalität von Bildschirmen von der reinen Anzeige zur integrierten Ein- und Ausgabe erweitert (SCHENK & RIGOLL 2010, S. 17).

Einfache Bedien- und Anzeigeelemente	Stationäre und mobile HMI-Panels
 <ul style="list-style-type: none"> ▪ Bedienung über Hebel, Tasten, Handräder, etc. ▪ Visualisierung über Lampen, Analoganzeigen, etc. ▪ Geringer Funktionsumfang bei unidirektionaler Interaktion 	 <ul style="list-style-type: none"> ▪ Bedienung über Tastaturen, Touch-Displays, etc. ▪ Visualisierung über (LCD-) Bildschirme ▪ Großer Funktionsumfang bei bidirektionaler Interaktion

Abbildung 2-3: Etablierte Interaktionstechnologien

Dem Grundprinzip der berührungsempfindlichen Oberflächen folgend, wurden in den letzten Jahren auch Bedienkonzepte entwickelt, die eine gleichzeitige Erkennung mehrerer Berührungspunkte ermöglichen. THANNER ET AL. 2010 bezeichnen *Multi-Touch* (vgl. Abbildung 2-4, links) als „(...) generell alle Eingabegeräte (...), bei denen mindestens zwei Berührungspunkte auf einem Bildschirm erkannt und verfolgt werden können“ (THANNER ET AL. 2010). Durch den bereits hohen Verbreitungsgrad dieser Technologie im Konsumgüterbereich rückt die Multi-Touch Bedienung auch vermehrt ins Blickfeld des Maschinen- und Anlagenbaus (MAYER & PANTFÖRDER 2014, S. 484). Auch MACHATE ET AL. (2014, S. 13 f.) sehen in Multi-Touch eine wichtige Interaktionstechnologie für den Maschinen- und Anlagenbau und begründen dies mit der natürlichen und intuitiven Bedienung, die

aufgrund leicht erlernbarer Gesten möglich ist. Gesten sind in diesem Kontext als Bewegungen von einem oder mehreren Berührungspunkten nach einem vorgegebenen Bewegungsmuster zu verstehen (DORAU 2011), anhand derer Benutzer bspw. eine aktuelle Ansicht drehen, näher heranzoomen oder den angezeigten Inhalt wechseln können (KNOLL 2011). Somit sind Gesten im Kontext der Multi-Touch Bedienung spezifischer als im allgemeinen Sprachgebrauch definiert, vgl. DUDEN (2018).

Einen weiteren Trend im Bereich industrieller Mensch-Maschine-Schnittstellen bilden nach HECHTMAN (2017) *Web-HMIs* (vgl. Abbildung 2-4, Mitte). Der Einzug webbasierter HMIs im Maschinen- und Anlagenbau ist nicht nur mit der einfachen Handhabung und der Integration der Software, sondern auch durch die Möglichkeit zur plattformübergreifenden Nutzbarkeit zu begründen (SCHRECK 2014, S. 51). Einerseits ist bei Web-HMIs keine Installation auf den Endgeräten notwendig, da der Zugriff über Internetbrowser erfolgt, andererseits ermöglichen Entwicklungsplattformen wie HTML 5 (HOGAN 2011) und JavaScript (FLANAGAN 2007) die Nutzung der Technologie über PCs und mobile Geräte (STEGERWALD 2014, S. 1 f.). In diesem Zusammenhang sind auch *Industrie-Apps* (vgl. Abbildung 2-4, rechts) als potenzielle Mensch-Maschine-Schnittstellen für die Produktionstechnik zu nennen. Der Grundgedanke ist dabei, dass Mitarbeiter zur Maschinenbedienung ihre eigenen mobilen Endgeräte verwenden und auf diesen das HMI als eigenständige Applikation installieren (GORECKY ET AL. 2017, S. 231 f.). Neben der ortsunabhängigen Visualisierung ergibt sich insbesondere der Vorteil, dass keine zusätzliche Hardware für die Interaktion benötigt wird (PEISSNER & HIPPEL 2013, S. 32).

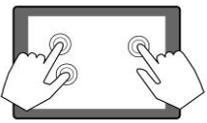
Multi-Touch	Web-HMI	Industrie-App
		
<ul style="list-style-type: none"> ▪ Gleichzeitige Erkennung mehrerer Berührungspunkte ▪ Nutzung von Gesten für Standardoperationen ▪ Hoher Verbreitungsgrad im Konsumgüterbereich 	<ul style="list-style-type: none"> ▪ Ausführung des HMIs über Internetbrowser ▪ Möglichkeit zur plattformübergreifenden Nutzung ▪ Keine Installation auf dem jeweiligen Endgerät 	<ul style="list-style-type: none"> ▪ Ausführung des HMIs als Applikation auf Smart-Device ▪ Möglichkeit zur ortsunabhängigen Visualisierung ▪ Keine eigene Hardware zur Interaktion notwendig

Abbildung 2-4: Aktuelle Trends bei industriellen Mensch-Maschine-Schnittstellen

Zur Umsetzung der zuvor beschriebenen Interaktionstechnologien gibt es bereits eine Vielzahl an *Softwarelösungen* bzw. Entwicklungsumgebungen (RICHTER & REINHART 2015, S. 262). Zur Implementierung von industriellen Mensch-

Maschine-Schnittstellen liefern sowohl die meisten Panel-Hersteller als auch Drittanbieter spezifische Softwarelösungen, die auf den Kontext einer Maschinenbedienung zugeschnitten sind, wie bspw. SIEMENS (2017). Demgegenüber besitzen integrierte Entwicklungsumgebungen (SHELLY & ROSENBLATT 2011, S. 656-658) oft eine größere Flexibilität hinsichtlich der zu realisierenden Funktionalität und des Designs der zu entwickelnden Mensch-Maschine-Schnittstelle. Innerhalb dieser Entwicklungsumgebungen kann meist auf spezifische Frameworks zurückgegriffen werden, die Bibliotheken für die HMI-Entwicklung bereitstellen (TAI 2007). Einen weiteren wichtigen Baustein zur Unterstützung der Implementierung von Mensch-Maschine-Schnittstellen bilden sog. Patterns (GAMMA ET AL. 2015), die bewährte Lösungen für wiederkehrende Entwurfsprobleme in der HMI-Entwicklung anbieten. Diese umfassen sowohl Muster hinsichtlich der zu wählenden Softwarearchitektur als auch Vorlagen für das konkrete Verhalten einer Applikation. Ein in der HMI-Entwicklung bekannter Vertreter ist bspw. das MVVM-Muster (Model, View, ViewModel), welches eine Trennung der Darstellung und Logik einer Mensch-Maschine-Schnittstelle vorsieht und dadurch die Arbeiten des HMI-Designers von denen des HMI-Entwicklers entkoppelt (SORENSEN & MIHAILESCU 2010).

2.2.2 Potenziale zukünftiger Interaktionstechnologien

Die im letzten Abschnitt vorgestellten Interaktionskonzepte besitzen eine generelle Gemeinsamkeit hinsichtlich der verwendeten Sinnesmodalitäten des Menschen: Über den visuellen Kanal (z. B. Bildschirme) erhalten Benutzer Informationen über den Maschinenzustand, während der Mensch seine Eingaben über den haptischen Kanal (z. B. Tasten) an das mechatronische System übermittelt (ZÜHLKE 2012, S. 217). Da die beiden Sinnesmodalitäten anhand der Richtung des Informationsflusses klar voneinander abgegrenzt werden können, spricht man in diesem Fall von einer unimodalen Interaktion (ZÜHLKE 2012, S. 217). Demgegenüber ermöglichen Konzepte der multimodalen Benutzerinteraktion die gleichzeitige und bidirektionale Informationsübermittlung über verschiedene Sinneskanäle, was grundsätzlich zu einer Verbesserung der Kommunikation zwischen Mensch und technischem System beiträgt (OVIATT 2012, S. 405). Zur Umsetzung multimodaler Interaktionskonzepte können verschiedene Technologien zum Einsatz kommen, die sich bislang vor allem im Konsumgüterbereich etablieren konnten. Beispielhaft hierfür werden nachfolgend die Konzepte der *Sprachsteuerung*, *Raumgestensteuerung* sowie *Augmented & Virtual Reality* vorgestellt, die im Unterschied zu gegenständlichen Mensch-Maschine-Schnittstellen (ISHII & ULLMER 2012) oder Gehirn-Computer-Schnittstellen (MILLAN 2002) in ersten industriellen Anwendungen erprobt werden.

Eine sprachliche Interaktion mit Maschinen ist bereits seit einigen Jahrzehnten Gegenstand der Forschung (KOLSTER 2014, S. 12; MARKOWITZ 1996). Das Grundprinzip der *Sprachsteuerung* (vgl. Abbildung 2-5, links) basiert auf der Erkennung festgelegter Kommandos oder natürlicher Sprache über Mikrofone sowie der Generierung von Sprachmeldungen über Lautsprecher oder Kopfhörer (SCHENK & RIGOLL 2010, S. 123). Da die Sprachkommunikation auch zwischen Menschen am häufigsten eingesetzt wird, besitzt diese Interaktionsform das größte Potenzial für den Dialog zwischen Mensch und technischem System (ZÜHLKE 2012, S. 230). Auch in Produktionsprozessen kann eine sprachliche Ein- und Ausgabe von Vorteil sein, wenn bspw. die Aufmerksamkeit des Nutzers oder seine motorischen Interaktionsmöglichkeiten situationsbedingt eingeschränkt sind (GORECKY ET AL. 2017, S. 223). Während sich Sprachsteuerungen für Büroanwendungen, wie bspw. LINGUATEC (2017), oder im Automobil bereits etablieren konnten (vgl. BENGLER ET AL. 2015A, S. 656), führen die spezifischen Randbedingungen¹ in der Industrie dazu, dass hier bislang nur prototypische Anwendungen umgesetzt werden konnten (KOLSTER 2014, S. 13 f.). Als Beispiel sei an dieser Stelle der Ansatz von REUTH ET AL. (2002) genannt, der die Umsetzung eines sprachlichen HMIs zur Unterstützung von Greif- und Spannprozessen beim 3D-Laserschweißen adressiert.

Neben der sprachlichen Komponente bilden Gesten eine wichtige Interaktionsform, über die Menschen miteinander kommunizieren (WACHSMUTH ET AL. 2008, S. 1 f.). Um diese für den Dialog mit technischen Systemen zu nutzen, wird an Ansätzen zur *Raumgestensteuerung* (vgl. Abbildung 2-5, Mitte) geforscht. Das Konzept basiert auf der Erkennung von 3D-Gesten des Benutzers, die in einem Messbereich erfasst und verarbeitet werden (KOLSTER 2014, S. 18 f.). Die Aufnahme kann dabei gerätebasiert (z. B. über Datenhandschuhe) oder kamerabasiert (z. B. über Tiefenkameras) erfolgen (TH WILDAU 2017). Neben der Erkennung von Handgesten als häufigstem Anwendungsfall (REN ET AL. 2011) gibt es auch Konzepte, die eine Aufnahme des ganzen Körpers oder der Augenregion fokussieren, wie bspw. SONG ET AL. (2011) oder JOOS ET AL. (2003). Auch im industriellen Umfeld konnte die Raumgestensteuerung bereits prototypisch eingesetzt werden. AUER (2012) beschreibt bspw. einen Ansatz, der mittels des Microsoft Kinect®-Sensors eine Gestensteuerung von Produktionsanlagen ermöglicht. LAMBRECHT (2014) nutzt hingegen 3D-Gesten für die Programmierung von Industrierobotern. Einer breiten Etablierung stehen aktuell allerdings noch Restriktionen bezüglich der korrekten Gestenerfassung gegenüber, die für eine industrielle Anwendung benötigt wird (NAUMANN ET AL. 2014, S. 516).

¹ z. B.: Erschwerung der Spracherkennung durch häufige Störgeräusche im Fabrikumfeld

Weitere Konzepte zur zukünftigen Maschinen- und Anlagenbedienung ergeben sich schließlich aus den Technologien der *Augmented* bzw. *Virtual Reality* (vgl. Abbildung 2-5, rechts). Diese besitzen die Gemeinsamkeit, dem Benutzer virtuelle Objekte zur Verfügung zu stellen, die er wahrnehmen und mit denen er zum Teil interagieren kann (REIF ET AL. 2007, S. 3). Während allerdings der Anwender bei der Virtual Reality vollständig in die virtuelle Welt eintaucht und mit dieser interagiert (BURDEA & COIFFET 2003, S. 2 f.), wird unter der Augmented Reality eine Anreicherung der realen Umgebung um digitale Informationen verstanden (TÖNNIS 2010, S. 1-3). Grundsätzlich bilden für beide Konzepte Technologien wie Head-up-Displays (NEWMAN 1995) oder Smartphones eine geeignete Basis. In Abhängigkeit des gewünschten Immersionsgrads in die virtuelle Welt werden aber auch weiterführende Technologien, wie Head-Mounted-Displays (BUBB 2015, S. 595) oder CAVEs (CRUZ-NEIRA ET AL. 1992), eingesetzt. Durch den Immersionsgrad ergeben sich auch im Hinblick auf die Anwendungsbereiche Unterschiede für beide Technologien: Virtual Reality bietet sich u. a. im Bereich der Simulation an, um Mitarbeiter in virtuellen Umgebungen zu schulen (STONE 2001). Ebenso kann durch Virtual Reality die Planung von z. B. Fahrzeugprototypen (PURSCHKE ET AL. 1998), Roboteranwendungen (ROBMANN 2012A) oder gesamten Fabriken (BRACHT & FAHLBUSCH 2001) unterstützt werden. Demgegenüber bietet sich der Einsatz von Augmented Reality vor allem an, wenn Personen auf digitale Unterstützung bei physischen Vorgängen, wie bspw. dem Führen eines Kraftfahrzeugs (BENGLER ET AL. 2015B) oder der Montage (REINHART & PATRON 2003) angewiesen sind bzw. generell Informationen zu vorhandenen Objekten benötigen (MEHLER-BICHLER & STEIGER 2014, S. 85 ff.). Eine breite industrielle Anwendung beider Technologien ist derzeit allerdings noch nicht möglich, was u. a. durch die noch nicht vollends ausgereifte Hardware zu begründen ist (WIESEND 2016; BLUM 2017, S. 3).

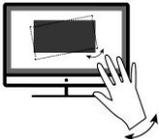
Sprachsteuerung	Raumgestensteuerung	Augmented & Virtual Reality
 <ul style="list-style-type: none"> ▪ Erkennung von Sprach-eingaben durch Mikrofone ▪ Generierung von Sprach-ausgaben über Lautsprecher ▪ Größtes Potenzial für die Mensch-Maschine-Interaktion 	 <ul style="list-style-type: none"> ▪ Erkennung und Auswertung von 3D-Gesten des Nutzers ▪ Geräte- oder Kamera-basierte Umsetzungsvarianten ▪ Anwendungsfälle: Hand-, Körper- und Augenerkennung 	 <ul style="list-style-type: none"> ▪ Wahrnehmung und Interaktion mit virtuellen Objekten ▪ AR: Anreicherung der Realität mit Informationen ▪ VR: Vollständiges Eintauchen in eine virtuelle Welt

Abbildung 2-5: Zukünftige Technologien zur Maschinen- und Anlageninteraktion

2.3 Softwareergonomie

Neben der Nutzung geeigneter Interaktionstechnologien ist es für die Akzeptanz von Mensch-Maschine-Schnittstellen von zentraler Bedeutung, diese unter Berücksichtigung ergonomischer Aspekte zu entwickeln (KOCH ET AL. 1991, S. 1). Dabei werden sowohl die anwendungsgerechte Gestaltung von Ein- und Ausgabegeräten sowie Arbeitsplätzen im Rahmen der Hardwareergonomie (SKROTZKI 2011) als auch die Umsetzung gebrauchstauglicher Programme im Rahmen der Softwareergonomie (HERCZEG 2009) adressiert. Während die Hardwareergonomie primär für die Hersteller von Ein- und Ausgabegeräten relevant ist, ist das Themenfeld der Softwareergonomie vor allem für den in dieser Arbeit fokussierten Maschinen- und Anlagenbau von großer Wichtigkeit. Dies lässt sich dadurch begründen, dass die Software zur Interaktion mit Maschinen und Anlagen von den Herstellern produktionstechnischer Systeme im Rahmen des Entwicklungsprozesses selbst implementiert bzw. projiziert wird, während die Ein- und Ausgabegeräte lediglich von entsprechenden Systemlieferanten eingekauft werden und somit hinsichtlich der Hardwareergonomie kaum beeinflusst werden können (ZÜHLKE 2012, S. 36).

Auf Grundlage dieser Vorüberlegung werden im Rahmen dieses Abschnitts die zum Verständnis dieser Arbeit relevanten Grundlagen der Softwareergonomie präsentiert. Aufbauend auf einer generellen Begriffsdefinition sowie einer Beschreibung der zentralen Ziele in Abschnitt 2.3.1, werden in Abschnitt 2.3.2 wichtige Qualitätsmerkmale einer gebrauchstauglichen Software vorgestellt. In Abschnitt 2.3.3 erfolgt letztlich eine Darstellung der für diese Arbeit relevanten Normen und Richtlinien.

2.3.1 Definition und Ziele der Softwareergonomie

Wie bereits OETINGER (1988, S. 82) herausgearbeitet hat, sind die meisten heutigen IT-Anwendungen durch den kontinuierlichen Dialog zwischen einem Benutzer und einem Computer bzw. einer konkreten Software gekennzeichnet. Die Software fungiert dabei als Werkzeug, mit dem der Benutzer interagiert, um eine spezielle Aufgabe, wie das Erstellen eines Dokuments, durchzuführen (HERCZEG 2009, S. 2). Um dieses Werkzeug bestmöglich verwenden zu können, hat sich seit Beginn der 1980er-Jahre das Themenfeld der Softwareergonomie entwickelt (HERCZEG 2009, S. 6-7). Über den Begriff des Werkzeugs lässt sich auch die Verbindung zum Wissenschaftsgebiet der Ergonomie im Allgemeinen herstellen, in dessen Rahmen bereits seit Mitte des 20. Jahrhunderts an der Anpassung und Optimierung von Werkzeugen und Arbeitsumgebungen an die Bedürfnisse des Menschen geforscht wird (BUBB 2000; BULLINGER 1994, S. 4). In diesem Zusammenhang sind auch die

Begriffe der Produkt- und Produktionsergonomie zu nennen, die sich zur Abgrenzung der unterschiedlichen Ansätze zur Gestaltung von Produkten und Arbeitsplätzen etabliert haben (BUBB 2012, S. 24). Obwohl viele generelle ergonomische Prinzipien (vgl. LAURIG 1990) auf Software übertragbar sind, rechtfertigt insbesondere die Interaktion bei Computer-Anwendungen die Etablierung der Softwareergonomie als wissenschaftliche Disziplin (HERCZEG 2009, S. 6). Dabei hat sich in weiten Teilen der Forschung die folgende Definition von RÖDIGER (1987) für den Begriff der Softwareergonomie durchgesetzt.

„Die Software-Ergonomie ist die Lehre von der Anpassung eines dialogfähigen Arbeitssystems an die kognitiven und intellektuellen Eigenschaften des Menschen, der in einem organisatorischen Kontext arbeitet. (...) Das Ziel der Software-Ergonomie ist es, Werkzeuge (Programme) zu schaffen, die von unterschiedlichen Benutzern innerhalb des organisatorischen Kontextes ihrer Aufgabenstellung entsprechend eingesetzt werden können.“ (RÖDIGER 1987, S. 160)

Die obige Definition wird von vielen Experten auf dem Gebiet der Softwareergonomie hinsichtlich der konkreten Ziele noch verfeinert. KOCH ET AL. (1991) nennen als zentrale Ziele die Gewährleistung einer effizienten Aufgabenausführung, die Minimierung möglicher negativer physischer und psychischer Auswirkungen sowie die Erhaltung und Verbesserung der Entwicklungsmöglichkeiten des Menschen (KOCH ET AL. 1991, S. 47). HERCZEG (2009) hingegen prägt für das Hauptziel der Softwareergonomie den Begriff der Gebrauchstauglichkeit, welche durch die Erfüllung der Basiskriterien Effektivität, Effizienz und Zufriedenstellung bestimmt wird (HERCZEG 2009, S. 7-9). Unter *Effektivität* versteht HERCZEG (2009) in diesem Kontext die möglichst vollständige und korrekte Erfüllung der Aufgaben. *Effizienz* hingegen adressiert die Minimierung des Aufwands, um diese Aufgaben möglichst zuverlässig und ohne Fehler durchzuführen. Die *Zufriedenstellung* wird als die Freiheit von Beeinträchtigungen und die positive Einstellung gegenüber der Nutzung der Software definiert (HERCZEG 2009, S. 7-9; DIN EN ISO 9241-11, S. 16-20).

Um die eben beschriebenen Ziele und Basiskriterien erreichen zu können, widmet sich die Softwareergonomie primär der Aufgabe, Qualitätsmerkmale (vgl. Abschnitt 2.3.2) zu definieren, deren Beachtung eine hohe Gebrauchstauglichkeit von interaktiver Software sicherstellen soll (HERCZEG 2009, S. 13). Durch diese Fokussierung lässt sich die Softwareergonomie vom Themenfeld des Usability Engineerings (vgl. NIELSEN 1993) abgrenzen, in dessen Rahmen an Gestaltungsprozessen, Modellen und Methoden geforscht wird, um gebrauchstaugliche Software gemäß diesen Qualitätsmerkmalen entwickeln zu können.

2.3.2 Zentrale Qualitätsmerkmale ergonomischer Software

Auf Basis der zuvor beschriebenen Basiskriterien für die Gebrauchstauglichkeit von Software (Effektivität, Effizienz und Zufriedenstellung) lassen sich weiterhin konkrete Qualitätsmerkmale ableiten, die für das Verständnis interaktiver Systeme und die benutzerzentrierte Entwicklung von großem Nutzen sind (HERCZEG 2009, S. 155). Eine geeignete Grundlage, um diese Merkmale zu strukturieren und dadurch unterschiedliche Perspektiven bei der Bewertung einer interaktiven Software einzunehmen, bietet das IFIP²-Modell für Mensch-Maschine-Schnittstellen (DZIDA 1988). Dieses Modell gliedert HMIs in eine *Ein-/Ausgabe*-, eine *Dialog*- und eine *Werkzeugschnittstelle* und stellt zudem eine Verbindung des Benutzers bzw. des Rechners zum jeweiligen Arbeitsumfeld her (vgl. Abbildung 2-6).

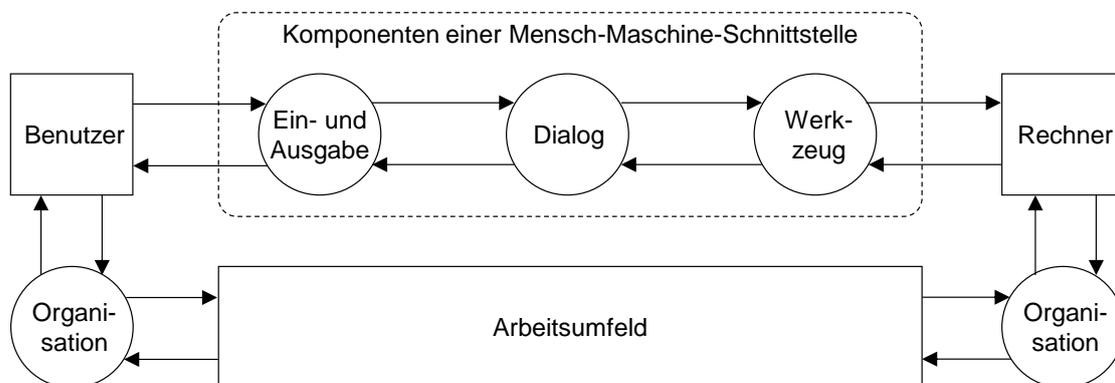


Abbildung 2-6: IFIP-Modell für Mensch-Maschine-Schnittstellen
(in Anlehnung an DZIDA 1988)

Jeder dieser drei Schnittstellen lassen sich spezifische Qualitätsmerkmale zuordnen, um sie näher zu charakterisieren und somit differenziertere Analysen und Bewertungen von Mensch-Maschine-Schnittstellen vorzunehmen zu können (HERCZEG 2009, S. 156). Abbildung 2-7 gibt einen Überblick der einzelnen Qualitätsmerkmale, die sich im Kontext der Softwareergonomie etabliert haben. Bei der Entwicklung dieser Merkmale wurde auf eine allgemeingültige Formulierung geachtet, so dass diese anwendungs- und branchenunabhängig genutzt werden können. Dies macht teilweise eine Spezialisierung für einzelne Anwendungsfelder, wie z. B. den Maschinen- und Anlagenbau, erforderlich (vgl. Abschnitt 2.3.3), aber dennoch unterstützt eine Berücksichtigung der nachfolgend vorgestellten Merkmale grundsätzlich die Umsetzung ergonomischer bzw. gebrauchstauglicher Mensch-Maschine-Schnittstellen.

² International Federation for Information Processing

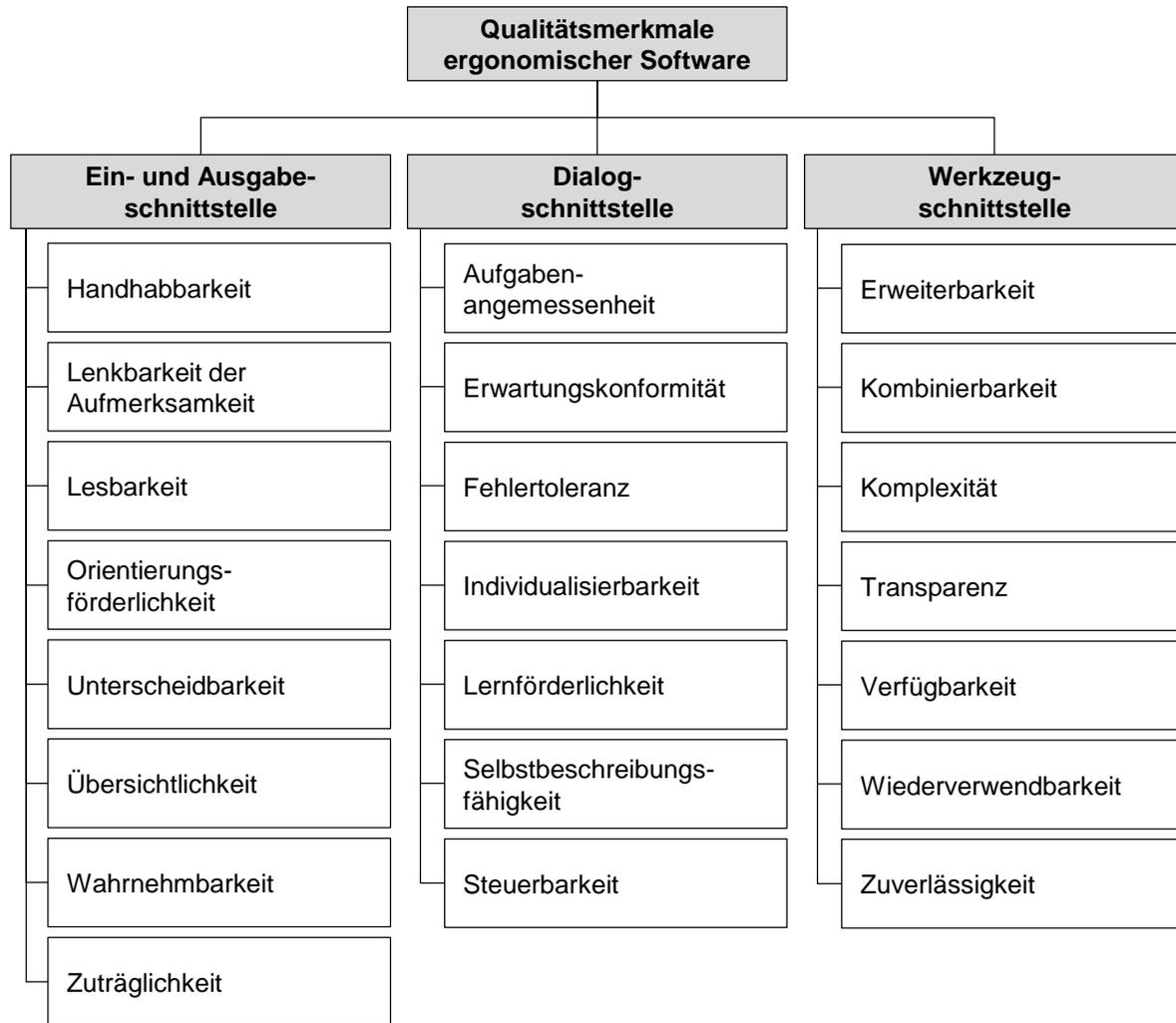


Abbildung 2-7: Qualitätsmerkmale ergonomischer Software (in Anlehnung an HERCZEG 2009, KOCH ET AL. 1991 und DIN EN ISO 9241-110)

In der *Ein- und Ausgabeschnittstelle* eines interaktiven Systems werden zunächst Regeln für die Ausgaben des Systems und die Eingaben des Benutzers definiert. Somit wird festgelegt, wie spezifische Informationen eingegeben oder dem Nutzer übermittelt werden sollen. Die hierfür in Abbildung 2-7 dargestellten Qualitätsmerkmale adressieren dabei u. a. minimale bzw. einheitliche Schriftgrößen, zu verwendende Farbschemata und Layouts oder die Platzierung von Bedien- und Anzeigeelementen. Demgegenüber widmen sich die Qualitätsmerkmale der *Dialog-schnittstelle* der dialogtechnischen Abbildung der Funktionalität eines Systems. Dialoge werden dabei den oben einzelnen Qualitätsmerkmalen gerecht, wenn sie bspw. dem Benutzer zu jeder Zeit offensichtlich sind, das Erlernen einer Software unterstützen oder sich in Richtung und Geschwindigkeit an den Bedürfnissen des Benutzers orientieren. In der *Werkzeugschnittstelle* wird schließlich die Funktionalität der Mensch-Maschine-Schnittstelle selbst adressiert. So sollen Benutzer

u. a. auf die ganze Funktionalität eines Systems zugreifen und diese Funktionalität auch ohne Eingriff in dessen Programmierung erweitern können. Weiterhin wird in diesem Zusammenhang auch eine Vorhersehbarkeit der Reaktionen des Systems für den Benutzer gefordert. Für eine detaillierte Beschreibung aller in Abbildung 2-7 dargestellten Qualitätsmerkmale von Mensch-Maschine-Schnittstellen sei an dieser Stelle auf HERCZEG (2009, S. 163-194), DIN EN ISO 9241-110 (S. 7-17) oder KOCH ET AL. (1991, S. 80-81) verwiesen.

2.3.3 Relevante Normen und Richtlinien

Die eben eingeführten Qualitätsmerkmale wurden bereits größtenteils in nationale und internationale Normen und Richtlinien überführt, wobei sich die DIN EN ISO 9241 als zentrale Norm etablieren konnte. Die Allgemeingültigkeit der dort formulierten Qualitätsmerkmale macht es allerdings notwendig, auch spezifische Normen und Richtlinien für konkrete Technologien oder Anwendungsgebiete zu definieren. Für die Gestaltung von Mensch-Maschine-Schnittstellen technischer Anlagen wurde bspw. die VDI/VDE 3850 angefertigt. Eine weitere Detaillierung bilden sog. Styleguides, die auf einen konkreten Anwendungsfall, wie z. B. spezielle Produkte oder Produktgruppen, zugeschnitten sind und dadurch den höchsten Detailgrad besitzen (STONE ET AL. 2005, S. 169-170). Die meisten dieser Styleguides werden direkt in Unternehmen entwickelt, es existieren aber auch wissenschaftliche Ansätze, wie der OSACA/HÜMNOS Styleguide (FRAUNHOFER IAO 1997) für die Gestaltung der HMI-Panels von Werkzeugmaschinen. Diese drei Beispiele, die für die vorliegende Arbeit die größte Relevanz besitzen, werden im Folgenden kurz vorgestellt. Für einen breiteren Überblick hinsichtlich vorhandener Normen und Richtlinien im Umfeld der Softwareergonomie sei auf KOLSTER (2014, S. 28-31) verwiesen, der zudem die Relevanz der einzelnen Ansätze hinsichtlich einer Anwendung im Maschinen- und Anlagenbau herausstellt.

DIN EN ISO 9241

Der internationale Standard DIN EN ISO 9241 „*Ergonomie der Mensch-System-Interaktion*“ ist nach ZÜHLKE (2012, S. 178) die bedeutendste Norm zur Entwicklung von Mensch-Maschine-Schnittstellen. In dieser sind weite Teile des weltweit verfügbaren Wissens zur Konzeption, Gestaltung und Bewertung von Mensch-Maschine-Schnittstellen zusammengeführt, wobei neben der Software auch die ergonomische Auslegung der Hardware und den jeweiligen Arbeitsumgebungen betrachtet wird. Dazu gliedert sich die Norm in 37 Teile, wobei das Spektrum von Grundsätzen zur Dialoggestaltung bis zu Anforderungen an elektronische optische

Anzeigen reicht. Ursprünglich adressierte die Norm lediglich Bürotätigkeiten mit Bildschirmgeräten, jedoch wurde von Beginn an auf eine allgemeingültige Formulierung der Richtlinien und Gestaltungsgrundsätze geachtet. Dadurch lässt sich die DIN EN ISO 9241 auf Bereiche, wie den Maschinen- und Anlagenbau, übertragen, die sich von ihren Randbedingungen und eingesetzten Interaktionstechnologien vom Büroumfeld unterscheiden (ZÜHLKE 2012, S. 178).

Den spezifischen Fragestellungen der Softwareergonomie wird in der DIN EN ISO 9241 insbesondere in den Teilen 11-16, 110, 129 und 143 begegnet. Dabei umfasst Teil 11 zunächst grundsätzliche Leitsätze hinsichtlich der Gebrauchstauglichkeit von Software, wobei u. a. die drei Basiskriterien Effektivität, Effizienz und Zufriedenstellung definiert und beschrieben werden. Weiterhin werden in Teil 110 (Grundsätze zur Dialoggestaltung) die Qualitätsmerkmale der Dialogschnittstelle (vgl. Abschnitt 2.3.2) anhand von Beispielen detailliert erläutert. Darauf aufbauend befassen sich die weiteren genannten Teile mit spezifischeren Fragestellungen, wie bspw. der Dialogführung mittels Menüs (Teil 14) oder Formularen (Teil 143).

VDI/VDE-Richtlinie 3850

Der allgemeingültige Charakter der DIN EN ISO 9241 macht es erforderlich, für spezifische Anwendungsfelder und Branchen auch detailliertere Richtlinien als die generischen Qualitätsmerkmale ergonomischer Software zu definieren (HERCZEG 2009, S. 158). Für den Maschinen- und Anlagenbau ist hierbei besonders die VDI/VDE-Richtlinie 3850 mit dem Titel „*Gebrauchstaugliche Gestaltung von Benutzungsschnittstellen für technische Anlagen*“ relevant. Die drei Blätter der Richtlinie befanden sich seit 2010 in Überarbeitung, liegen seit Juli 2015 aber vollständig in einer erweiterten Fassung vor.

Blatt 1 der VDI/VDE 3850 thematisiert zunächst grundsätzliche Prinzipien und Empfehlungen zur Gestaltung von Mensch-Maschine-Schnittstellen. Dabei wird insbesondere auf die Bereiche der Aufgaben-, Dialog- und Informationsdarstellung sowie die Auswahl und Kombination von Dialogtechniken eingegangen. Darüber hinaus wird ein möglicher Prozess beschrieben, um gebrauchstaugliche Mensch-Maschine-Schnittstellen für technische Anlagen zu realisieren. Demgegenüber werden in Blatt 2 der Richtlinie mögliche Eingabegeräte (Maus, Trackball, etc.) thematisiert, die bei einer Informationsdarstellung durch Bildschirme zum Einsatz kommen können. Neben einer Klassifikation von Eingabegeräten und Empfehlungen zum Einbau werden auch Anforderungen (z. B. der Nutzer und des Nutzungskontexts) formuliert, die Rückschlüsse hinsichtlich der Einsatzpotenziale

einzelner Eingabegeräte erlauben. Blatt 3 der Richtlinie beschäftigt sich schließlich speziell mit Touchscreens als Interaktionstechnologie und trägt damit der Tatsache Rechnung, dass diese Technologie im industriellen Umfeld stark an Bedeutung gewonnen hat (ZÜHLKE 2012, S. 212). Basierend auf einer Darstellung der Stärken und Schwächen von touchbasierten Mensch-Maschine-Schnittstellen sowie grundsätzlichen Rahmenbedingungen werden insbesondere Aspekte der Gestaltung, wie bspw. die Größe und Anordnung von Bedienelementen, diskutiert. Einen letzten wichtigen Aspekt bildet schließlich die Migration von bestehenden Mensch-Maschine-Schnittstellen auf touchbasierte Systeme.

OSACA/HÜMNOS Styleguide Werkzeugmaschinen

Als Beispiel eines Styleguides im Umfeld von Mensch-Maschine-Schnittstellen des Maschinen- und Anlagenbaus wird schließlich der OSACA/HÜMNOS Styleguide (FRAUNHOFER IAO 1997) vorgestellt. Der im Rahmen des HÜMNOS-Forschungsprojekts (PRITSCHOW 1997) veröffentlichte Leitfaden verfolgt das generelle Ziel, „(...) die Gestaltung von CNC-Benutzungsoberflächen von Werkzeugmaschinen so zu vereinheitlichen, dass der Benutzer der Oberfläche ohne Umgewöhnung die Steuerungen verschiedener Hersteller bedienen kann“ (FRAUNHOFER IAO 1997). Um diesem Ziel gerecht zu werden, bietet der Styleguide Handlungsempfehlungen für die Gestaltung von benutzer- und aufgabengerechten HMI-Panels in drei verschiedenen Bereichen. Im ersten Teil wird ein Regelwerk beschrieben, welches die allgemeinen Qualitätsmerkmale ergonomischer Software (vgl. Abschnitt 2.3.2) referenziert, diese aber um zusätzliche Gestaltungsprinzipien, wie z. B. für Bedienbereiche, ergänzt. Des Weiteren werden im Regelwerk des Styleguides konkrete Richtlinien zur Farb-, Bildschirm- und Dialoggestaltung sowie zur Informationsanzeige für Mensch-Maschine-Schnittstellen von Werkzeugmaschinen angeboten. Der zweite Teil des Styleguides fokussiert demgegenüber die einheitliche Verwendung von Begrifflichkeiten im Kontext von Werkzeugmaschinen. In einem Lexikon werden rund 350 Begriffe im Umfeld der Fertigungstechnik erläutert und um entsprechende Bildzeichen und Abkürzungen ergänzt. Im dritten Teil des Styleguides werden schließlich sogenannte Basisdialoge definiert, die in allen Steuerungen von Werkzeugmaschinen auftreten und mittels des Styleguides entsprechend gebrauchstauglich und einheitlich umgesetzt werden können.

2.4 Modellbildung und Simulation

Wie bereits in Kapitel 1 dargelegt, basiert das im Rahmen dieser Arbeit erarbeitete Vorgehensmodell auf der Verwendung von Modellen zur interdisziplinären Entwicklung von Mensch-Maschine-Schnittstellen im Kontext mechatronischer Systeme. Daher werden im folgenden Abschnitt die Grundlagen zur Modellbildung und Simulation sowohl von mechatronischen Systemen im Allgemeinen als auch Mensch-Maschine-Schnittstellen im Speziellen beschrieben. Nach einer generellen Beschreibung der in diesem Zusammenhang relevanten Begriffe in Abschnitt 2.4.1, werden in Abschnitt 2.4.2 Möglichkeiten aufgezeigt, wie mechatronische Systeme und Mensch-Maschine-Schnittstellen modelliert werden können. Abschließend werden in Abschnitt 2.4.3 Ansätze zur Simulation selbiger vorgestellt.

2.4.1 Beschreibung des System-, Modell- und Simulationsbegriffs

In den folgenden Teilabschnitten werden die Begriffe des Systems, des Modells und der Simulation definiert und beschrieben. Da für diese Begriffe vielfältige Definitionen existieren, die maßgeblich vom jeweiligen Untersuchungsgegenstand abhängen³, beziehen sich die folgenden Ausführungen auf die VDI 3633, in der die Begriffe speziell im Kontext technischer Systeme definiert und beschrieben werden.

Systembegriff

Der Begriff des Systems wurde im bisherigen Verlauf der Arbeit primär im Kontext mechatronischer und interaktiver Systeme verwendet, die jeweils spezifisch anhand der VDI 2206 bzw. der DIN EN ISO 9241-110 definiert wurden. Zum besseren Verständnis der Modellbildung und Simulation wird an dieser Stelle zudem der Systembegriff im Allgemeinen definiert: Gemäß der VDI 3633 – BLATT 1 (S. 4) lässt sich ein *System* als eine von ihrer Umwelt abgegrenzte Menge von Elementen beschreiben, die miteinander in Beziehung stehen und durch Regeln und Attribute charakterisiert werden. Neben einem konkreten Verhalten und Zustand ist ein System insbesondere durch eine Systemgrenze gekennzeichnet, wobei über Schnittstellen Ein- und Ausgangsgrößen mit der jeweiligen Umwelt ausgetauscht werden. Ein zentrales Charakteristikum des Systemdenkens ist die Zerlegung von Systemen in Subsysteme und nicht weiter zerlegbare Elemente, die beide über Schnittstellen miteinander gekoppelt sein können und somit in Wechselwirkung stehen. Abbildung 2-8 veranschaulicht die eben eingeführten Merkmale anhand eines beispielhaften Systems mit Subsystemen, Elementen und Schnittstellen.

³ Siehe bspw. GÜNTHER & VELTEN (2014, S. 4) oder HELLER (2012, S. 20)

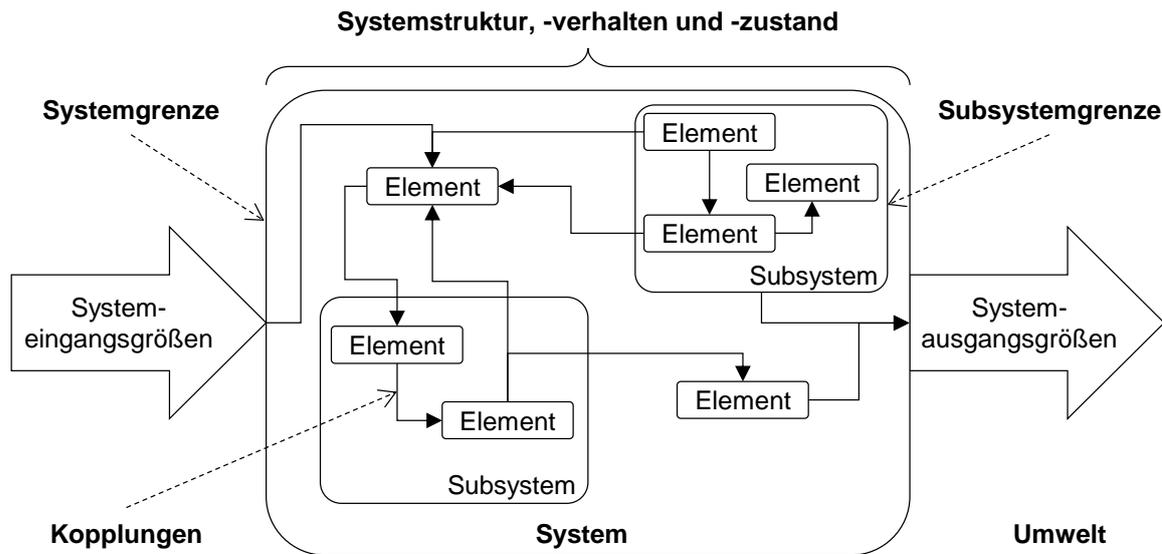


Abbildung 2-8: Grundstruktur eines Systems nach VDI 3633

Modellbegriff

Eng verknüpft mit dem Systemgedanken ist der Begriff des Modells. Modelle werden grundsätzlich zur Abstraktion realer Systeme benötigt, um die Komplexität in technischen Systemen beherrschen zu können (HENSEL 2011, S. 24). Vor diesem Hintergrund definiert die VDI 3633 ein *Modell* als eine „(...) vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System“ (VDI 3633 – BLATT 1, S. 3). Diese Definition macht deutlich, dass ein Modell nicht alle Eigenschaften eines Systems beinhalten muss, sondern lediglich diejenigen, die für den jeweiligen Modellierungszweck notwendig sind (BUNGARTZ ET AL. 2013, S. 4 ff.).

Im Kontext von Modellen sind für das Verständnis dieser Arbeit weiterhin die Begriffe Modellierungstechnik, Metamodell und Modelltransformation von großer Relevanz. In einer *Modellierungstechnik* werden nach HOLTEN (2000, S. 4-7) die Bestandteile definiert, aus denen sich ein konkretes Modell zusammensetzen darf. Zur Beschreibung solcher Modellierungstechniken werden üblicherweise selbst Modelle eingesetzt, die dann als *Metamodelle* bezeichnet werden (ALT 2012, S. 24).

Dagegen lässt sich eine *Modelltransformation* nach MARSCHALL (2005, S. 29-30) als die Umwandlung eines Quellmodells in ein Zielmodell definieren. Dabei wird auf eine Reihe an Transformationsregeln zurückgegriffen, die beschreiben, welche Elemente eines Quellmodells in ein Zielmodell übernommen werden und welche Rahmenbedingungen bei der Übertragung beachtet werden müssen (CZARNECKI & HELSEN 2003, S. 3-4).

Simulationsbegriff

Neben dem grundsätzlichen Nutzenpotenzial von Modellen, Systeme über abstrahierte Abbilder der Realität zu veranschaulichen, werden diese meist auch dazu eingesetzt, um das Verhalten von Systemen zu analysieren und zu evaluieren (BUNGARTZ ET AL. 2013, S. 7 ff.). Generell bezeichnet man ein solches Experimentieren mit Modellen als *Simulation* (GUALA 2002, S. 63). In der VDI 3633 wird dieser Begriff demensprechend definiert als 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 3633 – BLATT 1, S. 3).

Diese allgemeingültige Definition des Simulationsbegriffs lässt grundsätzlich offen, in welchem Umfeld eine Simulation ausgeführt werden kann, so dass auch physikalische Experimente, wie z. B. PKW-Crashtests, als Simulationen verstanden werden können (GUALA 2002, S. 63). Im Kontext dieser Arbeit hingegen bezieht sich der Simulationsbegriff ausschließlich auf die *Computersimulation*, die sich nach HARTMANN (1996, S. 81) als die Durchführung von Simulationen mit Hilfe eines Computers bzw. eines Computerprogramms definieren lässt.

2.4.2 Modellbildung bei mechatronischen Systemen und Mensch-Maschine-Schnittstellen

Im Rahmen der Konzeption und Umsetzung von mechatronischen Systemen können unterschiedliche Modelle zur Realisierung einzelner Entwicklungsaufgaben eingesetzt werden (ANACKER & GAUSEMEIER 2014, S. 9). Das Spektrum reicht von Modellen für disziplinspezifische Fragestellungen bis zu ganzheitlichen interdisziplinären Ansätzen. In den folgenden Abschnitten wird ein Überblick zentraler Modelle bei der Entwicklung mechatronischer Systeme gegeben, wobei speziell die interdisziplinäre Modellbildung in der frühen Entwicklungsphase fokussiert wird. Darauf aufbauend wird beschrieben, welche Modelle die Entwicklung von Mensch-Maschine-Schnittstellen als Teile mechatronischer Systeme unterstützen können.

Modellbildung bei mechatronischen Systemen

Wie bereits zuvor erwähnt, sind an der Entwicklung mechatronischer Systeme im Wesentlichen die Disziplinen Mechanik, Elektronik und Software beteiligt. Für deren individuelle Entwicklungsaufgaben haben sich unterschiedliche *disziplinspezifische Modelle* etabliert. So werden bspw. in der Mechanik 3D-CAD-Modelle verwendet, um die geometrische Konstruktion eines zu entwickelnden Systems zu

unterstützen (HEHENBERGER 2011, S. 123 ff.). Demgegenüber werden im Bereich der Elektronik Modelle u. a. zum Entwurf von Schaltplänen oder Leiterplatten eingesetzt (ZAFIROV & ROUBANOV 2014, S. 137-160). In der Softwareentwicklung werden Modelle schließlich angewandt, um ein zu implementierendes Programm auf abstrakter Ebene mittels Diagrammen oder Ähnlichem zu spezifizieren (KEBLER 2014, S. 161-174). Hierfür wurden eigene Modellierungstechniken erarbeitet, deren bekanntester Vertreter die Unified Modeling Language ist (BOOCH ET AL. 2005).

Der spezielle Charakter mechatronischer Systeme (vgl. Abschnitt 2.1) macht es allerdings erforderlich, auch *interdisziplinäre Modelle* im Laufe des Entwicklungsprozesses einzusetzen. Speziell in der frühen Entwicklungsphase, in der zunächst ein gemeinsames Verständnis über das zu erarbeitende System zwischen den beteiligten Disziplinen aufgebaut werden muss, bietet sich die Verwendung solcher Modelle an (EIGNER ET AL. 2012, S. 13). Dabei sind insbesondere Funktionsmodelle von Bedeutung, die eine formale Beschreibung, Strukturierung und Hierarchisierung der Systemfunktionen ermöglichen (BECERRIL ET AL. 2016, S. 700). Gängige Ansätze in diesem Kontext werden bspw. von PAHL ET AL. (2003, S. 214-226) oder von KALLMEYER (1998) beschrieben. Letzterer macht aber deutlich, dass neben Funktionen auch weiterführende Aspekte mechatronischer Systeme berücksichtigt bzw. modelliert werden müssen, um einen zielgerichteten Systementwurf zu ermöglichen (KALLMEYER 1998, S. 81 ff.). Vor diesem Hintergrund wird bspw. von DOROCIAK ET AL. (2014) die Modellierungstechnik CONSENS zur Beschreibung der Prinziplösung mechatronischer Systeme vorgeschlagen. Diese Technik setzt sich aus acht Partialmodellen zusammen, wobei die zuvor erwähnten Funktionen eines dieser Modelle darstellen. Hinzu kommen weitere Modelle zur Abbildung von Anforderungen, Anwendungsszenarien oder Wirkstrukturen, die über partialmodellübergreifende Beziehungen vernetzt sind (DOROCIAK ET AL. 2014, S. 199). Eine vergleichbare Modellierungstechnik, die im industriellen Umfeld jedoch weiter verbreitet ist (KERNSCHMIDT & VOGEL-HEUSER 2013, S. 1114), bildet die Systems Modeling Language (SysML⁴). Diese Technik stellt eine Erweiterung der auf Softwaresysteme beschränkten UML dar und setzt sich aus vier zentralen Aspekten zur Abbildung komplexer Systeme zusammen (FRIEDENTHAL ET AL. 2015, S. 17). Abbildung 2-9 gibt einen Überblick dieser Aspekte, die eine interdisziplinäre Systemmodellierung in der frühen Entwicklungsphase ermöglichen.

⁴ Für die SysML wurden bereits Erweiterungen, wie die SysML4Mechanics (FELDMANN ET AL. 2014), erarbeitet, um die Rahmenbedingungen spezifischer Systeme berücksichtigen zu können.

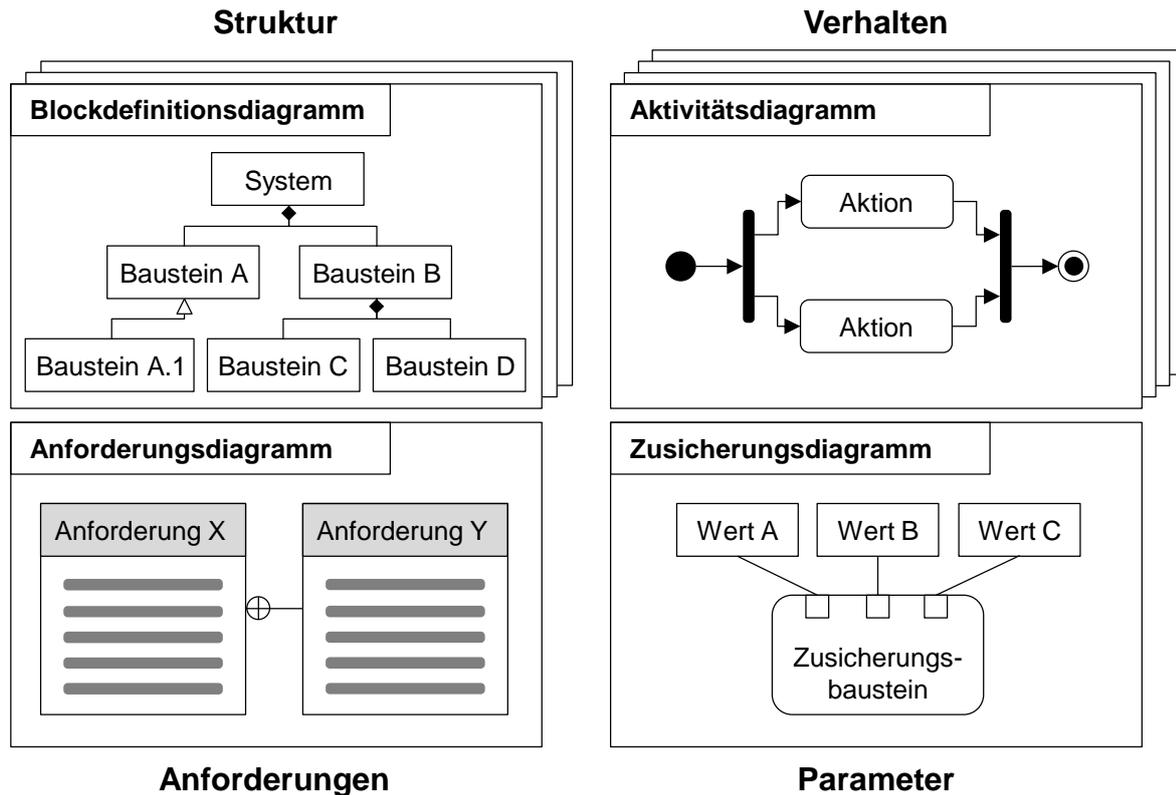


Abbildung 2-9: Die vier zentralen Aspekte der SysML
(in Anlehnung an FRIEDENTHAL ET AL. 2015, S. 17)

Der erste Aspekt ist dabei die *Struktur* eines zu entwickelnden Systems. Hierfür bietet die SysML drei verschiedene Diagrammtypen an, u. a. das Blockdefinitionsdiagramm. Dieses ermöglicht die abstrakte Beschreibung von Systembausteinen sowie deren Abhängigkeiten über Mechanismen wie Aggregation, Komposition oder Generalisierung. Auch zur Modellierung des *Verhaltens* von Systemen oder einzelnen Bestandteilen bietet die SysML unterschiedliche Diagrammtypen an. In Aktivitätsdiagrammen wird bspw. ein konkretes Verhalten über die Vernetzung von elementaren Aktionen und deren Verbindungen mit Kontroll- und Datenflüssen definiert. Über zusätzliche Modellierungselemente, wie Entscheidungs- oder Parallelisierungsknoten lassen sich auch komplexe Verhaltensmuster abbilden. Einen dritten zentralen Modellierungsaspekt der SysML bilden *Anforderungen*. In dem zugehörigen Diagramm werden diese über eine eindeutige ID und eine textuelle Beschreibung definiert sowie untereinander verknüpft, um auch mögliche Abhängigkeiten darstellen zu können. Den letzten zentralen Aspekt bilden *Parameter*, die in dem sog. Zusicherungsdiagramm beschrieben werden. Die zentrale Aufgabe dieser Diagrammart ist die Modellierung von parametrischen Beziehungen innerhalb oder zwischen einzelnen Systembausteinen. (ALT 2012, S. 40-59; FRIEDENTHAL ET AL. 2015, S. 29-38)

Modellbildung bei Mensch-Maschine-Schnittstellen

Die Entwicklung von Mensch-Maschine-Schnittstellen im Kontext mechatronischer Systeme ist maßgeblich durch die Umsetzung von Software geprägt. Obwohl sich hierfür bereits vielfältige Modellierungstechniken, wie z. B. die UML, etablieren konnten, macht es der Charakter von HMIs erforderlich, auch spezifischere Modelle im Verlauf des Entwicklungsprozesses einzusetzen, um eine hohe Gebrauchstauglichkeit der zu entwickelnden Software sicherzustellen (ZÜHLKE 2012, S. 59 f.). Es sei an dieser Stelle erwähnt, dass die nachfolgend vorgestellten Modelle zwar generell die Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau unterstützen können, aufgrund der geringen Fokussierung von HMIs in der Produktionstechnik bislang allerdings kaum genutzt werden. Das Haupteinsatzgebiet liegt aktuell im Bereich von Computeranwendungen.

Hier bilden *graphische Modelle* einer Mensch-Maschine-Schnittstelle eine erste Möglichkeit zur Unterstützung des Entwicklungsprozesses. Diese ermöglichen eine Abstimmung mit Kunden bzw. späteren Anwendern, eignen sich aber ebenfalls zur internen Dokumentation und Abstimmung im Entwicklungsteam (RICHTER & FLÜCKIGER 2013, S. 52 ff.). Je nach dem konkreten Inhalt lassen sich graphische Modelle weiterhin unterteilen in Wireframes, Mock-ups und Prototypen (TREDER 2016). Unter *Wireframes* versteht man dabei einfache schematische Zeichnungen, mittels derer einzelne Bedien- und Anzeigeelemente von HMIs auf abstrakter Ebene definiert und positioniert werden können. Im Fokus stehen primär das generelle Layout und das Navigationskonzept, während Detailspekte wie die spätere Farbgestaltung nicht betrachtet werden (PREIM & DACHSELT 2015, S. 119). Dagegen werden *Mock-ups* eingesetzt, um ein möglichst exaktes graphisches Abbild einzelner Ausschnitte eines HMIs zu modellieren. Es handelt sich dabei um statische Graphiken, die mit speziellen Designwerkzeugen erstellt werden (CAO 2017). Schließlich werden bei der Entwicklung von Mensch-Maschine-Schnittstellen auch *Prototypen* eingesetzt, um ein HMI hinsichtlich seiner Gebrauchstauglichkeit evaluieren zu können. Im Unterschied zu Wireframes und Mock-ups sind Prototypen in der Regel ablauffähige Modelle, die jeweils einen konkreten Ausschnitt einer Mensch-Maschine-Schnittstelle hinsichtlich Funktion und Design abbilden (PREIM & DACHSELT 2015, S. 120-122).

Neben rein graphischen Modellen lassen sich bei der Entwicklung von Mensch-Maschine-Schnittstellen auch umfassende *Modellierungstechniken* einsetzen, die eine ganzheitliche Abbildung von HMIs zum Ziel haben und sich nicht auf visuelle Aspekte beschränken. Ausgewählte Beispiele solcher Modellierungstechniken sind

u. a. die UsiXML (VANDERDONCKT 2008), die UMLi (DA SILVA & PATON 2000) oder die UseML (REUTHER 2003). Diese Techniken basieren auf drei zentralen Modellen, die für eine zielgerichtete modellbasierte Entwicklung von Mensch-Maschine-Schnittstellen benötigt werden (LUYTEN 2004, S. 18; ZÜHLKE 2012, S. 161 f.). Im Einzelnen handelt es sich hierbei um das Aufgaben-, das Dialog- und das Präsentationsmodell (vgl. Abbildung 2-10). Im *Aufgabenmodell* werden zunächst die Benutzungsaufgaben einzelner Nutzer spezifiziert und hierarchisch strukturiert (ZÜHLKE 2012, S. 161). Im Ansatz von MEIXNER & GÖRLICH (2008) kommen dabei elementare Benutzungsobjekte (wie „Informieren“ oder „Auslösen“) zum Einsatz, mittels derer die Benutzungsaufgaben formal aus elementaren Grundaufgaben zusammengesetzt werden können. Auf der Basis modellierter Aufgaben kann im *Dialogmodell* der zeitliche Ablauf von Interaktionen mit Ereignissen sowie Bedien- und Anzeigeelementen beschrieben werden (ZÜHLKE 2012, S. 162). Dabei können Zustandsautomaten oder Petrinetze zur Modellierung eingesetzt werden (GÖTZE 1995), es existieren aber auch spezifischere Ansätze wie bspw. das Guilet-Dialogmodell von RÜCKERT & PAECH (2009). Im *Präsentationsmodell* wird schließlich die Darstellung des HMIs definiert. Es wird grundsätzlich spezifiziert, welche visuellen, haptischen oder auditiven Interaktionsobjekte eingesetzt werden, um Benutzungsaufgaben bzw. -dialoge zu realisieren (ZÜHLKE 2012, S. 162). Neben der Spezifikation von Interaktionsobjekten und deren Abhängigkeiten mit bspw. Klassendiagrammen umfassen Präsentationsmodelle in der Regel auch einfache graphische Modelle (vgl. Wireframes), um das generelle Layout und die Positionierung von Interaktionsobjekten festzulegen (VANDERDONCKT 2008, S. 3).

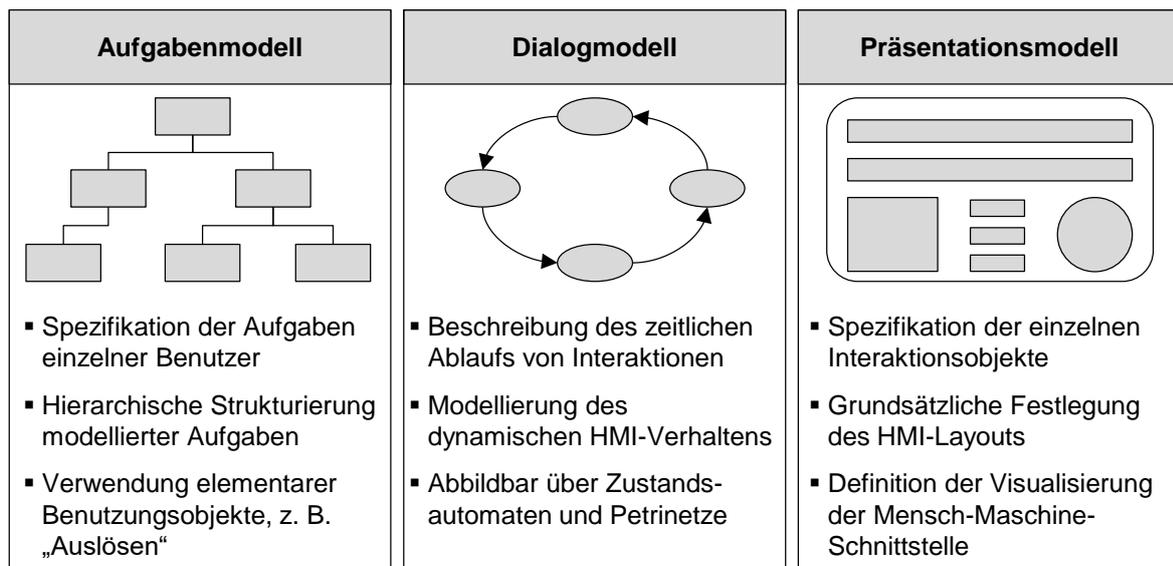


Abbildung 2-10: Modelle zur Abbildung von Mensch-Maschine-Schnittstellen

2.4.3 Simulation von mechatronischen Systemen und Mensch-Maschine-Schnittstellen

Wie in Abschnitt 2.4.1 bereits dargelegt, werden Modelle häufig zur Simulation eingesetzt, um das Systemverhalten evaluieren zu können. Dies lässt sich grundsätzlich auch auf das Anwendungsfeld von mechatronischen Systemen und deren Mensch-Maschine-Schnittstellen übertragen. Die für das Verständnis dieser Arbeit notwendigen Grundlagen zur Simulation in beiden Bereichen werden in den nachfolgenden Abschnitten vorgestellt.

Simulation von mechatronischen Systemen

Analog zur Modellbildung lässt sich der Simulationseinsatz bei mechatronischen Systemen generell in disziplinspezifische und interdisziplinäre Ansätze unterteilen. Hinsichtlich der *disziplinspezifischen Simulation* lässt sich auch die weiterführende Unterteilung in Mechanik, Elektronik und Software erneut aufgreifen. In der mechanischen Entwicklung finden sich für die Simulation zwei Hauptanwendungsgebiete, wobei die zuvor eingeführten 3D-CAD-Modelle jeweils als Basis dienen. Einerseits werden in Festkörpersimulationen mittels der Finite-Elemente-Methode (FEM) detaillierte mechanische Analysen zur Ermittlung von bspw. Spannungen oder Festigkeiten einzelner Bauteile durchgeführt (VAJNA ET AL. 2009, S. 242-284). Andererseits wird in 3D-Kinematik- bzw. Mehrkörpersimulationen das Bewegungsverhalten von komplexen technischen Systemen analysiert und ausgewertet (VAJNA ET AL. 2009, S. 285-313). Demgegenüber haben sich in der Elektronik vor allem Schaltungssimulationen etabliert, wobei modellierte Schaltpläne die Basis bilden (ZAFIROV & ROUBANOV 2014, S. 154). Ziel dieses Simulationsansatzes ist generell die Evaluierung der Funktionalität und des Verhaltens von digitalen und analogen Schaltungen sowie die Ermittlung relevanter Kenngrößen, wie bspw. des Stromflusses (PILLAGE ET AL. 1999, S. 1-2). Im Bereich der Softwareentwicklung wird ebenfalls das Ziel verfolgt, das Verhalten eines Programms frühzeitig zu evaluieren und zu analysieren. Zu diesem Zweck können einfache ablauffähige Softwareprototypen eingesetzt werden, die entweder eine geforderte Funktionalität vereinfacht nachbilden oder nur Ausschnitte der gesamten Funktionalität beinhalten (ARNOWITZ ET AL. 2007, S. 3-4). Ferner existieren vielfältige Ansätze, um eine entwickelte Software in einer simulierten Umgebung zu beurteilen (DUSTIN ET AL. 2001, S. 4.; ROBMANN 2012B, S. 186 ff.). Diese Tests können in verschiedenen Ausbaustufen und Testumgebungen durchgeführt werden. Es handelt sich somit nicht um eine Simulation gemäß der eingangs vorgestellten Definition, da kein Modell, sondern die Software selbst erprobt wird.

Eine spezielle Variante, um eine entwickelte Software hinsichtlich ihrer Funktionalität zu testen, bildet die Virtuelle Inbetriebnahme (VIBN). Dabei wird die Software mit einem simulierten Maschinenmodell gekoppelt und kann dadurch bereits unter quasi realen Betriebsbedingungen getestet werden, ohne dass eine aufgebaute Maschine verfügbar sein muss (WÜNSCH 2007, S. 29 f.). Da für den Aufbau dieser Maschinenmodelle in der Regel 3D-CAD-Modelle als Grundlage verwendet werden, ist diese Variante bereits der *interdisziplinären Simulation* zuzuordnen. Generell finden sich in jüngster Vergangenheit vermehrt Ansätze, die eine Kopplung von bewährten disziplinspezifischen Simulationsansätzen vorsehen. Als Beispiel sei an dieser Stelle auf BREZINA ET AL. (2011) verwiesen, die eine Kopplung einer Mehrkörpersimulation mit einer regelungstechnischen Simulation zur Entwicklung mechatronischer Systeme vorschlagen. Weiterhin existieren mittlerweile erste Standards, wie das Functional Mock-up Interface (BLOCHWITZ ET AL. 2012), die den Datenaustausch zwischen vielfältigen Simulationsansätzen ermöglichen. Darüber hinaus haben sich auch umfassende Simulationsumgebungen etabliert, die Aspekte unterschiedlicher Entwicklungsdisziplinen abdecken und in der Regel auf mathematischen Beschreibungen eines zu realisierenden Systems beruhen. Als Beispiel sei das Simulationswerkzeug Dymola (DASSAULT SYSTEMES 2017) genannt, welches auf der Modellierungstechnik Modelica (FRITZSON 2010) basiert und eine Abbildung von verschiedenen Komponenten über Differentialgleichungen, Ports und Konnektoren ermöglicht. Abschließend sei erwähnt, dass die in Abschnitt 2.4.2 eingeführten Modellierungstechniken, wie die SysML, in der Regel nur zur Beschreibung und nicht zur Simulation von mechatronischen Systemen eingesetzt werden. Es gibt allerdings erste Ansätze, wie bspw. von BÜLTERMANN & BRANDSTÄTTER (2014), die eine Transformation von SysML-Modellen in ablauffähige Modellierungssprachen, wie z. B. Modelica, ermöglichen.

Simulation von Mensch-Maschine-Schnittstellen

Die im vorherigen Abschnitt vorgestellten Methoden zur Evaluation von Software lassen sich grundsätzlich auf das spezielle Themenfeld von Mensch-Maschine-Schnittstellen übertragen. Auch hier existieren verschiedene Ansätze, um die *Umgebung einer Mensch-Maschine-Schnittstelle*, in diesem Fall den Benutzer, zu simulieren (KAUSLER 2008). Dabei können die zuvor beschriebenen Aufgabenmodelle als Basis verwendet werden können (KAKLANIS ET AL. 2016). Mögliche Benutzermodelle werden bspw. von LEVIN ET AL. (2000) oder von PIETQUIN (2004) beschrieben. Dabei ist zu beachten, dass Benutzer lediglich auf abstrakter Ebene anhand ihrer Interaktionsmöglichkeiten und Aufgaben modelliert und simuliert werden. Dadurch unterscheiden sich diese Ansätze insbesondere von

umfänglichen dreidimensionalen Menschmodellen, wie bspw. dem RAMSIS-Ansatz von GEUB & BUBB (1994), die sich primär zur Evaluation der hier nicht relevanten Aspekte der Hardwareergonomie eignen.

Neben dem Test bereits implementierter Mensch-Maschine-Schnittstellen in einer simulierten Umgebung ist es für eine hohe Gebrauchstauglichkeit ebenfalls wichtig, in frühen Entwicklungsphasen auch *Modelle von Mensch-Maschine-Schnittstellen* zu simulieren und dabei Anwender einzubeziehen (ZÜHLKE 2012, S. 106 f.). Für diese Simulationen bilden die zuvor beschriebenen visuellen Modelle eine geeignete Basis. Bereits einfache händisch gezeichnete Wireframes können im Rahmen des „Paper Prototyping“ eingesetzt werden, um mit Benutzern eine zu entwickelnde Mensch-Maschine-Schnittstelle auf abstrakter Ebene zu diskutieren (SNYDER 2003). Wichtig ist zu erwähnen, dass Benutzer dabei nicht nur einen statischen Ausschnitt einer Mensch-Maschine-Schnittstelle evaluieren, sondern sich anhand verschiedener handgezeichneter Skizzen durch ein HMI navigieren und sich somit bereits einen ersten Eindruck hinsichtlich der Gebrauchstauglichkeit machen können (SNYDER 2003, S. 4). Ein darauf aufbauendes Konzept zur HMI-Simulation ist das „Wizard-of-Oz-Prototyping“ (PREIM & DACHSELT 2015, S. 126-127). Der Benutzer testet dabei eine Mensch-Maschine-Schnittstelle, von der nur die unmittelbar wahrnehmbaren Komponenten realisiert sind, wie bspw. das Frontend eines graphischen HMIs. Die eigentliche Funktionalität der Mensch-Maschine-Schnittstelle wird von einer zweiten Person, dem „Wizard“, simuliert, möglichst ohne dass dies dem eigentlichen Benutzer bekannt ist (PREIM & DACHSELT 2015, S. 126-127). Während sich das „Paper Prototyping“ vor allem zur Evaluation von graphischen HMIs anbietet, eignet sich der „Wizard-of-Oz“-Ansatz auch zur frühzeitigen Bewertung weiterer Interaktionskonzepte, insbesondere der Sprachsteuerung (KLEMMER ET AL. 2000). Darüber hinaus können auch die in Abschnitt 2.4.2 bereits vorgestellten ablauffähigen Prototypen eingesetzt werden, um eine Mensch-Maschine-Schnittstelle zu simulieren und dadurch hinsichtlich ihrer Gebrauchstauglichkeit zu evaluieren (BÄUMER ET AL. 1996). Wie bei der Softwareentwicklung im Allgemeinen können hierfür vereinfachte Varianten der zu entwickelnden Software in entsprechenden HMI-Entwicklungsumgebungen umgesetzt werden. Aufgrund der hohen Bedeutung des Prototyping im Kontext der Entwicklung von Mensch-Maschine-Schnittstellen konnten sich zudem spezielle Werkzeuge etablieren, die einen schnellen und einfachen Aufbau von visuellen HMI-Modellen ermöglichen (PREIM & DACHSELT 2015, S. 141-145).

3 Stand der Wissenschaft und Technik

Aufbauend auf den zuvor dargestellten Grundlagen werden im folgenden Kapitel die Entwicklungsprozesse im Maschinen- und Anlagenbau untersucht. Dazu werden in den Abschnitten 3.1 und 3.2 die zentralen Charakteristika bei der Entwicklung von mechatronischen Systemen im Allgemeinen sowie deren Mensch-Maschine-Schnittstellen im Speziellen dargelegt. Auf dieser Basis werden in Abschnitt 3.3 aktuelle Entwicklungsansätze für mechatronische Systeme vorgestellt. Dabei erfolgt auch eine erste Analyse, inwiefern diese Ansätze die Konzeption und Umsetzung von HMIs unterstützen. Weiterhin werden in Abschnitt 3.4 spezielle Entwicklungsansätze betrachtet, die sich ausschließlich Mensch-Maschine-Schnittstellen widmen. Da die HMI-Entwicklung im Maschinen- und Anlagenbau bisher nur bedingt betrachtet wird (vgl. Kapitel 1), adressieren die vorgestellten Ansätze nur zum Teil industrielle Anwendungen. Der Großteil der Konzepte ist hingegen auf Mensch-Maschine-Schnittstellen von Computeranwendungen ausgerichtet. Jedoch können Teilaspekte dieser Ansätze auch im Rahmen dieser Arbeit genutzt werden, um sie in einen ganzheitlichen mechatronischen Entwicklungsansatz zu integrieren. Dieser Gesichtspunkt wird in Abschnitt 3.5 aufgegriffen, um den resultierenden Handlungsbedarf für die vorliegende Arbeit aufzuzeigen.

3.1 Überblick über die Entwicklung mechatronischer Systeme

Wie aus den vorangegangenen Kapiteln ersichtlich wird, umfasst die Entwicklung von Maschinen und Anlagen ein breites Spektrum an notwendigen Aktivitäten, um von einer initialen Produktidee zu einem voll funktionsfähigen mechatronischen System zu gelangen (ISERMANN 2007, S. 17-19). Dies lässt sich insbesondere durch die gestiegene Bedeutung von Elektronik und Software in den letzten Jahren begründen. Neben zahlreichen Entwicklungsaktivitäten innerhalb dieser Gewerke macht der interdisziplinäre Charakter von heutigen Maschinen und Anlagen ein ganzheitliches Denken in Systemen erforderlich, was insbesondere in der frühen Entwicklungsphase zu weiterführenden Aktivitäten im Kontext des Systementwurfs führt (VDI 2206, S. 24). Einen ganzheitlichen Überblick über die einzelnen Aktivitäten, die bei einer Maschinen- oder Anlagenentwicklung generell durchzuführen sind, liefert das mechatronische Referenzmodell, welches im Forschungsprojekt MEPRONA von der Technischen Universität München gemeinsam mit zahlreichen Unternehmen des Maschinen- und Anlagenbaus erarbeitet wurde (VDMA 2015). Das Modell definiert neun Prozessgebiete, die jeweils Cluster für unterlagerte Aktivitäten und konkrete Aktionen darstellen (vgl. Abbildung 3-1). Neben den

wertschöpfenden Prozessgebieten, wie bspw. dem Systementwurf, werden auch unterstützende Entwicklungsaufgaben, wie die Projektplanung, in dem Modell erfasst. Im Prozessgebiet „Systemrealisierung“ unterteilt das Modell die einzelnen Aktivitäten anhand der drei typischen Disziplinen Mechanik, Elektronik und Software, berücksichtigt aber auch die nötigen Tätigkeiten zur interdisziplinären Abstimmung, was durch die Aktivität „System realisieren“ deutlich wird.

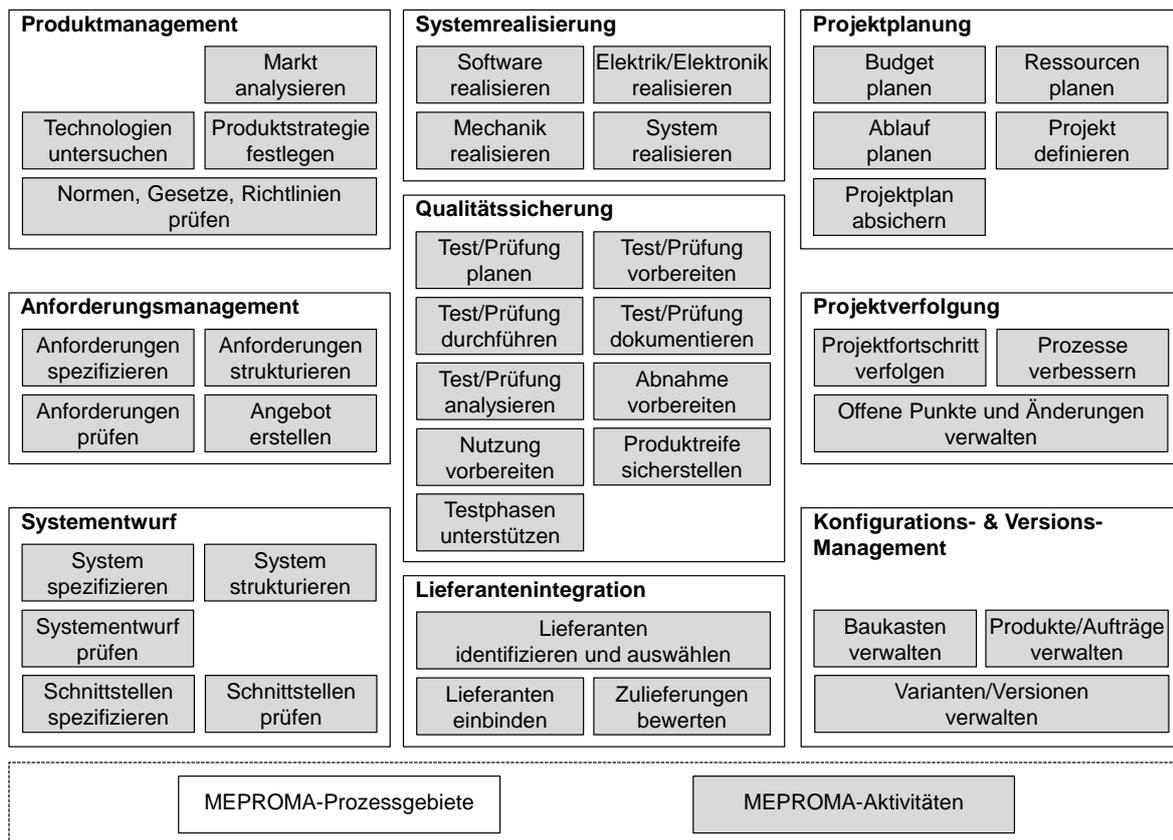


Abbildung 3-1: Prozessgebiete und Aktivitäten im Referenzmodell aus dem Forschungsprojekt MEPRONA (in Anlehnung an VDMA 2015, S. 15)

Aus der Vielzahl und Diversität der einzelnen Prozessgebiete und Aktivitäten dieses Referenzmodells lässt sich folgern, dass an der Entwicklung von Maschinen und Anlagen in der Regel mehrere Personengruppen beteiligt sind. Der VDMA-Leitfaden „Projektmanagement für interdisziplinäre Systementwicklungen“ definiert insgesamt 29 Beteiligte (sog. Stakeholder), die in projektspezifische Rollen, wie den Qualitätsmanager oder die Projektleitung, und projektunspezifische Rollen, wie den Vertrieb oder die Geschäftsleitung, unterteilt werden können (VDMA 2005, S. 139-141). Dieses Spektrum wird von MAUDERER (2010, S. 42 ff.) auf sechs zentrale Rollen reduziert, die in einem mechatronischen Entwicklungsprozess von besonderer Relevanz sind. Unter diesen ist zunächst der *Kunde* zu nennen, der im

Wesentlichen Anforderungen an das zu entwickelnde System definiert sowie das fertiggestellte System prüft und abnimmt. Die Aufgaben des *Anlagendesigners*⁵ bestehen demgegenüber in der Interpretation der Kundenanforderungen und der darauf aufbauenden Ausarbeitung einer Systemspezifikation. Die Entwickler (*Mechanik-, Elektrik- sowie maschinennahe und -ferne Softwareentwickler*) sind schließlich für die Konkretisierung und Umsetzung der Systemspezifikation hinsichtlich ihrer jeweiligen Teilbereiche verantwortlich.

Die Anzahl notwendiger Aktivitäten und involvierter Personen macht es weiterhin erforderlich, die Entwicklungsprozesse im Maschinen- und Anlagenbau nach einem systematischen Vorgehen zu durchlaufen (GAUSEMEIER ET AL. 2000, S. 95). Die noch immer vorhandene Fokussierung der Mechanik (CADET ET AL. 2017, S. 46) sowie ein generelles Denken in Disziplinen (HOLM ET AL. 2013, S. 263) führen dazu, dass die Entwicklung von Maschinen und Anlagen aktuell meist in einem sequenziellen Prozess durchlaufen wird (vgl. Kapitel 1). Innerhalb der einzelnen Disziplinen haben sich eigene Vorgehensmodelle etabliert, wie bspw. die VDI 2221 für die Mechanik oder die VDI/VDE 2422 für die Elektronik. Um aber den Anforderungen, wie bspw. hinsichtlich einer steigenden Kundenindividualität oder kürzer werdenden Entwicklungszeiten, gerecht zu werden, bedarf es zukünftig integrativer Herangehensweisen, welche speziell die Interdisziplinarität in der Entwicklung fokussieren und ein simultanes Engineering der beteiligten Gewerke ermöglichen (SPATH & DANGELMAIER 2016, S. 3). Auch hierfür wurden bereits zahlreiche Vorgehensmodelle erarbeitet. Aufgrund der besonderen Bedeutung für die vorliegende Arbeit werden diese Modelle detailliert in Abschnitt 3.3 vorgestellt.

Einen letzten wichtigen Aspekt für die Entwicklung mechatronischer Systeme, der für die vorliegende Arbeit von großer Bedeutung ist, bildet die Modularisierung. Die Aufteilung von Maschinen und Anlagen in voneinander gekapselte, losgelöste Einheiten dient im Wesentlichen der Vereinfachung von Systemen und der Wiederverwendung von Teillösungen (GÖPFERT & STEINBRECHER 2000, S. 6). Dadurch ist es möglich, mechatronische Systeme aus einem Baukasten bereits erarbeiteter Module aufzubauen und so den Aufwand für Neuentwicklungen gering zu halten sowie Entwicklungszeiten und -kosten zu senken (EITELWEIN ET AL. 2012, S. 80). Auch hinsichtlich der Modularisierung erkennt man im Maschinen- und Anlagenbau aktuell noch die starke Fokussierung der Mechanik, die dazu führt, dass Module hauptsächlich an der mechanischen Struktur des Systems ausgerichtet werden (FELDMANN ET AL. 2012, S. 1692 f.). Um allerdings der gestiegenen Bedeutung von

⁵ Oft auch als „Systems Engineer“ bezeichnet

Elektronik und Software gerecht zu werden, wird aktuell an verschiedenen Konzepten im Bereich der mechatronischen Modularisierung, wie bspw. REINHART ET AL. (2009) oder POSSEL-DÖLKEN (2010), geforscht. Gemein ist den einzelnen Forschungsrichtungen die grundlegende Motivation, eine Maschine oder Anlage so einzuteilen, dass sich wiederverwendbare Module finden lassen, die jeweils eine Mechanik, Elektronik und Software beinhalten und dadurch direkt zur Realisierung einer bestimmten Funktion eingesetzt werden können (LÜDER ET AL. 2010, S. 2). Dieser Grundgedanke findet sich mittlerweile auch in einzelnen mechatronischen Entwicklungsansätzen wieder, die in Abschnitt 3.3 näher betrachtet werden.

3.2 Überblick über die Entwicklung von Mensch-Maschine-Schnittstellen

In Analogie zum vorangegangenen Abschnitt werden im Folgenden zunächst zentrale Entwicklungsaktivitäten für Mensch-Maschine-Schnittstellen definiert. Zu deren Einordnung kann ebenfalls das mechatronische Referenzmodell des VDMA (2015) verwendet werden. Da die HMI-Entwicklung hier allerdings nur rudimentär abgebildet ist, wurde das Modell von BERG ET AL. (2017) um die wichtigsten Aufgaben bezüglich der Konzeption und Realisierung von Mensch-Maschine-Schnittstellen erweitert. Insgesamt wurden hierbei 68 Aktionen identifiziert, die bei einer HMI-Entwicklung im Maschinen- und Anlagenbau umzusetzen sind. Die Prozessgebiete und Aktivitäten des Referenzmodells konnten dabei zur Gruppierung der einzelnen Aktionen mit geringen Anpassungen genutzt werden. Die Gesamtheit der von BERG ET AL. (2017) angeführten Aktionen macht deutlich, dass die Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau im Wesentlichen von der Konzeption und Implementierung der jeweiligen Software geprägt ist, während hinsichtlich der Hardware hauptsächlich geeignete Technologien auszuwählen und zu beschaffen sind. Abbildung 3-2 gibt einen exemplarischen Überblick über diese Aktionen innerhalb der wichtigsten Prozessgebiete des mechatronischen Referenzmodells. Wichtig ist zu erwähnen, dass aufgrund der noch geringen Berücksichtigung von Mensch-Maschine-Schnittstellen in den Entwicklungsprozessen des Maschinen- und Anlagenbaus viele der definierten Aktionen (insbesondere im Anforderungsmanagement und im Systementwurf) oft nicht durchgeführt werden. In der Regel wird die HMI-Entwicklung auf wenige Aktionen im Bereich der System-Realisierung beschränkt, was zu unausgereiften und nicht gebrauchstauglichen Mensch-Maschine-Schnittstellen führt (REINHART ET AL. 2010, S. 36).

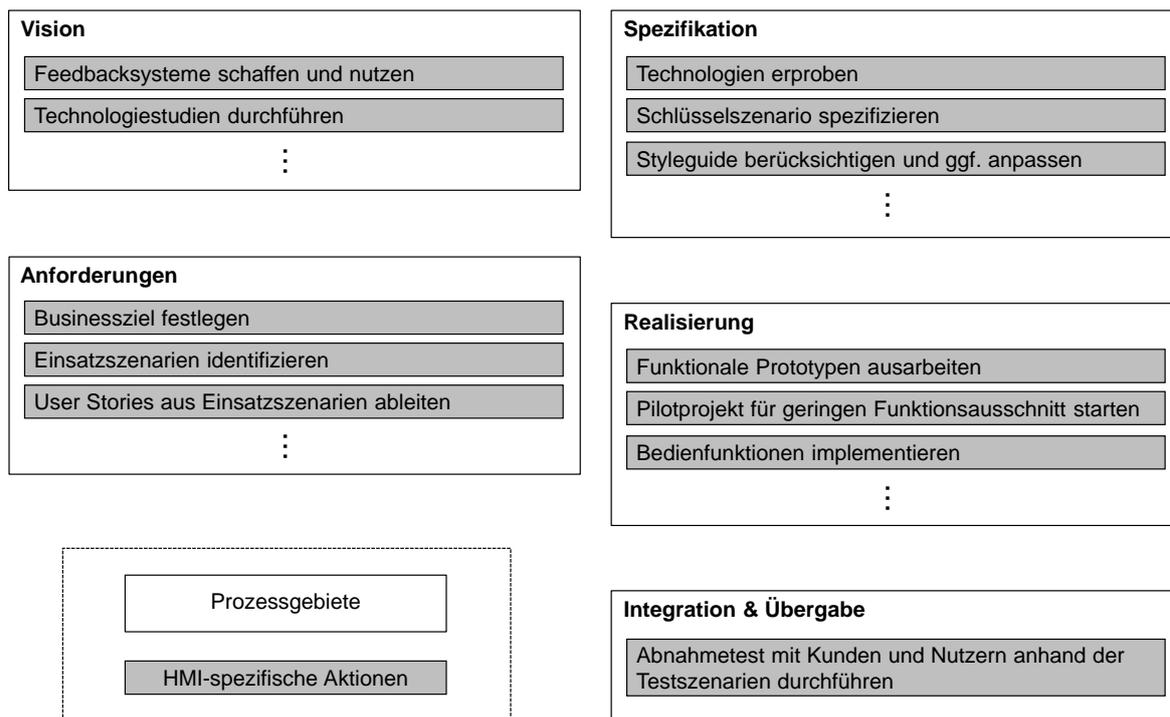


Abbildung 3-2: Exemplarische HMI-spezifische Aktionen im Kontext der MEPROMA-Prozessgebiete (in Anlehnung an BERG ET AL. 2017, S. 47)

Die geringe Bedeutung von HMIs im Maschinen- und Anlagenbau führt weiterhin dazu, dass die eben beschriebenen Aktionen meist ausschließlich von der Rolle des Softwareentwicklers durchgeführt werden (vgl. Abschnitt 3.1). Die steigende Bedeutung des Designs und der Usability von Mensch-Maschine-Schnittstellen werden es zukünftig allerdings erforderlich machen, die einzelnen Aufgaben durch ein Team von Experten unterschiedlicher Fachrichtungen zu realisieren (ZÜHLKE 2012, S. 36; BENGLER 2011, S. 77). ZHENG ET AL. (2011) schlagen dafür insgesamt vier Rollen vor, die an der Konzeption und Implementierung eines HMIs beteiligt sind. Unter diesen ist zunächst der *Usability-Experte* zu nennen, der generell für die Erreichung einer hohen Gebrauchstauglichkeit verantwortlich ist und dazu u. a. den Nutzungskontext analysiert sowie die zentralen Anwendungsfälle identifiziert. Auf dieser Basis ist der *Designer* hauptsächlich für die Darstellung der Mensch-Maschine-Schnittstelle verantwortlich, um u. a. Icons oder das generelle Farbschema des HMIs festzulegen und zu gestalten. Die Hauptaufgabe des *Entwicklers* liegt demgegenüber in der Implementierung der geforderten Funktionalität und des Verhaltens der Mensch-Maschine-Schnittstelle. Der *Integrator* ist schließlich dafür verantwortlich, die Aktivitäten aller Beteiligten zu koordinieren und zu synchronisieren sowie grundsätzlich die Gesamtarchitektur des HMIs festzulegen.

Wie beim mechatronischen Entwicklungsprozess im Allgemeinen ist es auch für die HMI-Entwicklung im Speziellen notwendig, die relevanten Aktionen nach einem systematischen Vorgehen durchzuführen. Im Kontext von Computeranwendungen haben sich bereits zahlreiche Herangehensweisen etabliert, welche eine Umsetzung gebrauchstauglicher Mensch-Maschine-Schnittstellen ermöglichen (RICHTER & FLÜCKIGER 2013, S. 16 f.). Diese Ansätze wurden inzwischen auch für die Domäne des Maschinen- und Anlagenbaus adaptiert und optimiert. Die für die vorliegende Arbeit relevanten Vorarbeiten in diesem Bereich werden explizit in Abschnitt 3.4 thematisiert. Dabei ist allerdings zu erwähnen, dass sich bisherige Ansätze zumeist kaum mit einer ganzheitlichen mechatronischen Herangehensweise in Einklang bringen lassen, was gleichzeitig eine zentrale Motivation für die vorliegende Arbeit darstellt.

Analog zu Abschnitt 3.1 bildet die Modularisierung den letzten wichtigen Aspekt bei der Entwicklung von Mensch-Maschine-Schnittstellen, der für die vorliegende Arbeit von großer Bedeutung ist. Auch für HMIs existieren Bestrebungen, bereits erarbeitete Lösungen in verschiedenen Entwicklungsvorhaben wiederzuverwenden (HÄRTER 2016). So werden bspw. unternehmensspezifische Styleguides erstellt, die übergreifend für alle Mensch-Maschine-Schnittstellen in einem Maschinenpark eingesetzt werden. Oftmals werden aber auch ganze Ausschnitte eines HMIs, wie einzelne Bedien- und Anzeigeelemente oder das Navigationskonzept, systemübergreifend realisiert. Innerhalb eines konkreten Entwicklungsvorhabens können diese unmittelbar verwendet werden oder müssen lediglich für den Anwendungsfall angepasst bzw. parametrisiert werden.

3.3 Entwicklungsansätze für mechatronische Systeme

Basierend auf dem Überblick über die Entwicklung von mechatronischen Systemen und Mensch-Maschine-Schnittstellen werden im Folgenden aktuelle Lösungsansätze vorgestellt, die sich mit systematischen Vorgehensweisen zur Realisierung mechatronischer Systeme beschäftigen. Dabei werden zunächst in Abschnitt 3.3.1 generelle Ansätze kurz zusammengefasst, die ein simultanes Engineering sowie eine interdisziplinäre Herangehensweise unterstützen. Darauf aufbauend werden in Abschnitt 3.3.2 spezielle Ansätze im Detail betrachtet, die hierfür den Einsatz interdisziplinärer Systemmodelle vorschlagen. Schließlich wird in Abschnitt 3.3.3 analysiert, inwiefern die vorgestellten Ansätze bereits die Entwicklung von Mensch-Maschine-Schnittstellen unterstützen.

3.3.1 Generelle Vorgehensmodelle und Entwicklungsmethoden

Um Entwicklungsprozesse in der Mechatronik nach einer strukturierten Vorgehensweise durchführen zu können, wurden in den vergangenen Jahren unterschiedliche Vorgehensmodelle und Entwicklungsmethoden erarbeitet. Der bekannteste Vertreter ist das V-Modell der Mechatronik gemäß der VDI 2206. Dieses Modell stellt eine Weiterentwicklung des ursprünglichen V-Modells für die Entwicklung von Softwaresystemen dar und überträgt den dort definierten Ablauf auf mechatronische Problemstellungen. Dabei bilden Anforderungen den Ausgangspunkt, die in einem Entwicklungsauftrag beschrieben sind. In der ersten Phase, dem Systementwurf, wird auf deren Basis ein domänenübergreifendes Lösungskonzept erarbeitet, das die physikalischen und logischen Wirkweisen des zukünftigen Produkts beschreibt. Anschließend wird dieses Konzept im Rahmen des domänenspezifischen Entwurfs unter Verwendung jeweils eigener Vorgehensweisen in den Disziplinen konkretisiert. In der dritten Phase, der Systemintegration, werden die zuvor erarbeiteten Ergebnisse schließlich zu einem Gesamtsystem bzw. fertigen Produkt zusammengeführt. Hierbei wird mittels Tests fortlaufend überprüft, ob das entwickelte System dem Lösungskonzept entspricht. Flankiert werden die Arbeiten innerhalb dieser Phasen durch eine kontinuierliche Untersuchung der Systemeigenschaften mit Hilfe von Modellen und rechnergestützten Werkzeugen zur Simulation. Das eben eingeführte V-Modell wurde mittlerweile bereits mehrfach als Grundlage für weitere Vorgehensmodelle verwendet. BENDER (2005) konkretisiert das Modell bspw. durch die Einführung von drei vertikalen Ebenen (System-, Subsystem- und Komponentenebene). Eine darauf aufbauende Variante des V-Modells zur Entwicklung mechatronischer Systeme wird schließlich von ANDERL ET AL. (2012) beschrieben. Das V-Modell wird hier im Bereich des domänenspezifischen Entwurfs durch einen zusätzlichen Integrationsschritt erweitert, da die weitgehend unabhängige Entwicklung in den einzelnen Disziplinen oftmals zu Problemen in der Integrationsphase führt (NATTERMANN & ANDERL 2013, S. 939).

Eine Alternative zu den dargestellten Varianten des V-Modells bildet das Quality-Gate-Modell (VDMA 2005). In diesem Modell wird die Entwicklung von mechatronischen Systemen durch sog. Quality Gates in sechs Projektphasen unterteilt. Die einzelnen Gates stellen Messpunkte dar, an denen am Ende jeder Phase definierte Zwischenergebnisse disziplinübergreifend bewertet werden. Dies ermöglicht eine Synchronisation der drei zentralen Entwicklungsstränge, die innerhalb der einzelnen Phasen durchaus gewollt auseinanderdriften dürfen, um kreative Freiräume zu schaffen. In Ergänzung zu der übergeordneten Vorgehensweise werden im Quality-

Gate-Modell auch konkrete Prozessschritte und Entwicklungsaktivitäten definiert, die innerhalb der einzelnen Phasen typischerweise zu durchlaufen sind. Dabei wird insbesondere der interdisziplinäre Systementwurf betrachtet, um bereits frühzeitig ein gemeinsames Verständnis zwischen allen Projektbeteiligten und dem Kunden zu erzielen. Hierfür wird der Einsatz einer entsprechenden Spezifikationstechnik vorgeschlagen, die sich aus fünf zentralen Sichten, wie bspw. der Struktur- und Ablaufsicht, zusammensetzt. Das Ziel beim Einsatz der Spezifikationstechnik ist die Dokumentation aller relevanten Anforderungen, Rahmenbedingungen und Systembestandteile in einem zentralen Dokument, welches den Entwicklungsprozess bis zur Abnahme des finalen Produkts begleitet.

Ein weiterer Leitfaden, um mechatronische Systeme nach einem systematischen Vorgehen zu entwickeln, wurde von ISERMANN (2008) erarbeitet. Seine Methodik definiert 13 zentrale Entwurfsschritte, die anhand der jeweils durchzuführenden Aufgaben entsprechend präzisiert werden. Grundsätzlich folgt die von ISERMANN (2008) entwickelte Herangehensweise dem Ablauf des V-Modells. Basierend auf einer Anforderungsklä rung und -spezifikation wird im Systementwurf eine Unterteilung in entsprechende Module und Komponenten durchgeführt. In den darauf aufbauenden Schritten erfolgt die disziplinspezifische Umsetzung dieser Komponenten, die schließlich sukzessive zu einem Gesamtsystem zusammengeführt und dabei getestet werden. Die Unterteilung des Prozesses in 13 Entwurfsschritte ermöglicht allerdings eine differenziertere Betrachtung eines mechatronischen Entwicklungsprozesses, in der unter anderem explizit auf die Rolle von Prototypen eingegangen wird. Weiterhin definiert ISERMANN (2008) innerhalb der Entwurfsschritte konkrete Entwicklungsaufgaben, resultierende Zwischenergebnisse sowie einzusetzende Modelle und Werkzeuge, wobei auch die Konzeption und Realisierung von Mensch-Maschine-Schnittstellen thematisiert wird (vgl. Abschnitt 3.3.3).

Eine grundsätzlich andere Herangehensweise zur Entwicklung mechatronischer Systeme wird weiterhin von KLEIN (2016) eingeführt, der eine Integration von Aspekten der agilen Softwareentwicklung (COHEN ET AL. 2003) in mechatronische Entwicklungsprozesse ermöglicht. Dabei ist die agile Entwicklung stark von einem iterativen und inkrementellen Vorgehen geprägt, in dem die schnelle Implementierung funktionierender Software und deren kontinuierliche Abstimmung mit dem Kunden im Vordergrund stehen (BECK ET AL. 2001). In seinem Ansatz verfolgt KLEIN (2016) das Ziel, diesen Gedanken auf die Entwicklung mechatronischer Systeme im Maschinen- und Anlagenbau zu übertragen. Dabei wird insbesondere eine Herangehensweise beschrieben, um agile Techniken in den mechatronischen Entwicklungsprozess zu integrieren. Dafür nutzt KLEIN (2016) das mechatronische

Referenzmodell des VDMA (vgl. Abschnitt 3.1) und stellt den einzelnen Aktionen agile Techniken in einer Domain Mapping Matrix gegenüber. Dies ermöglicht die Aussage, welche Aktionen eines mechatronischen Entwicklungsprozesses sinnvoll durch agile Techniken unterstützt werden können, wobei die Techniken von KLEIN (2016) in einzelne Aktivitäten, Artefakte, Rollen und Visualisierungsmethoden unterteilt werden. Bei insgesamt 28 agilen Techniken wurden 839 Kombinationen zu den Aktionen in einem mechatronischen Entwicklungsprozess herausgearbeitet. Dabei zeigt sich, dass speziell das Anforderungsmanagement, die Projektverfolgung und die Qualitätssicherung durch agile Techniken unterstützt werden können.

Neben den bisher vorgestellten Herangehensweisen existiert eine Vielzahl weiterer Ansätze, die den mechatronischen Entwicklungsprozess unterstützen können. Von GAUSEMEIER & FELDMANN (2006) wurde bspw. ein Vorgehensmodell erarbeitet, das die integrative Entwicklung von Produkt und Fertigungssystem für integrierte elektromechanische Baugruppen ermöglicht. Auch LÜCKEL ET AL. (2000) erarbeiteten ein Vorgehensmodell, das die Komplexitätsproblematik mechatronischer Systeme adressiert und dafür den Einsatz rechnergestützter Entwurfsverfahren vorsieht. Weitere Modelle für die Entwicklung mechatronischer Systeme wurden u. a. von CHAN & LEUNG (1996) oder von SUH (2001) erarbeitet. Darüber hinaus ist für mechatronische Systeme generell das Themenfeld des Systems Engineerings relevant, das sich mit der interdisziplinären Entwicklung komplexer technischer Systeme beschäftigt. Die wichtigsten Methoden und Techniken in diesem Bereich werden von HABERFELLNER ET AL. (2015) oder INCOSE (2015) beschrieben. Schließlich können auch allgemeine Ansätze der Produktentwicklung bei mechatronischen Systemen zum Einsatz kommen, wie sie z. B. von PONN & LINDEMANN (2011) oder von EHRENSPIEL & MEERKAMM (2017) dargestellt werden.

3.3.2 Spezielle Ansätze zur modellbasierten Entwicklung

Aufgrund der in Kapitel 1 dargestellten Potenziale bezüglich des Einsatzes von Modellen in der Entwicklung mechatronischer Systeme wurden in den vergangenen Jahren verschiedene Ansätze in diesem Bereich erarbeitet. Dabei bilden die zuvor dargestellten Vorgehensmodelle oft die Grundlage. Im Folgenden werden die relevanten Vorarbeiten im Detail vorgestellt, bevor schließlich ein Ausblick auf weiterführende Ansätze im Kontext der modellbasierten Systementwicklung erfolgt. Im Einzelnen werden die Ansätze von EIGNER (2013), GAUSEMEIER ET AL. (2014), VDMA (2010), HENSEL (2011) sowie HACKENBERG ET AL. (2015) präsentiert.

Modellbasierte virtuelle Produktentwicklung nach EIGNER

Das Vorgehensmodell von EIGNER (2013) adressiert die Komplexität im Systementwurf mechatronischer Systeme, die durch das Zusammenspiel verschiedener Disziplinen entsteht. Zu diesem Zweck erweitert EIGNER (2013) das V-Modell der VDI 2206 mit Ansätzen der modellbasierten Systementwicklung (vgl. Abbildung 3-3). Dabei lassen sich grundsätzlich drei Ebenen der Modellbildung unterteilen. In der *interdisziplinären Modellbildung und Spezifikation* wird ein zu entwickelndes System anhand qualitativer Modelle beschrieben, die Informationen hinsichtlich Anforderungs-, Funktions-, Verhaltens- und logischen Systemstrukturen beinhalten. Dazu eignen sich graphische Beschreibungssprachen wie die SysML. Auf der Ebene der *interdisziplinären Modellbildung und ersten Simulation* werden bereits quantitative, simulierbare Modelle verwendet, die nach wie vor mehrere Disziplinen einbeziehen. Dabei können multi-physikalische Simulationen wie Modelica zum Einsatz kommen. Auf der Ebene der *disziplinspezifischen Modellbildung und Simulation* werden schließlich CAD- oder CASE-Modelle verwendet, die mit disziplinspezifischen Werkzeugen aufgebaut werden.

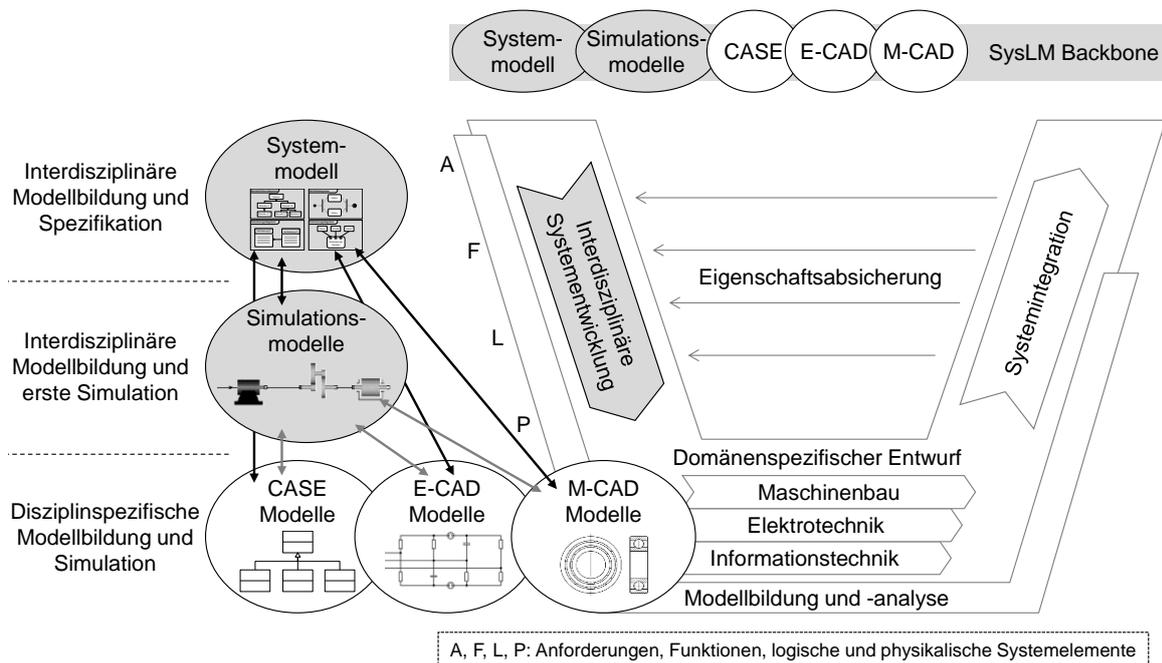


Abbildung 3-3: Erweitertes V-Modell für die modellbasierte Systementwicklung (in Anlehnung an EIGNER 2013, S. 102)

Die einzelnen Modelle werden im linken Ast des V-Modells zur Abbildung unterschiedlicher aufeinander aufbauender Aspekte eines mechatronischen Systems genutzt. Den Anfang bildet dabei eine Anforderungsanalyse, in der Kundenanfor-

derungen in logisch konsistente, technische Anforderungen übersetzt werden. Auf dieser Basis wird im interdisziplinären Systementwurf eine funktionale Lösung entwickelt, mit der alle beteiligten Disziplinen einverstanden sind. Dabei steht die Modellierung einer Funktions- und Verhaltensbeschreibung im Vordergrund. In einem weiteren Schritt wird das erarbeitete Lösungskonzept durch logische Komponenten konkretisiert, die funktionale Elemente und Verhalten realisieren. Durch den Einsatz disziplinspezifischer Modelle werden diese Komponenten schließlich in die tatsächlichen physikalischen Elemente des Systems überführt.

Neben dem Modelleinsatz im Rahmen der interdisziplinären Systementwicklung fokussiert der Ansatz von EIGNER (2013) auch die Verwaltung der während der Entwicklung entstehenden Informationen. Dafür wird ein eigenes Datenmodell vorgeschlagen, welches grundsätzlich auf gängigen PLM-Lösungen basiert, diese allerdings im Bereich der interdisziplinären Systemmodellierung erweitert. Da das entwickelte Datenmodell neben disziplinspezifischen Modellen auch funktionale und verhaltensorientierte Beschreibungsmodelle verwalten kann und somit den gesamten Lebenszyklus eines Systems begleitet, wird in diesem Kontext der Begriff des Systems Lifecycle Management (SysLM) verwendet.

Domänenübergreifende Entwicklungsmethodik nach GAUSEMEIER ET AL.

Ein vergleichbarer Ansatz für die modellbasierte Entwicklung mechatronischer Systeme wird von GAUSEMEIER ET AL. (2014) vorgeschlagen. Das hier dargestellte Vorgehensmodell unterteilt den Entwicklungsprozess auf oberster Ebene in eine domänenübergreifende Konzeption sowie eine domänenspezifische Konkretisierung (vgl. Abbildung 3-4). In der ersten Phase wird dabei zunächst eine Prinziplösung auf Basis der Modellierungstechnik CONSENS (vgl. Abschnitt 2.4.2) erarbeitet. Diese Lösung wird in den einzelnen Domänen daraufhin entsprechend konkretisiert, um eine vollständige Beschreibung des mechatronischen Systems zu erhalten.

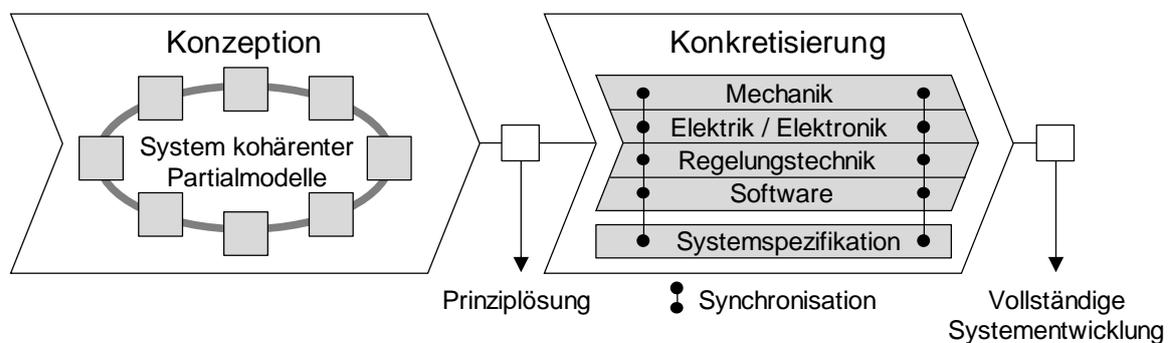


Abbildung 3-4: Konzeption und Konkretisierung mechatronischer Systeme (in Anlehnung an GAUSEMEIER ET AL. 2014, S. 67)

Die Phase der Konzeption ist dabei in vier Schritte untergliedert, in denen unter Verwendung eines Systems kohärenter Partialmodelle eine Prinziplösung des mechatronischen Systems gebildet wird. Der erste Schritt dient dabei zur Planung und Klärung der Aufgabenstellung. In diesem Rahmen erfolgt eine Analyse der Entwicklungsaufgaben, um mögliche Anwendungsszenarien und Anforderungen zu ermitteln. Im zweiten Schritt erfolgt eine erste Konzeption auf Systemebene, wobei zunächst die Hauptfunktionen und darauf aufbauende Funktionsstrukturen bestimmt werden. Anschließend werden für jedes Anwendungsszenario unterschiedliche Lösungsvarianten modelliert, bevor die besten ausgewählt und zu einer Prinziplösung auf Systemebene konsolidiert werden. Für einen tieferen Einblick wird das konzipierte System im dritten Schritt in Module unterteilt. Für jedes definierte Modul werden die zuvor beschriebenen Schritte durchlaufen, um zu Prinziplösungen auf Modulebene zu gelangen. Im letzten Schritt werden diese Lösungen zu einer detaillierten Prinziplösung für das Gesamtsystem integriert. Durch eine Evaluation des Konzepts werden auftretende Widersprüche identifiziert und eliminiert.

Die im Rahmen der Konzeption entwickelte Prinziplösung bildet die Basis für die Konkretisierung, in der die einzelnen Domänen⁶ mittels eigener Vorgehensweisen ein vollständiges mechatronisches System entwickeln. Über den gesamten Verlauf dieser Phase ist es allerdings entscheidend, dass die domänenspezifischen Modelle konsistent zur Systemspezifikation bleiben. Der erste Schritt ist dabei die automatische Generierung initialer domänenspezifischer Modelle aus der Systemspezifikation durch Modelltransformationen. Da aber während der Konkretisierung auch Änderungen auftreten können, die relevant für andere Domänen sind, ist es notwendig, diese zu erkennen und die involvierten Domänen zu synchronisieren. Dies erfolgt anhand der Systemspezifikation, die bei jeder relevanten Änderung zu aktualisieren ist. Da auch die domänenspezifischen Modelle zum aktuellen Stand der Systemspezifikation konsistent sein müssen, kann über den Entwicklungsverlauf eine Synchronisation der beteiligten Domänen sichergestellt werden.

AQUIMO-Entwicklungsmethode

Im vom BMBF geförderten Forschungsprojekt AQUIMO wurde ebenfalls das Ziel verfolgt, den Entwicklungsprozess mechatronischer Systeme durch den Einsatz eines interdisziplinären Modellierungswerkzeugs zu unterstützen (VDMA 2010). Die Basis für den Ansatz bildeten die Erkenntnisse aus dem Forschungsprojekt

⁶ Im Unterschied zu den bisher dargestellten Ansätzen fassen GAUSEMEIER ET AL. (2014) die Regelungstechnik nicht als Teil der Software, sondern als eigene Disziplin auf.

„Föderal“ (LITTO ET AL. 2004), in dem eine Informationsarchitektur aufgebaut wurde, die bestehende Modelle (z. B. CAD) in ein übergeordnetes Entwurfssystem integriert. Dieser Ansatz wurde im Rahmen von AQUIMO aufgegriffen, um eine Entwicklungsmethode und ein Modellierungswerkzeug aufzubauen, mit denen es möglich ist, durchgängige und reproduzierbare Entwicklungsprozesse zu erreichen. Die Inhalte des Projekts „Föderal“ wurden um den Aspekt des ganzheitlichen mechatronischen Ansatzes und die Modellierung derartiger Komponenten erweitert. Mit Hilfe eines einheitlichen Modells sollen alle an einem Entwicklungsprozess Beteiligten die Funktionen entwerfen und das Zusammenspiel von Mechanik, Elektrik und Software diskutieren. Die zugehörige Entwicklungsmethode sieht dazu eine Vorgehensweise vor, in der ausgehend von einem initialen Aufbau eines formalen Modells funktionaler Strukturen zwischen Phasen des interdisziplinären und des disziplinspezifischen Engineerings alterniert wird (vgl. Abbildung 3-8).

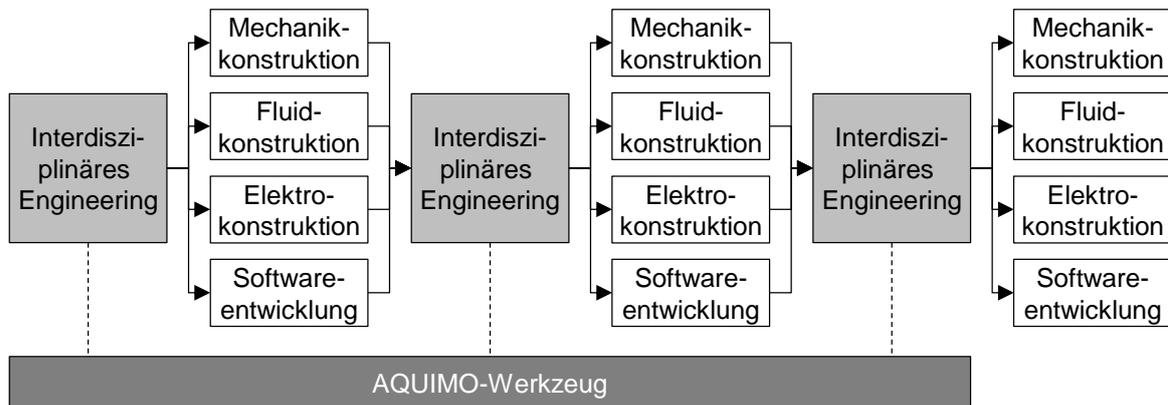


Abbildung 3-5: Modellbasierter Entwicklungsprozess mit dem AQUIMO-Werkzeug (in Anlehnung an VDMA 2010)

Dabei wird in den Phasen des disziplinspezifischen Engineerings zwar die Fluidtechnik als eigene Disziplin berücksichtigt, von der modellbasierten Herangehensweise ist der Ansatz aber mit den vorigen Beispielen vergleichbar. Die entwickelte Modellierungstechnik ermöglicht die formale Spezifikation von geometrischen Systembestandteilen, funktionalen Strukturen bis zur Sensor-Aktor-Ebene sowie den notwendigen Abläufen und Verhaltensweisen einzelner Komponenten. Der formale Charakter der Modellierungstechnik ermöglicht die automatische Generierung eines Simulationsmodells aus dem interdisziplinären Entwurf, was eine frühe simulative Bewertung verschiedener Lösungsalternativen gestattet. Ein weiterer zentraler Aspekt des entwickelten Ansatzes ist die Umsetzung eines Baukastenprinzips, mit dessen Hilfe Maschinen und Anlagen disziplinübergreifend auf einer Bibliotheksbasis zusammengestellt und konfiguriert werden können.

Modellbasierter Entwicklungsprozess nach HENSEL

Im Unterschied zu den bisher vorgestellten Konzepten fokussiert HENSEL (2011) in seinem Ansatz speziell die Elektronik- und Softwareanteile des mechatronischen Entwicklungsprozesses. Sein Ansatz erweitert das 3-Ebenen-Modell von BENDER (2005) um ein zentrales Informations- und Simulationsmodell, welches im linken Ast des V-Modells erarbeitet wird, die Basis für den domänenspezifischen Entwurf bildet und im rechten Ast zu Testzwecken verwendet wird (vgl. Abbildung 3-6). Auf Systemebene wird dabei zunächst ein disziplinübergreifendes Funktionsmodell erarbeitet, das in den anschließenden Phasen des Subsystem- und Komponententwurfs speziell für die Elektro- und Softwareentwicklung detailliert wird. Im Einzelnen wird die Entwicklung der Hardwarekonfiguration, der Steuerungslogik, der Verfahrenskennlinien sowie der Mensch-Maschine-Schnittstelle in dem zentralen Systemmodell adressiert. Sobald diese vier Elemente innerhalb des Informations- und Simulationsmodells ausreichend detailliert beschrieben sind, werden die entsprechenden Daten zunächst aufbereitet, danach automatisiert ausgeleitet und in die jeweiligen Zielsysteme integriert. Die dadurch generierten Bestandteile können im rechten Ast des V-Modells gegen das Simulationsmodell in einer „Hardware in the Loop“-Anordnung getestet und dadurch optimiert werden.

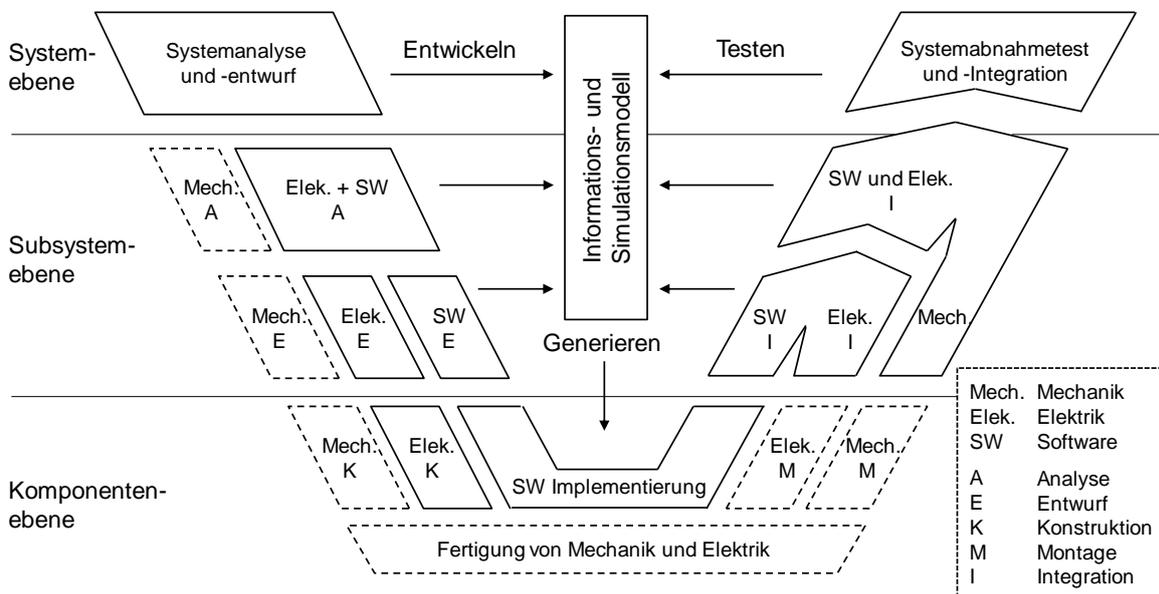


Abbildung 3-6: Vi-Modell zur Entwicklung von Automatisierungslösungen
(in Anlehnung an HENSEL 2011, S. 73)

Die innerhalb dieses Vorgehensmodells eingesetzte Modellierungstechnik zum Aufbau des Informations- und Simulationsmodells basiert auf dem sog. FOCUS-Framework zur modellbasierten Entwicklung von verteilten Systemen (BROY &

STOLEN 2001). Das wesentliche Element dieses Ansatzes bilden Komponenten, die hierarchisch gegliedert sind und über Ports und Kanäle Informationen austauschen. Über diskrete Zustandsautomaten wird das Verhalten einzelner Komponenten beschrieben. Um den Ansatz bei der Entwicklung mechatronischer Systeme nutzen zu können, wurde die Modellierungstechnik zunächst von HUMMEL (2011) adaptiert und insbesondere um die Modellierung hybrider Zustandsautomaten sowie dreidimensionaler Körper erweitert. Die hier entwickelte Modellierungstechnik „STEM“ wurde von HENSEL (2011) genutzt und um die individuellen Aspekte der Elektro- und Softwareentwicklung ergänzt. Dazu beinhaltet die Modellierungstechnik u. a. ein Meta-Modell zur Abbildung von graphischen Mensch-Maschine-Schnittstellen auf Basis der in Abschnitt 2.4 eingeführten Wireframes.

IMoMeSA-Entwicklungsansatz nach HACKENBERG ET AL.

Eine weitere Herangehensweise zur modellbasierten Entwicklung mechatronischer Systeme wurde im Rahmen des Forschungsprojekts IMoMeSA von HACKENBERG ET AL. (2015) erarbeitet. Der Ansatz basiert ebenfalls auf der Modellierungstechnik von HUMMEL (2011) und erweitert diese im Bereich des Anforderungsmanagements und der mechatronischen Modularisierung. Die resultierende Technik umfasst acht Teilmodelle, die zur Modellierung mechatronischer Systeme verwendet werden können (vgl. Abbildung 3-10). In den ersten Entwicklungsschritten stehen zunächst die Modellierung informeller Anforderungen, die Spezifikation der Schnittstelle des Systems mittels Ports sowie die Formulierung von Einsatzszenarien und Eigenschaften im Vordergrund. Mit dem Teilmodell „Monitore“ kann anschließend der zu realisierende Prozess des Systems erfasst und in einzelne Aktivitäten unterteilt werden. Dieser Schritt bildet die Grundlage für die Entscheidung, ob das Systemverhalten bereits auf dieser Abstraktionsebene modelliert werden kann oder ob eine Zerlegung des Systems in einzelne mechatronische Komponenten notwendig ist. Im zuletzt genannten Fall werden neue Komponenten erzeugt, welche dieselben Modellierungselemente wie das mechatronische System beinhalten. Für jede dieser Komponenten können somit iterativ die zuvor genannten Entwicklungsschritte durchlaufen werden, wodurch sukzessive eine hierarchische Komponentenstruktur des zu modellierenden mechatronischen Systems erarbeitet wird. Sobald ein System dadurch in hinreichend kleine Einheiten zerlegt wurde, wird den modellierten Komponenten ein Verhalten mittels Zustandsautomaten hinterlegt. Darüber hinaus können Bauteile ergänzt werden, welche die geometrische Ausprägung der jeweiligen Komponente darstellen.

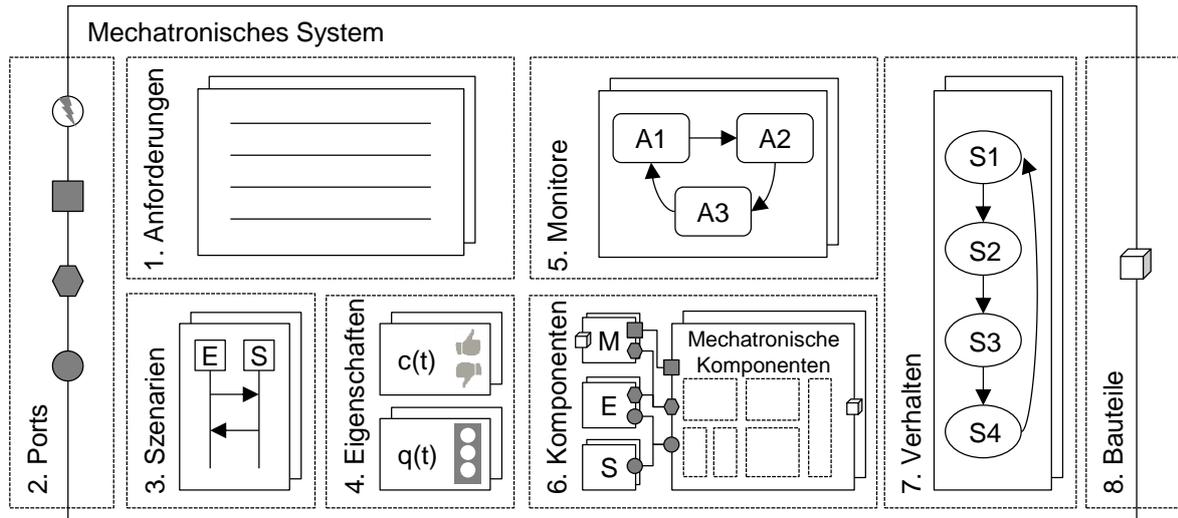


Abbildung 3-7: Modellierungstechnik für mechatronische Systeme
(in Anlehnung an HACKENBERG ET AL. 2015, S. 18)

Der eben beschriebene Ablauf ordnet sich im übergeordneten Vorgehensmodell von HACKENBERG ET AL. (2015) in die Phase der Konzeption ein, deren Ergebnis ein mechatronisches Konzept des zu entwickelnden Systems ist. An diese Phase schließt sich, ähnlich wie bei GAUSEMEIER ET AL. 2014, die Verfeinerung an, in der unter Zuhilfenahme disziplinspezifischer Modelle ein vollständiger virtueller Prototyp des zu entwickelnden Systems ausgearbeitet wird. In der Phase des physikalischen Aufbaus wird schließlich die reale Maschine bzw. Anlage hergestellt und in Betrieb genommen. Da durch den modellbasierten Entwicklungsansatz speziell die Konzeption und Verfeinerung unterstützt werden, sind für diese Phasen konkrete Workflows definiert. Für die Phase der Konzeption wird festgelegt, wie die Elemente der Modellierungstechnik beim Aufbau eines mechatronischen Konzepts zum Einsatz kommen und welche Rolle die Simulation zur Qualitätssicherung spielt. Für die Phase der Verfeinerung wird beschrieben, wie einzelne Komponenten in einem inkrementellen Ansatz detailliert werden und wie das Systemmodell die Konsistenz zwischen den beteiligten Disziplinen sichern kann.

Ausblick auf weiterführende Ansätze

Aufgrund des großen Potenzials einer modellbasierten Herangehensweise bei der Entwicklung mechatronischer Systeme wurde in den vergangenen Jahren auch weitere Modellierungstechniken und Vorgehensmodelle erarbeitet. FOLLMER (2012) beschreibt bspw. einen Ansatz, der sich mit der Erstellung von mechatronischen Systemmodellen und deren Simulation befasst. Auch KERNSCHMIDT ET AL. (2014) stellen ein Vorgehensmodell auf Basis einer SysML-Modellarchitektur dar, um

insbesondere die Wiederverwendung in der mechatronischen Systementwicklung zu steigern. Weitere Ansätze auf Basis der SysML werden bspw. von CAO ET AL. (2011) oder von QAMAR ET AL. (2009) vorgestellt. Eine generelle Alternative zur Modellierung mechatronischer Systeme bietet die Automation Markup Language (AutomationML) von DRATH (2010), die ein Datenformat für eine hierarchische, komponentenbasierte Modellierung von technischen Anlagen definiert. DRATH (2010) beschreibt dabei auch den Einsatz des Modells während des Entwicklungsprozesses. Schließlich haben sich auch im Umfeld des Systems Engineerings (vgl. Abschnitt 3.3.1) zahlreiche modellbasierte Entwicklungsmethoden, wie bspw. der SYSMOD-Ansatz von WEILKIENS (2016), etabliert. Eine Darstellung weiterer relevanter Ansätze in diesem Bereich liefert ESTEFAN (2007).

3.3.3 Analyse der Ansätze hinsichtlich der Berücksichtigung von Mensch-Maschine-Schnittstellen

Wie bereits anhand der Ausführungen in den letzten beiden Abschnitten deutlich wurde, wird die Umsetzung von Mensch-Maschine-Schnittstellen in Entwicklungsansätzen für mechatronische Systeme nur zum Teil berücksichtigt. So wird z. B. in der VDI 2206 das HMI zwar als wichtiger Bestandteil eines mechatronischen Systems aufgefasst, jedoch werden in keiner Entwicklungsphase eigene Methoden oder Herangehensweisen beschrieben, die speziell die HMI-Entwicklung unterstützen. Ähnlich verhält es sich bei Ansätzen des Systems Engineerings und der Produktentwicklung, die eher übergreifende Methoden und Herangehensweisen definieren, ohne dabei auf die einzelnen Bestandteile der Systeme bzw. Produkte einzugehen. Auch von KLEIN (2016) werden Mensch-Maschine-Schnittstellen nicht explizit berücksichtigt, was sich durch den Bezug des Ansatzes auf das mechatronische Referenzmodell begründen lässt (vgl. Abschnitt 3.1). Für die Ansätze im Bereich der modellbasierten Entwicklung lassen sich zum Teil vergleichbare Aussagen treffen. So wird bspw. von EIGNER (2013) lediglich abstrakt definiert, wie ein modellbasierter Entwicklungsansatz entlang des V-Modells gestaltet sein kann, ohne dabei jedoch konkret auf einzelne Modelle und beteiligte Disziplinen einzugehen. Auch die Ansätze von GAUSEMEIER ET AL. (2014), VDMA (2010) und HACKENBERG ET AL. (2015) berücksichtigen Mensch-Maschine-Schnittstellen nicht explizit und konzentrieren sich im Bereich der Softwareentwicklung hauptsächlich auf die Konzeption und Umsetzung der Maschinensteuerung. Die jeweils zugrundeliegenden Modellierungstechniken bieten zwar grundsätzlich die Möglichkeit, einzelne Aspekte von Mensch-Maschine-Schnittstellen abzubilden, jedoch wird nicht darauf eingegangen, wie eine Modellierung durchgeführt werden kann.

Im Gegensatz zu den im letzten Absatz analysierten Arbeiten gibt es auch einzelne Entwicklungsansätze für mechatronische Systeme, in denen Mensch-Maschine-Schnittstellen zumindest ausschnittsweise adressiert werden. Unter diesen ist zuerst die Entwurfsmethodik von ISERMANN (2008) zu nennen. Diese beschreibt, dass bereits im Rahmen des Systementwurfs die Mensch-Maschine-Schnittstelle zu spezifizieren ist, ohne dabei jedoch darauf einzugehen, wie diese Aufgabe konkret umgesetzt werden kann. Auch im Komponentenentwurf werden HMIs erneut adressiert, wobei allerdings nur dargestellt wird, welche Technologien zum Einsatz kommen können. Einen etwas detaillierteren Einblick in die Entwicklung von Mensch-Maschine-Schnittstellen liefert demgegenüber das Quality-Gate-Modell (VDMA 2005). Hier wird definiert, dass ein HMI bereits im Rahmen der Systemspezifikation hinsichtlich Anforderungen, Benutzergruppen, Interaktionstechnologien, möglichen Dialogen und geforderten Funktionalitäten zu beschreiben ist. Anhand von Beispielen und möglichen Beschreibungsmitteln wird darauf eingegangen, wie eine Spezifikation dieser Aspekte erfolgen kann. Auch im modellbasierten Entwicklungsprozess von HENSEL (2011) wird die Entwicklung von Mensch-Maschine-Schnittstellen explizit betrachtet. Der Ansatz konzentriert sich dabei allerdings auf graphische HMIs und deren Abbildung im Rahmen eines mechatronischen Systemmodells sowie die Absicherung der Funktionalität in der Simulation. Aspekte der Gebrauchstauglichkeit oder der Nutzereinbindung werden hingegen nicht berücksichtigt.

3.4 Entwicklungsansätze für Mensch-Maschine-Schnittstellen

Aufgrund der meist fehlenden Berücksichtigung von HMIs in den Vorgehensweisen zur Entwicklung mechatronischer Systeme ist es im Rahmen dieser Arbeit weiterhin erforderlich, Ansätze zu betrachten und zu analysieren, die sich ausschließlich diesem Thema widmen. Auch hier kann zwischen generellen Vorgehensmodellen und Entwicklungsmethoden sowie speziellen Ansätzen zur modellbasierten Entwicklung unterschieden werden, von denen die jeweils relevanten Vorarbeiten in den Abschnitten 3.4.1 bzw. 3.4.2 vorgestellt werden. Neben Ansätzen, die sich unmittelbar auf Mensch-Maschine-Schnittstellen technischer Systeme beziehen, werden auch spezielle Herangehensweisen aus der Informatik beschrieben, die primär Computeranwendungen adressieren, sich aber auch auf die dieser Arbeit zugrundeliegende Zielsetzung übertragen lassen. Abschließend werden die dargestellten Ansätze in Abschnitt 3.4.3 hinsichtlich einer möglichen Anwendbarkeit im Kontext mechatronischer Entwicklungsprozesse analysiert.

3.4.1 Generelle Vorgehensmodelle und Entwicklungsmethoden

Zur Entwicklung gebrauchstauglicher HMIs haben sich zahlreiche Vorgehensmodelle und Entwicklungsmethoden etabliert. Darunter finden sich auch vereinzelt Ansätze, die speziell HMIs von technischen Systemen adressieren. Als Beispiele werden nachfolgend die Ansätze von ZÜHLKE (2012) und der VDMA-Leitfaden „Software-Ergonomie“ (VDMA 2004) vorgestellt. Ein breiteres und differenzierteres Spektrum an Entwicklungsansätzen findet sich allerdings im Bereich der Informatik. Aus der Vielzahl potenzieller Herangehensweisen werden nachfolgend die Arbeiten von MAYHEW (1999), SY (2007) und GOTHELF & SEIDEN (2013) vorgestellt, bevor ein Ausblick auf weiterführende Ansätze angeboten wird.

Nutzergerechte Entwicklung der Mensch-Maschine-Schnittstelle nach ZÜHLKE

Vor dem Hintergrund komplexer werdender Benutzeraufgaben im Maschinen- und Anlagenbau wurde von ZÜHLKE (2012) ein Vorgehensmodell entwickelt, das zukünftige Nutzer konsequent in die HMI-Entwicklung einbindet und es Entwicklern ermöglicht, bereits in einer frühen Phase erarbeitete Erkenntnisse zu evaluieren. Im Gegensatz zu klassischen Modellen der Softwareentwicklung, wie dem Wasserfallmodell, können Nutzer an beliebiger Stelle im Entwicklungsprozess zum Testen der erarbeiteten Konzepte einbezogen werden. Dadurch verläuft die Evaluation parallel zu den kreativen, gestalterischen Phasen, woraus ein Vorgehensmodell mit vier Phasen und fünf Arbeitsschwerpunkten resultiert (vgl. Abbildung 3-11). Die Phasen dürfen dabei aber nicht isoliert, sondern müssen vielmehr überlappend betrachtet werden, um HMIs konsistent und nutzerzentriert umsetzen zu können.

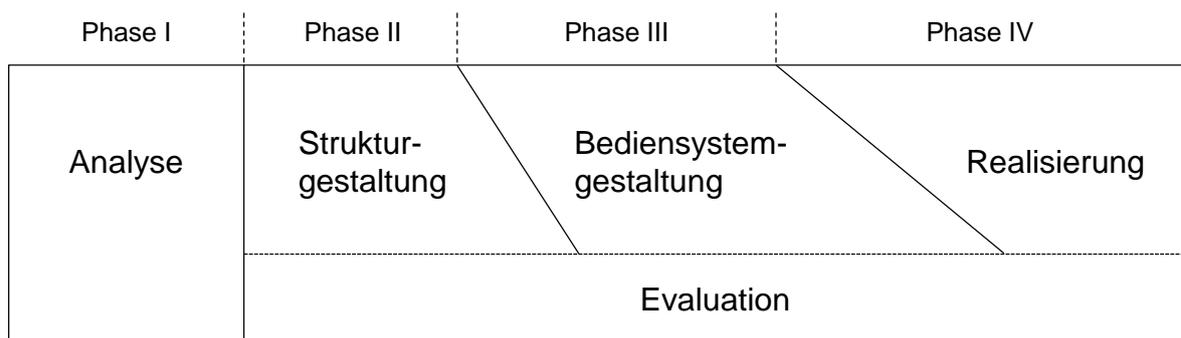


Abbildung 3-8: Entwicklungsprozess für Mensch-Maschine-Schnittstellen
(in Anlehnung an ZÜHLKE 2012, S. 37)

In der *Analysephase* steht zunächst die Untersuchung der Nutzer, deren Arbeitsaufgaben sowie der Arbeitsumgebung im Vordergrund. Darüber hinaus wird auch die technische Anwendung betrachtet, um Aussagen über Funktionalität und

Einsatzgebiet abzuleiten. Zur Erhebung dieser Daten wird ein dreistufiges Vorgehen zur Nutzerbefragung vorgeschlagen, wobei als Ergebnis technologie-, nutzungs- und handlungsorientierte Daten entstehen, welche die Basis zur Ausrichtung der Entwicklung an den Bedürfnissen der späteren Nutzer bilden. In der Phase der *Strukturgestaltung* wird daraufhin das Ziel verfolgt, die Bedienung des technischen Systems aufgabenorientiert und plattformunabhängig zu strukturieren. Dazu werden die Analyseergebnisse in einem sog. Benutzungsmodell harmonisiert, in dem die Aufgaben, Tätigkeiten und Handlungen in sog. Benutzungsobjekten beschrieben werden. Über Attribute enthalten Benutzungsobjekte zudem Informationen über Zuständigkeiten, Rechte und Beziehungen. Im letzten Schritt der Strukturgestaltung werden Benutzungsobjekte mit Funktionen des technischen Systems verknüpft, um sicherzustellen, dass alle spezifizierten Aufgaben durch entsprechende Systemfunktionen realisiert werden können. Das entwickelte Benutzungsmodell kann anschließend in der Phase der *Bediensystemgestaltung* umgesetzt werden. Dieser Prozess beginnt mit der Auswahl der Hard- und Software, die zur Bedienung zum Einsatz kommen soll. Auf dieser Basis kann das Groblayout des HMIs festgelegt werden, was neben der Anordnung von Hardwareelementen auch die Maskenaufteilung und das Navigationskonzept bei bildschirmgestützten Mensch-Maschine-Schnittstellen beinhaltet. Im nächsten Schritt erfolgt die beispielhafte Umsetzung von einzelnen Dialogen, um Benutzern die grundsätzliche Handhabung mit dem System zu verdeutlichen. Abschließend werden die einzelnen Interaktionsobjekte, wie z. B. Buttons, festgelegt sowie hinsichtlich ihrer Eignung zur Durchführung der Benutzeraufgaben überprüft. Die spezifizierten Dialoge und Interaktionsobjekte werden schließlich in der *Realisierung* in ein lauffähiges Programm überführt. Neben der Implementierung der dynamischen Aspekte der Mensch-Maschine-Schnittstelle wird auch die Anbindung an die Maschinensteuerung vorgenommen. Parallel zu den eben beschriebenen Phasen erfolgt kontinuierlich eine *Evaluation* von Zwischenergebnissen durch spätere Nutzer. Während in der Systemstrukturgestaltung die Aufgaben innerhalb des Benutzungsmodells analysiert werden, wird im weiteren Verlauf der Entwicklung der Einsatz von lauffähigen Prototypen vorgeschlagen, um ein fundiertes Feedback der Nutzer zu erhalten.

VDMA-Gestaltungsprozess für Bedienoberflächen

Im VDMA-Leitfaden Software-Ergonomie (VDMA 2004) werden zentrale Regeln für die Auslegung von Bedienoberflächen im Maschinen- und Anlagenbau definiert und anhand verschiedener Anwendungsbeispiele veranschaulicht. Die Regeln bauen dabei auf den Grundsätzen zur Dialoggestaltung gemäß der DIN EN ISO 9241 auf und konkretisieren diese, ähnlich wie in der VDI/VDE 3850, hinsichtlich einer

Anwendung im Maschinen- und Anlagenbau. Darüber hinaus wird ein Gestaltungsprozess für Bedienoberflächen definiert, um gebrauchstaugliche HMIs bei einer systematischen Einbindung der Benutzer realisieren zu können. Der Ansatz gliedert sich in sechs Aktivitäten, die sich über alle Phasen eines Entwicklungsprojekts, von der Planung bis zur Implementierung, erstrecken (vgl. Abbildung 3-9). Für jede Aktivität werden verschiedene Methoden vorgeschlagen, die sich in der Industrie bewährt haben, um die jeweiligen Ergebnisse zu erzielen. Darüber hinaus werden Praxistipps angeführt, um die einzelnen Aktivitäten im Maschinen- und Anlagenbau bestmöglich durchführen zu können.

Aktivität	Analysieren des Nutzungskontexts	Ableiten von Anforderungen		Begleiten Entwicklung	
Methoden	<ul style="list-style-type: none"> ▪ Kontextworkshops ▪ Beobachten ▪ Interviews ▪ Fokusgruppen ▪ Aufgabenanalysen 	<ul style="list-style-type: none"> ▪ Anforderungsworkshops 	<ul style="list-style-type: none"> ▪ Gestaltungsworkshops ▪ Szenario-basiertes Design 	<ul style="list-style-type: none"> ▪ Papier-Prototypen ▪ Interaktive Prototypen ▪ Experten Review ▪ Usability Test mit Benutzern 	<ul style="list-style-type: none"> ▪ Gestaltungsworkshops

Abbildung 3-9: Aktivitäten und jeweilige Methoden zur Gestaltung von Bedienoberflächen (in Anlehnung an VDMA 2004, S. 51)

Die erste Aktivität befasst sich dabei ähnlich zum Ansatz von ZÜHLKE (2012) mit dem *Analysieren des Nutzungskontexts*. Dazu werden mittels der in Abbildung 3-9 genannten Methoden Informationen hinsichtlich der Benutzergruppen, deren Aufgaben sowie der Nutzungsumgebung gesammelt und systematisch aufbereitet. Auf dieser Basis erfolgt in der zweiten Aktivität das *Ableiten von Anforderungen*, wodurch die Gestaltungsziele für die Bedienoberfläche festgelegt werden. In Anforderungs-Workshops werden die Kontextinformationen aus der Analyse (z. B. Beleuchtungsqualität) genutzt, um Anforderungen an die Bedienbarkeit (z. B. Darstellung mit großem Kontrast) abzuleiten. Diese werden weiterhin priorisiert und mit Prüfkriterien für die spätere Evaluation verbunden. Daraufhin werden in der Aktivität des *Gestaltens* das Interaktionskonzept und die visuelle Darstellung für die Bedienoberfläche entwickelt. Zu diesem Zweck schlägt der Ansatz u. a. die Methode der Gestaltungs-Workshops sowie die Berücksichtigung von Ergonomie-Richtlinien aus bspw. der DIN EN ISO 9241 vor. Um die Gestaltungsideen für alle Beteiligten erfahrbar zu machen, sollen in der Aktivität des *Visualisierens* bereits frühzeitig graphische Umsetzungen dieser Ideen ausgearbeitet werden. Zu diesem Zweck können sowohl einfache Papier-Prototypen als auch funktionsfähige SW-

Prototypen erstellt werden. Die Ergebnisse der Visualisierung fließen unmittelbar in das *Evaluieren* ein, welches das Ziel verfolgt, die Gestaltungsideen hinsichtlich der definierten Anforderungen zu überprüfen. Dabei wird ein iteratives Vorgehen empfohlen, um Verbesserungspotenziale kontinuierlich in den Gestaltungsprozess einfließen zu lassen. Als Methoden für diese Aktivität werden Reviews durch Ergonomie-Experten und Usability Tests mit Benutzern vorgeschlagen. Die letzte Aktivität, die *Begleitung der Entwicklung*, adressiert schließlich die Realisierung der Bedienoberfläche, wobei insbesondere auf Rückwirkungen für die Gestaltung eingegangen wird, die bspw. aus geänderten Kundenanforderungen resultieren.

Usability Engineering Lifecycle nach MAYHEW

Im Gegensatz zu den bisher dargestellten Beispielen adressiert der Ansatz von MAYHEW (1999) nicht unmittelbar Mensch-Maschine-Schnittstellen im Kontext technischer Systeme. Der Ansatz gilt allerdings als eine der wichtigsten Grundlagen im Umfeld der nutzerzentrierten Entwicklung von HMIs (RICHTER & FLÜCKIGER 2013, S. 16) und wird vor diesem Hintergrund vorgestellt. Generell gliedert sich der Usability Engineering Lifecycle von MAYHEW (1999) in die Anforderungsanalyse, das Design/Testen/Entwickeln und die Installation. Jede Phase ist dabei weiterhin in konkrete Aufgaben unterteilt, die zur Erreichung einer hohen Gebrauchstauglichkeit beitragen. Darüber hinaus werden der Ablauf der Entwicklung sowie notwendige Dokumente als Ergebnisse einzelner Aufgaben definiert (vgl. Abbildung 3-10).

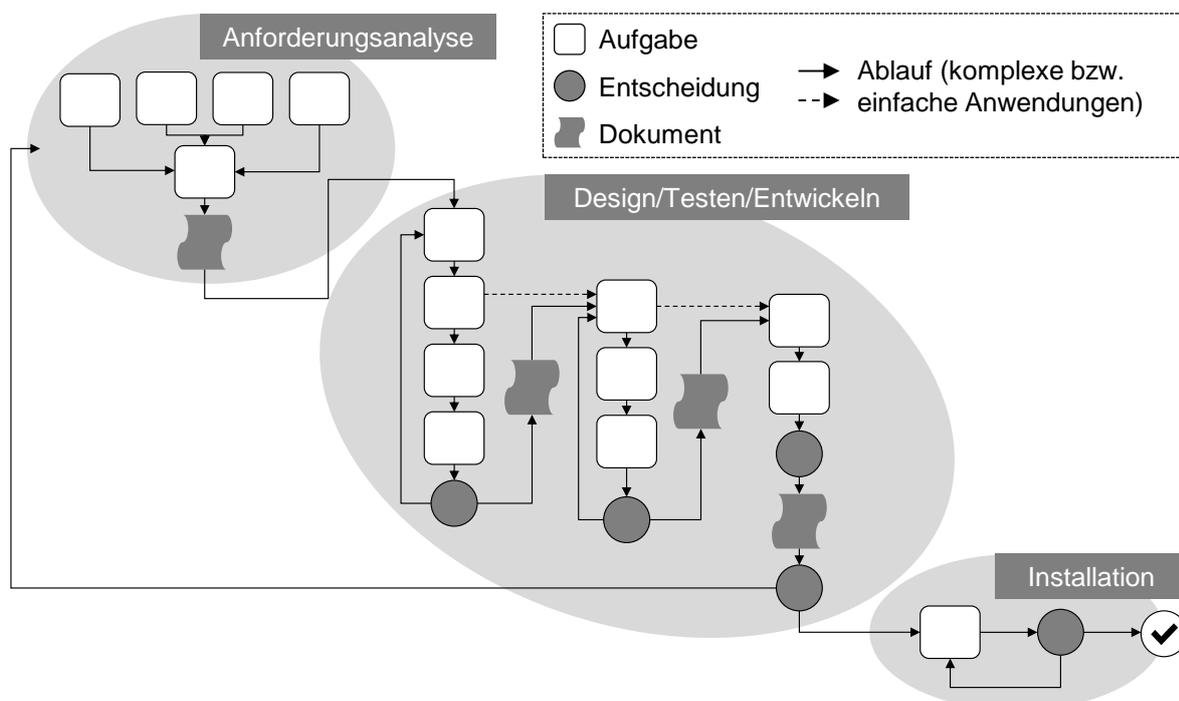


Abbildung 3-10: Usability Engineering Lifecycle (nach MAYHEW 1999, S. 34)

In der *Anforderungsanalyse* werden zunächst Profile einzelner Benutzergruppen anhand zentraler Kenngrößen, wie bspw. die Häufigkeit der Nutzung, erstellt. Für jede der identifizierten Gruppen wird in der Aufgabenanalyse eine Nutzerstudie hinsichtlich vorhandener Aufgaben und Arbeitsgewohnheiten durchgeführt, die in ein Modell der zugrundeliegenden Nutzerziele einfließen. In zwei weiteren Aufgaben werden die Potenziale und Grenzen der eingesetzten Interaktionstechnologie analysiert sowie relevante Gestaltungsrichtlinien recherchiert und überprüft. Die Ergebnisse dieser Aufgaben fließen schließlich in die Festlegung der Usability-Ziele ein, die in einem Styleguide für das zu entwickelnde HMI dokumentiert werden.

Die zweite Phase des Usability Engineering Lifecycle (*Design/Testen/Entwickeln*) lässt sich in drei Ebenen unterteilen. Die oberste Ebene beinhaltet vier Aufgaben, die sich mit abstrakten Designfragen beschäftigen. Dabei werden die Arbeitsabläufe bezüglich möglicher Optimierungspotenziale analysiert und ggf. verbessert. Weiterhin werden mittels konzeptioneller Modelle erste Design-Alternativen generiert, die in einem dritten Schritt durch Papier- oder Funktionsprototypen visualisiert werden. Diese wiederum können mit Nutzern erprobt werden, um zu entscheiden, ob ein Rücksprung in die vorigen Aufgaben notwendig ist. Sobald alle Fehler des konzeptionellen Modells beseitigt wurden, kann mit den Aufgaben der zweiten Ebene fortgefahren werden. Die erste dieser Aufgaben adressiert die Festlegung von produktspezifischen Standards für einzelne HMI-Elemente, um so eine Konsistenz über die Mensch-Maschine-Schnittstelle zu ermöglichen. Daraufhin werden aus diesen Standards und dem konzeptionellen Modell lauffähige Prototypen für Teilbereiche der Produktfunktionalität implementiert, die analog zur ersten Ebene evaluiert werden. Auf Basis valider und dokumentierter HMI-Standards, welche den formulierten Usability-Zielen gerecht werden, kann auf der dritten Ebene das Design des HMIs ausgearbeitet und mit Nutzern evaluiert werden.

Nach einer finalen Prüfung, ob neben den Usability-Zielen auch alle funktionalen Anforderungen erfüllt sind, kann die *Installation* erfolgen und das Produkt kann verwendet werden. Hinsichtlich der Gebrauchstauglichkeit ist es in dieser Phase erforderlich, Feedback der Nutzer einzuholen und auszuwerten sowie auf dieser Basis neue Releases zu planen oder die Erkenntnisse für Folgeprojekte zu dokumentieren.

Agile nutzerzentrierte Entwicklung nach SY

Basierend auf grundlegenden Herangehensweisen des Usability Engineerings, wie bspw. die Entwicklungsmethodik von MAYHEW (1999), wurden in der Informatik mittlerweile vielzählige Ansätze erarbeitet, um den technologischen Fortschritten

und neuen Herangehensweisen bei der Entwicklung von Software im Allgemeinen Rechnung zu tragen. Ein Beispiel bildet hierbei der Ansatz von SY (2007), der das Ziel verfolgt, Aspekte der agilen Softwareentwicklung mit den Konzepten des Usability Engineerings zu kombinieren. Der Ansatz verfolgt das zentrale Ziel, die an sich konträren Herangehensweisen einer späten Implementierung im Usability Engineering und einer frühen Umsetzung von lauffähigen Inkrementen in der agilen Entwicklung aufzulösen. Dazu unterteilt SY (2007) den Entwicklungsprozess in zwei parallele Spuren für Design und Implementierung (vgl. Abbildung 3-11). Abgesehen von einer gemeinsamen Iteration 0 zur Anforderungserhebung werden in beiden Spuren in jeder Iteration eigene Ergebnisse erarbeitet, die erst in der nächsten Iteration für die jeweils andere Spur relevant werden. So wird bspw. das in einer Iteration n entwickelte Design für einen konkreten Ausschnitt der Mensch-Maschine-Schnittstelle in der Iteration $n+1$ implementiert, während gleichzeitig der in der Iteration n erarbeitete Code in der Iteration $n+1$ getestet wird. Neben dem Test und Design besteht eine dritte Aufgabe für die Design-Spur in der Durchführung von User Researches (vgl. BAXTER ET AL. 2015), welche die Datenbasis für das Design der nächsten und die Implementierung der übernächsten Iteration bilden.

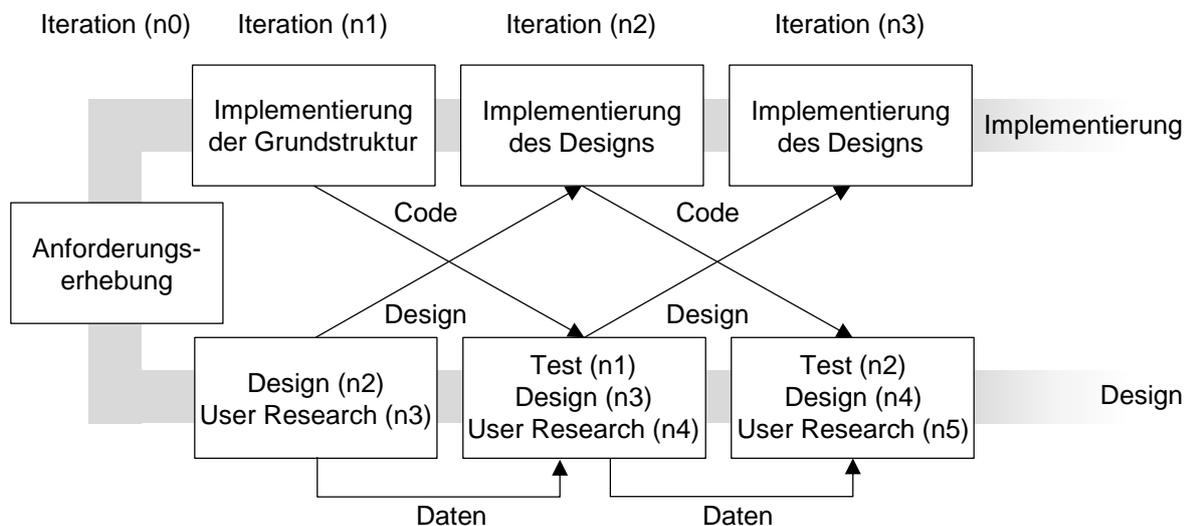


Abbildung 3-11: Parallele Spuren zur agilen nutzerzentrierten Entwicklung
(in Anlehnung an SY 2007, S. 118)

Neben der Gliederung der Entwicklung von Mensch-Maschine-Schnittstellen in die beiden dargestellten Spuren beschreibt der Ansatz von SY (2007) auch die Aufteilung von Design-Elementen in sog. Design Chunks. Diese stellen gekapselte Ausschnitte der graphischen Oberfläche einer Mensch-Maschine-Schnittstelle dar und ermöglichen eine inkrementelle Ausgestaltung und Evaluation von Bereichen des Designs im Rahmen des iterativen Entwicklungsvorgehens. Während der ein-

zelen Iterationen schlägt SY (2007) in Ergänzung zu Methoden und Artefakten der agilen Softwareentwicklung (vgl. KLEIN 2016) den Einsatz eines User Experience Boards sowie zugehöriger Issue Cards und Feature Cards vor. Diese bieten sich zur einfachen und schnellen Visualisierung von vorhandenen Usability-Problemen oder Feature-Anfragen sowie deren aktuellen Status an und ermöglichen dadurch eine kontinuierliche Fortschrittskontrolle während der Entwicklung.

Lean User Experience (Lean UX) nach GOTHELF & SEIDEN

Basierend auf den Prinzipien der agilen Softwareentwicklung erarbeiteten auch GOTHELF & SEIDEN (2013) eine Vorgehensweise zur nutzerzentrierten Realisierung von Mensch-Maschine-Schnittstellen. Ihr Ansatz verfolgt den Grundgedanken, schlanke Entwicklungsprozesse bei einer Minimierung von Verschwendungen zu ermöglichen. Neben der agilen Entwicklung stützt sich der Ansatz dazu auf das sog. Design Thinking, das sich als nutzerzentrierte Innovationsmethode zur Problemlösung und Ideenentwicklung etabliert hat (GÜRTLER & MEYER 2013, S. 13-15). Eine dritte Grundlage des Ansatzes bildet die „Lean Startup“-Methode, die eine schnelle Entwicklung von Prototypen vorschlägt, um Annahmen über den Absatzmarkt zu evaluieren und Kundenfeedback schnell in die Weiterentwicklungen des Produkts einfließen zu lassen (RIES 2012). Auf dieser Basis definieren GOTHELF & SEIDEN (2013) 15 Prinzipien, die eine schlanke nutzerzentrierte Entwicklung von Mensch-Maschine-Schnittstellen kennzeichnen. Diese adressieren die Zusammensetzung des Entwicklungsteams, die Fokussierung auf die zentralen Business-Ziele oder die Einbeziehung der späteren Nutzer.

Der Entwicklungsprozess selbst beginnt im Lean UX mit der Formulierung des vorliegenden Problems bzw. des zu erreichenden Ziels. Auf dieser Basis werden Annahmen abgeleitet, die im Verlauf der Entwicklung zu bestätigen oder widerlegen sind. Um diese Aufgabe zu vereinfachen, werden Annahmen in quantitative Hypothesen überführt, wobei der jeweilige Nutzen herauszustellen ist. Ergänzt werden Hypothesen um sog. Personas, mittels derer fiktive Benutzer mit individuellen Fähigkeiten und Bedürfnissen abgebildet werden. Erst auf dieser Basis wird mit der Suche nach konkreten Produkten oder Features begonnen, mit denen das angestrebte Ziel erreicht bzw. das vorliegende Problem gelöst werden kann. Bei der anschließenden Umsetzung dieser Features wird zunächst ein sog. Minimum Viable Product (MVP) fokussiert, welches sich mit möglichst geringem Aufwand realisieren lässt, aber Rückschlüsse hinsichtlich der aufgestellten Hypothesen und der Richtigkeit des gewählten Wegs ermöglicht. Beim Aufbau eines MVP bietet sich insbesondere der Einsatz von Prototypen an, die in kurzen Abständen mit

potenziellen Kunden erprobt werden. Das erhaltende Feedback fließt unmittelbar in die Entwicklung und den Ausbau des MVP ein. Der beschriebene Prozess lässt sich dazu in das agile Vorgehensmodell Scrum (vgl. SCHWABER & BEEDLE 2002) einbetten, wobei in den einzelnen Iterationen das initiale MVP kontinuierlich bis zu einem voll funktionsfähigen Produkt erweitert wird.

Ausblick auf weiterführende Ansätze

Neben den vorgestellten Konzepten wurden in der Informationstechnik zahlreiche weitere Ansätze für eine nutzerzentrierte HMI-Entwicklung ausgearbeitet. Unter diesen sei zunächst auf NIELSEN (1993) verwiesen, der eine umfassende Methodensammlung zur Sicherstellung der Gebrauchstauglichkeit anbietet. Vergleichbare Werke wurden auch von ABRAS ET AL. (2004), BENGLER ET AL. (2012), RICHTER & FLÜCKINGER (2013) oder SHNEIDERMAN ET AL. (2017) veröffentlicht. Weiterhin werden in Normen, wie der DIN EN ISO 9241-210, oder der VDI/VDE 3850 – BLATT 1, Vorgehensweisen definiert, wie HMIs interaktiver Systeme zu gestalten sind. Wegen der gestiegenen Bedeutung der agilen Entwicklung wurden neben SY (2007) weitere Ansätze erarbeitet, die Konzepte des Usability Engineerings mit den Prinzipien der agilen Entwicklung kombinieren. Unter diesen sei exemplarisch auf BEYER (2010), BUDWIG ET AL. (2009) oder WINTER ET AL. (2012) verwiesen.

Während viele Ansätze die HMI-Entwicklung interaktiver Systeme im Allgemeinen unterstützen, adressieren nur wenige Ansätze speziell die HMIs technischer Systeme wie Maschinen oder Anlagen. Mit ZÜHLKE (2012) und VDMA (2004) wurden bereits die zentralen Grundlagen für diese Arbeit beschrieben. Weitere Ansätze, die in das hier entwickelte Konzept eingeflossen sind, finden sich in der Prozessindustrie, wie bspw. TIMPE ET AL. (2002) oder URBAS ET AL. (2011).

3.4.2 Spezielle Ansätze zur modellbasierten Entwicklung

Analog zu mechatronischen Systemen haben sich auch für die Entwicklung von Mensch-Maschine-Schnittstellen modellbasierte Herangehensweisen etabliert, um den gestiegenen Anforderungen an die Gebrauchstauglichkeit gerecht zu werden und gleichzeitig den Entwicklungsaufwand möglichst gering zu halten (MEIXNER ET AL. 2011, S. 3-4). Auch in diesem Bereich finden sich vornehmlich Konzepte, die allgemein Mensch-Maschine-Schnittstellen interaktiver Systeme betrachten. Unter diesen werden die Ansätze von VANDERDONCKT & CALLEROS (2008), BECKER ET AL. (2011) und AHMED & ASHRAF (2007) vorgestellt. Demgegenüber ordnet sich der Ansatz von MEIXNER (2010) in die Methodik zur nutzerzentrierten

Entwicklung von Mensch-Maschinen-Schnittstellen nach ZÜHLKE (2012) ein und adressiert somit HMIs technischer Systeme. Analog zu den bisherigen Abschnitten erfolgt abschließend ein Ausblick auf weiterführende Ansätze.

Modellbasiertes Useware Engineering nach MEIXNER

Vor dem Hintergrund der Potenziale einer modellbasierten HMI-Entwicklung definiert MEIXNER (2010) eine Architektur, um das von ZÜHLKE (2004) formulierte Vorgehen zur Useware Entwicklung (vgl. Abbildung 3-8) durch den kontinuierlichen Einsatz von aufeinander aufbauenden Modellen zu unterstützen. Der Ansatz stützt sich auf die drei Kernmodelle (Aufgaben-, Dialog- und Präsentationsmodell), die zur Beschreibung von HMIs notwendig sind (vgl. Abschnitt 2.4.2). Neben eigenen Modellierungstechniken werden auch Werkzeuge vorgeschlagen, die bei der HMI-Entwicklung eingesetzt werden können (vgl. Abbildung 3-12).

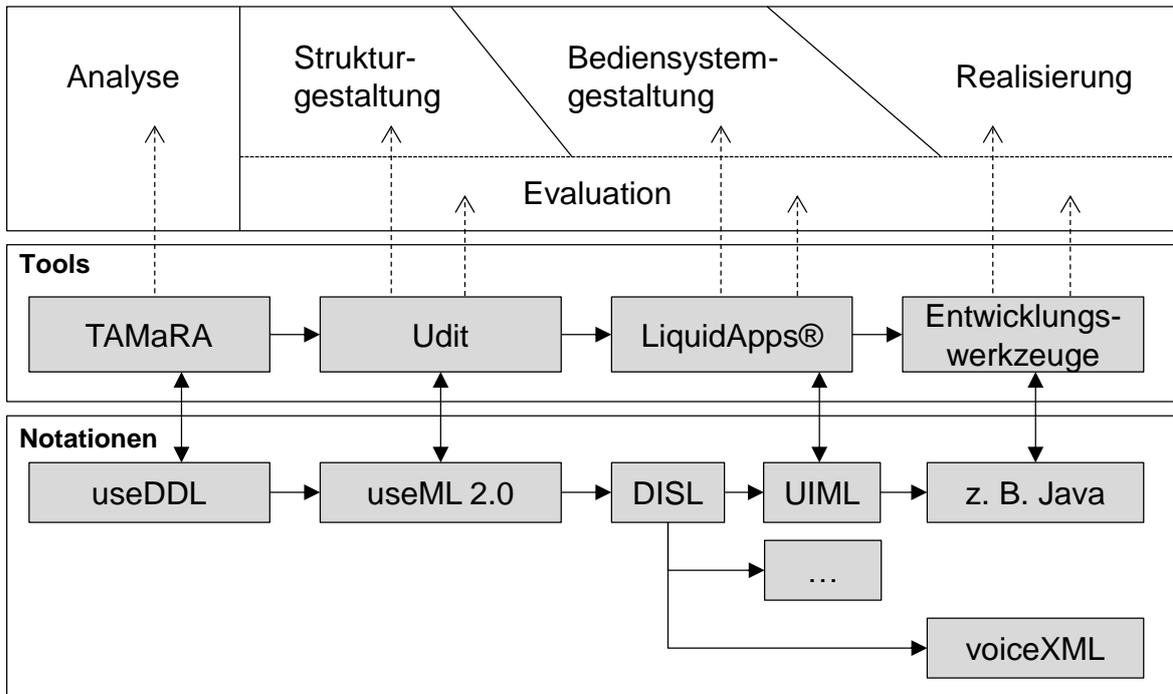


Abbildung 3-12: Architektur des modellbasierten Useware Engineering Prozesses (in Anlehnung an MEIXNER 2010, S. 125)

In der Analysephase kommen in Ergänzung zu den drei Kernmodellen zunächst eine Modellierungstechnik und ein Werkzeug zur Aufnahme der Analysedaten zum Einsatz. Die Grundlage bildet dabei die sog. Useware Data Description Language (*useDDL*), mittels derer die Analysedaten strukturiert, verwaltet und analysiert werden können. Gemäß dem zugrundeliegenden Vorgehensmodell werden diese Daten in der Strukturgestaltung verwendet, um das Benutzungsmodell als spezielle

Form eines Aufgabenmodells für das zu entwickelnde HMI aufzubauen. Die XML-basierte Useware Markup Language 2.0 (*useML 2.0*) bietet dabei die Möglichkeit, diese Modelle formal aufzubauen und zu verwalten. Das zugehörige Tool „Udit“ ermöglicht u. a. das graphische Erstellen, die simulative Absicherung sowie den Export in Dialog- und Präsentationsmodelle für die Bediensystemgestaltung. Für diese Phase schlägt MEIXNER (2010) zunächst die Verwendung eines plattform- und modalitätenunabhängigen abstrakten Dialog- und Präsentationsmodells vor. Dazu wird die sog. Dialog and Interaction Specification Language (*DISL*) eingesetzt, die eine Trennung von Struktur, Präsentation und Dialog ermöglicht. Hinsichtlich der Struktur werden abstrakte Interaktionsobjekte definiert und in eine hierarchische Beziehung zueinander gesetzt. Bezüglich der Präsentation und des Dialogs werden diese Interaktionsobjekte mit graphischen Eigenschaften bzw. Verhaltensmodellen (bestehend aus Regeln, Transitionen und Ereignissen) verknüpft. Im zweiten Teil der Bediensystemgestaltung werden diese abstrakten Modelle in konkrete Dialog- und Präsentationsmodelle überführt, die vom Typ des HMIs (z. B. grafik- oder sprachbasiert) abhängen. MEIXNER (2010) konzentriert sich auf grafikbasierte Mensch-Maschine-Schnittstellen und schlägt zu deren Modellierung die User Interface Markup Language (*UIML*) vor. Diese XML-basierte Sprache übernimmt die grundsätzliche Trennung von Struktur, Präsentation und Verhalten, ermöglicht aber eine deutlich detailliertere Spezifikation der einzelnen Interaktionsobjekte bis zu Schriftarten oder -größen. Diese Modelle können schließlich auf entsprechende Programmiersprachen (z. B. *Java*) transformiert werden und ermöglichen dadurch eine vereinfachte bzw. teilautomatisierte Realisierung.

HMI-Entwicklung mit der UsiXML nach VANDERDONCKT & CALLEROS

Einen vergleichbaren Ansatz zur durchgängig modellbasierten HMI-Entwicklung liefern VANDERDONCKT & CALLEROS (2008). Ihr Konzept basiert auf der User Interface eXtensible Markup Language (UsiXML), die sich aus acht Partialmodellen zusammensetzt. Diese werden verwendet, um die einzelnen Facetten einer Mensch-Maschine-Schnittstelle abzubilden und den Übergang bzw. das Verhältnis zwischen einzelnen Partialmodellen zu beschreiben. Generell lässt sich der Ansatz in die übergreifende Architektur zur modellbasierten Softwareentwicklung (vgl. KEMPA & MANN 2005, S. 300) einordnen, woraus vier Abstraktionsstufen für ein zu entwickelndes HMI resultieren (vgl. Abbildung 3-13). Zwischen den Abstraktionsstufen (mit Ausnahme des finalen HMIs) ermöglichen Graph-Transformationen (HECKEL 2006) einen nahtlosen Übergang. Zur Erzeugung des finalen HMIs können entweder Rendering-Engines zur direkten Interpretation von UsiXML-Modellen oder Verfahren der automatischen Codegenerierung eingesetzt werden.

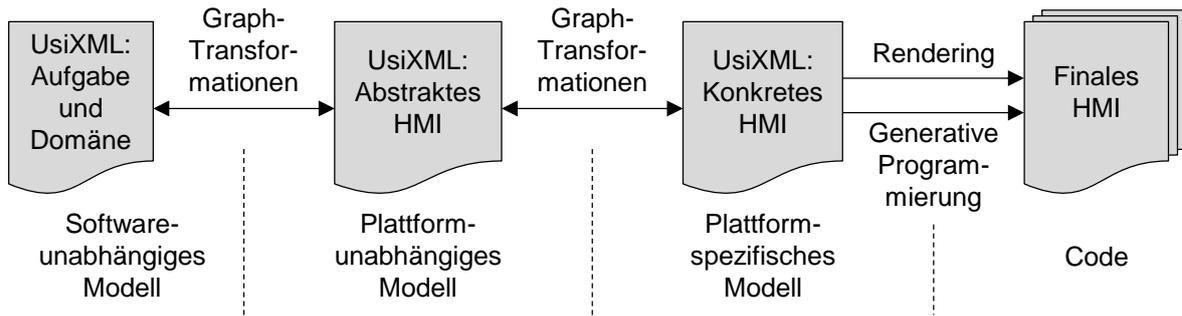


Abbildung 3-13: HMI-Entwicklung auf Basis der UsiXML
(in Anlehnung an VANDERDONCKT & CALLEROS 2008, S. 23)

Auf der ersten Abstraktionsstufe findet zunächst das *Aufgabenmodell* als eines der Kernmodelle einer modellbasierten HMI-Entwicklung Anwendung. In der UsiXML wird dieses Modell in der sog. „ConcurTaskTree“-Notation (PATERNO ET AL. 1997) spezifiziert, die eine hierarchische Zerlegung von Benutzeraufgaben ermöglicht. Es wird ergänzt um ein *Domänen-, Kontext- und Ressourcenmodell*. Ersteres gibt einen generellen Überblick, welche Objekte von einem Benutzer beim Umgang mit dem HMI manipuliert werden können. Demgegenüber erfolgt im Kontextmodell eine Untersuchung des Nutzungszusammenhangs, bestehend aus einer Modellierung des Nutzers sowie einer Abbildung der Plattform und der Umgebung. Im Ressourcenmodell werden schließlich die Ressourcen spezifiziert, die ein Benutzer zur Durchführung modellierter Aufgaben benötigt.

Zur Modellierung der Mensch-Maschine-Schnittstelle unterscheidet der Ansatz, wie auch MEIXNER (2010), zwischen einem *abstrakten* und *konkreten HMI-Modell*. Beide ermöglichen die Abbildung des Dialogs und der Präsentation auf unterschiedlichen Abstraktionsstufen. Auf der ersten Stufe wird die Mensch-Maschine-Schnittstelle unabhängig von der Modalität und der Technologie beschrieben. Dabei werden Ein- und Ausgangsgrößen des HMIs sowie darauf basierende Vorgänge modelliert und in sog. Containern und abstrakten Interaktionsobjekten visualisiert und strukturiert. Im konkreten HMI-Modell fokussieren auch VANDERDONCKT & CALLEROS (2008) graphische HMIs, wobei die Spezifikation und Anordnung von Widgets für einzelne abstrakte Interaktionsobjekte sowie die Festlegung der Navigationsstruktur und des Dialogverhaltens unterstützt werden.

Komplettiert werden die bisher dargestellten Modelle der UsiXML durch das *Mapping-* und das *Transformationsmodell*. Ersteres wird genutzt, um die Beziehungen zwischen einzelnen Modellen oder Modellelementen zu spezifizieren. Im Zweiten werden Transformationsregeln für den Übergang zwischen einzelnen Modellen definiert, um eine durchgängige HMI-Entwicklung zu ermöglichen.

Nutzerzentrierte und modellbasierte HMI-Entwicklung nach BECKER ET AL.

Einen weiteren Ansatz zur modellbasierten Entwicklung von gebrauchstauglichen Mensch-Maschine-Schnittstellen liefern BECKER ET AL. (2011). Im Unterschied zu den bisherigen Konzepten spezifizieren sie keine eigene Modellierungstechnik, sondern nutzen direkt die in der UML 2.3 (OMG 2010) definierten Modelltypen. Auch hinsichtlich des Entwicklungsablaufs baut der Ansatz auf einer etablierten Vorgehensweise zur nutzerzentrierten Gestaltung von interaktiven Systemen, der DIN EN ISO 9241-210, auf. Die dort definierten Aktivitäten wurden allerdings um übergeordnete Phasen sowie Verifikationsschritte erweitert (vgl. Abbildung 3-14). Der Ansatz beschreibt dabei die einzelnen Aufgaben innerhalb dieser Phasen sowie die eingesetzten Artefakte (UML-Diagramme oder selbst definierte Dokumente).

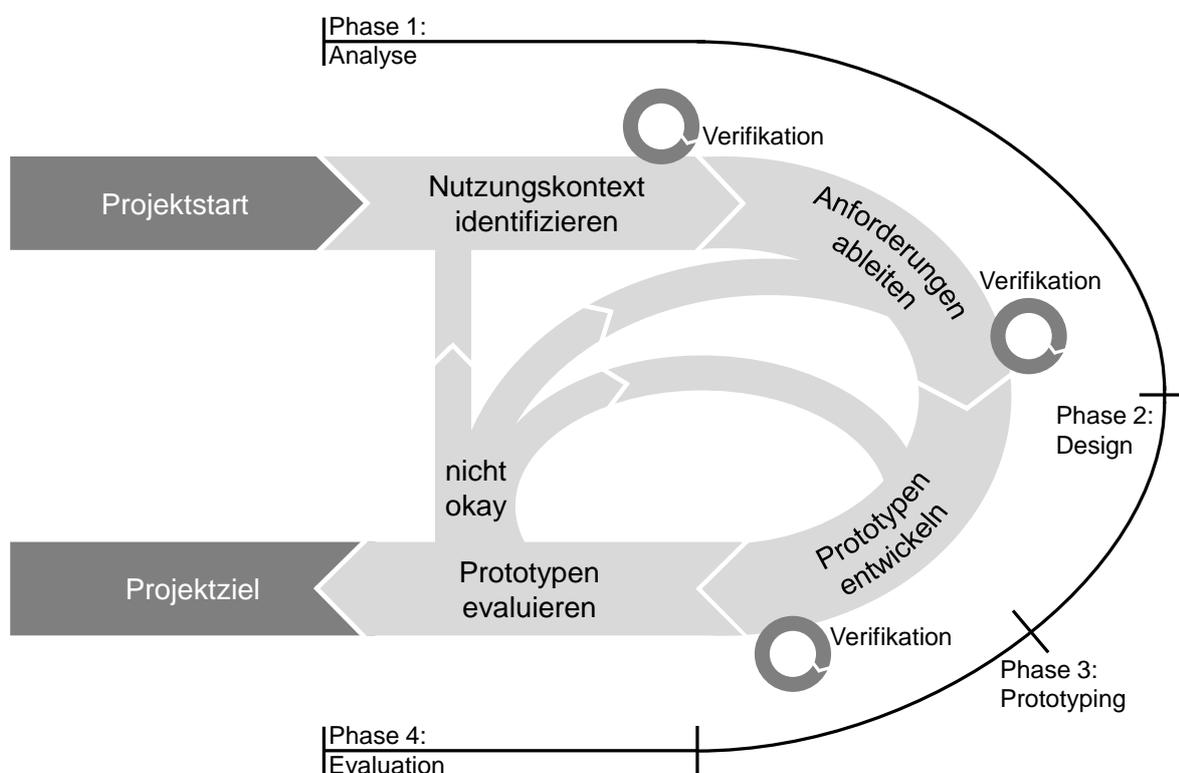


Abbildung 3-14: HMI-Entwicklung nach der ISO 9241-210 – erweitert um Phasen und Verifikationsschritte (in Anlehnung an BECKER ET AL. 2011, S. 3)

In der *Analysephase* schlagen BECKER ET AL. (2011) zunächst eine Identifikation des Nutzungskontexts vor, in der alle Leistungen, Aufgaben und Einschränkungen des HMIs sowie die relevanten Akteure aufgenommen und in prosaischer Form dokumentiert werden. Anschließend erfolgt eine Analyse der einzelnen Aufgaben anhand von Anwendungsfalldiagrammen. In diesen werden alle ermittelten Aufgaben modelliert, in Beziehung zueinander gesetzt und mit den einzelnen

Akteuren assoziiert. Jeder Anwendungsfall wird weiterhin durch ein Aktivitätsdiagramm konkretisiert, welches den genauen Ablauf einer Aufgabe visualisiert. In einem letzten Analyseschritt werden aus den Diagrammen natürlich-sprachliche Anforderungen abgeleitet und in einem Anforderungskatalog dokumentiert.

In der *Designphase* werden diese Erkenntnisse genutzt, um anhand eines statischen und dynamischen HMI-Modells festzulegen, wie die in der Analyse erarbeiteten Anforderungen erfüllt werden können. Im statischen Modell (Klassendiagramm) werden dabei die Interaktionsmöglichkeiten, wie z. B. ein Bildschirm, sowie daraus abgeleitete Interaktionsangebote, wie z. B. eine graphische Ausgabe, und konkrete Interaktionsumsetzungen, wie z. B. eine Schaltfläche, modelliert und über Assoziationen verknüpft. Für das dynamische Modell kommen Sequenzdiagramme zum Einsatz, die für jeden Anwendungsfall modelliert werden. Sie bestehen aus drei Lebenslinien und visualisieren den konkreten Ablauf einer Aufgabe sowie die dazu notwendigen Schnittstellen zwischen Benutzer, HMI und dem System. In einem letzten Schritt werden die Sequenzdiagramme mit den Interaktionsumsetzungen des statischen Modells angereichert, um einen Überblick zu erhalten, mit welchen Elementen einzelne Ein- bzw. Ausgaben realisiert werden sollen.

Im Rahmen des anschließenden *Prototypings* werden die HMI-Modelle genutzt, um in einem dreistufigen Vorgehen (Grobentwurf, Feinentwurf, Implementierung) eine lauffähige Mensch-Maschine-Schnittstelle zu erzeugen. Speziell der Feinentwurf ist dabei hervorzuheben, da hier die Sequenzdiagramme erneut aufgegriffen und zu sog. HMI-Fragmenten erweitert werden. Dazu werden die abstrakten Interaktionsmöglichkeiten durch einzelne Bilder eines HMI-Prototyps ersetzt und an den entsprechenden Interaktionspunkten der Sequenzdiagramme platziert. Auf dieser Basis kann die Mensch-Maschine-Schnittstelle implementiert und in der *Evaluationsphase* erprobt werden. Für diese Phase schlagen BECKER ET AL. (2011) Probandentests vor, in denen geprüft wird, ob alle Grundsätze der Dialoggestaltung nach DIN EN ISO 9241-110 eingehalten sind.

User Interface Engineering mit Design Patterns nach AHMED & ASHRAF

Vor dem Hintergrund, dass modellbasierte Entwicklungsansätze wegen des hohen Modellierungsaufwands in der Praxis nur selten eingesetzt werden, verfolgt der Ansatz von AHMED & ASHRAF (2007) das Ziel, den Modellbildungsprozess durch eine bessere Wiederverwendung von einzelnen Modellelementen zu unterstützen. Dazu wird der Einsatz sog. „Design Patterns“ vorgeschlagen, die in der Softwareentwicklung als bewährte Lösungen für häufig auftretende Problemstellungen

bekannt sind (vgl. GAMMA ET AL. 2015). Zu diesem Zweck wird ein Framework definiert, das eine Pattern-getriebene und eine modellbasierte Herangehensweise vereint und sich aus drei zentralen Bausteinen zusammensetzt. Diese sind zunächst eine Modelllandschaft zur Abbildung von Mensch-Maschine-Schnittstellen sowie ein zugehörigen Vorgehen für deren Aufbau, weiterhin eine Bibliothek von Design Patterns, die bei der Modellbildung eingesetzt werden können, und schließlich eine Methode zur Identifikation, Instanziierung und Anwendung der Patterns während des Aufbaus und der Verfeinerung der Modelle.

Hinsichtlich der *Modelllandschaft* orientiert sich der Ansatz an Vorarbeiten im Bereich der modellbasierten HMI-Entwicklung (vgl. PUERTA 1997) und definiert sieben Modelltypen, die in drei übergeordneten Phasen aufgebaut werden. In der ersten Phase werden dabei ein Domänen-, ein Nutzer- und ein Aufgabenmodell spezifiziert, um den Nutzungskontext zu definieren. Diese Modelle werden in der zweiten Phase genutzt, um ein Umgebungs-, ein Dialog und ein Präsentationsmodell anzufertigen. In der dritten Phase werden diese schließlich in ein sog. Layout-Modell überführt, welches direkt einer konkreten HMI-Instanz entspricht und nach einem automatischen Rendering-Prozess direkt ablauffähig ist.

Zum Aufbau dieser Modelle werden vier wesentliche Typen von *Design Patterns* definiert. Dabei werden zunächst sog. Aufgaben-Patterns eingesetzt, um einzelne Standard-Fragmente innerhalb eines Aufgabenmodells zu beschreiben. Ein Beispiel hierfür ist das Login-Pattern, welches beschreibt, aus welchen Teilaufgaben, Ein- und Ausgaben sich die Anmeldung eines Benutzers zusammensetzt. Weiterhin werden sog. Dialog-Patterns definiert, um speziell den Aufbau von Dialogmodellen zu vereinfachen. Als Beispiel sei an dieser Stelle das Wizard-Pattern genannt, das bei Dialogen einzusetzen ist, bei denen ein Benutzer verschiedene Eingaben in einer vorgegebenen Reihenfolge tätigt. Einen dritten Typ an Patterns bilden die sog. Präsentations-Patterns, um für eine konkrete Aufgabe geeignete HMI-Elemente (Buttons, Textfelder, etc.) auszuwählen. So definiert bspw. das Pattern „Eindeutiges Format“ welche HMI-Elemente einen Nutzer bei einer konkreten Aufgabe davon abhalten, syntaktisch falsche Eingaben zu tätigen. Die letzte Gruppe an Patterns, die sog. Layout-Patterns, werden dazu eingesetzt, die abstrakten HMI-Elemente aus dem Präsentationsmodell geeignet anzuordnen. Insbesondere der übergeordnete Aufbau einer graphischen Mensch-Maschine-Schnittstelle wird dabei fokussiert, für den sich unterschiedliche Patterns, wie bspw. das Liquid Layout (vgl. TIDWELL 2011, S. 186 ff.), etabliert haben.

Um Design Patterns beim Aufbau der jeweiligen Modelle einsetzen zu können, definiert der Ansatz eine vierstufige *Methode*. Dabei werden im ersten Schritt durch Partitionierung eines Modells entsprechende Bereiche identifiziert, für die sich der Einsatz von Patterns anbietet. Anschließend erfolgt die Auswahl eines Patterns für den identifizierten Bereich, wobei dieser Schritt maßgeblich vom Wissensstand der Entwickler abhängt. In einem dritten Schritt wird das ausgewählte Pattern auf den vorliegenden Kontext adaptiert und dadurch instanziiert. Dabei werden die definierten Variablen eines Patterns durch konkrete Werte ersetzt. Im letzten Schritt wird ein adaptiertes Pattern schließlich in das gesamte Modell integriert, wodurch sich auch Änderungen an weiteren Modellbestandteilen ergeben können.

Ausblick auf weiterführende Ansätze

Neben den Konzepten von MEIXNER (2010), VANDERDONCKT & CALLEROS (2008) oder BECKER ET AL. (2011) wurden weitere Ansätze erarbeitet, die grundsätzliche Herangehensweisen für eine modellbasierte HMI-Entwicklung definieren. Ein vergleichbares Konzept liefern z. B. PATERNO ET AL. (2009) mit ihrer Modellierungssprache MARIA, die sich mit den HMIs von Service-orientierten Applikationen beschäftigt. Weitere Ansätze werden u. a. von NAVARRE ET AL. (2009) oder BLUMENDORF ET AL. (2007) vorgestellt. Eine besondere Bedeutung haben in den letzten Jahren adaptive HMIs erlangt, die eine Anpassung der Mensch-Maschine-Schnittstelle an die Bedürfnisse des Nutzers ermöglichen. Für deren Entwicklung bzw. modellbasierte Abbildung haben sich eigene Ansätze, wie z. B. von LOPEZ-JAQUERO ET AL. (2003) oder WINKLER ET AL. (2007), etabliert. Auch der Ansatz von PEISSNER (2014) ist hierbei zu nennen, der zur Entwicklung von adaptiven Mensch-Maschine-Schnittstellen ein modellbasiertes Vorgehen unter Einbeziehung von Design Patterns vorschlägt. Generell rückt das Thema der Design Patterns zunehmend in den Fokus der modellbasierten HMI-Entwicklung, so dass neben dem vorgestellten Konzept von AHMED & ASHRAF (2007) weitere Ansätze, wie zum Beispiel von PETRASCH (2007) oder MÄRTIN ET AL. (2013), erarbeitet wurden.

Wie zu Beginn des Abschnitts erwähnt, ist der Ansatz von MEIXNER (2010) einer von wenigen, der die modellbasierte HMI-Entwicklung speziell für technische Systeme adressiert. Weitere Konzepte, die in die vorliegende Arbeit eingeflossen sind, lassen sich erneut im Bereich der Prozessindustrie finden. Darunter ist insbesondere der Ansatz von SIEMON (2001) hervorzuheben, der eine Entwurfsmethodik für Mensch-Maschine-Schnittstellen von Prozessleitsystemen unter Verwendung visueller Spezifikationsmethoden beschreibt.

3.4.3 Analyse der Ansätze hinsichtlich einer Übertragung auf mechatronische Entwicklungsprozesse

Die in diesem Abschnitt vorgestellten Ansätze betrachten die Konzeption und Umsetzung von Mensch-Maschine-Schnittstellen weitestgehend als eigenständige und abgeschlossene Entwicklungsaktivität. Dies steht zumindest teilweise im Widerspruch zu der in Kapitel 1 aufgezeigten Motivation, dass die Entwicklung von HMIs im Maschinen- und Anlagenbau in den gesamten Entwicklungsprozess einzubetten ist. Vor diesem Hintergrund wird im Rahmen dieses Abschnitts analysiert, wie sich die vorgestellten Ansätze auf den Kontext eines mechatronischen Entwicklungsprozesses übertragen bzw. in diesen einbetten lassen. Dabei werden zunächst die Ansätze betrachtet, die direkt HMIs technischer Systeme adressieren. Weiterhin wird aber auch untersucht, welche Aspekte der informationstechnisch geprägten Ansätze sich auf eine mechatronische Entwicklung übertragen lassen. Abschließend werden speziell die Modelle zur Abbildung von HMIs hinsichtlich vorhandener Parallelen und Unterschiede zu den allgemeinen Modellierungstechniken mechatronischer Systeme analysiert.

Die Herangehensweise von ZÜHLKE (2012) sowie das abgeleitete modellbasierte Vorgehen von MEIXNER (2010) bilden zwei wesentliche Konzepte für die nutzerzentrierte Entwicklung von *Mensch-Maschine-Schnittstellen für technische Systeme*. Die Ansätze befassen sich zwar mit einigen der Problemstellungen, die sich speziell bei HMIs von Maschinen bzw. Anlagen ergeben, wie z. B. Abhängigkeit des HMIs von der Maschinenfunktionalität, aber es werden generell keine Verbindungen zu einem übergeordneten Entwicklungsprozess hergestellt. Bereits in der Analysephase wird ein fertiggestelltes technisches System vorausgesetzt, weswegen diese Ansätze erst gegen Ende des mechatronischen Entwicklungsprozesses eingesetzt werden können. Es ist zwar durchaus vorstellbar, einige der Aktivitäten während der Maschinenentwicklung auszuführen, jedoch wird nicht darauf eingegangen, welche Aktivitäten dafür in Frage kommen und zu welchem Zeitpunkt im gesamten Prozess eine Durchführung sinnvoll ist. Demgegenüber wird in der Vorgehensweise nach VDMA (2004) zumindest ausschnittsweise adressiert, wie sich HMIs in den mechatronischen Entwicklungskontext einordnen lassen. Es wird diesbezüglich auf einzelne Aktivitäten eingegangen, die bereits beim Entwurf eines mechatronischen Systems durchzuführen sind, in der auch von der Maschine bzw. Anlage lediglich ein grundsätzliches Konzept vorliegt. Allerdings beschränkt sich der Ansatz auf diese generelle Einordnung und liefert somit keine Aussagen hinsichtlich Wechselwirkungen zwischen der HMI- und der übergeordneten Maschinenentwicklung.

Im Unterschied zu den im letzten Absatz analysierten Ansätzen adressieren die übrigen Vorgehensmodelle und Entwicklungsmethoden hauptsächlich *Mensch-Maschine-Schnittstellen von reinen Computeranwendungen* (wie bspw. Websites oder Apps), die sich nur bedingt auf einen mechatronischen Entwicklungskontext übertragen lassen. Ähnlich wie beim Ansatz von ZÜHLKE (2012) können die Konzepte vor allem dann eingesetzt werden, wenn die Bedienfunktionen einer Maschine bzw. Anlage bereits feststehen. Erst hier können sowohl grundlegende Ansätze des Usability Engineerings (vgl. MAYHEW (1999) oder NIELSEN (1993)) als auch innovative Konzepte der agilen, schlanken oder modellbasierten HMI-Entwicklung sinnvoll angewendet werden. Allerdings besteht die Möglichkeit, ähnlich wie bei ZÜHLKE (2012), einzelne Aspekte aus diesen Ansätzen parallel zur Maschinenentwicklung durchzuführen. Es ist ein erklärtes Ziel dieser Arbeit, diese Aspekte zu identifizieren und in eine ganzheitliche modellbasierte Herangehensweise einfließen zu lassen. Dabei soll insbesondere eine schlanke Herangehensweise, wie sie von GOTHELF & SEIDEN (2013) vorgeschlagen wird, berücksichtigt werden, um den geringen Entwicklungskapazitäten des Maschinen- und Anlagenbaus im Hinblick auf die Umsetzung von HMIs gerecht zu werden.

Wie bereits in Kapitel 1 veranschaulicht, kann die Integration von HMIs in den gesamten Entwicklungsprozess insbesondere bei einer modellbasierten Herangehensweise gelingen. Vor diesem Hintergrund werden schließlich Parallelen und Unterschiede zwischen der *Modellierung* von Mensch-Maschine-Schnittstellen (vgl. Abschnitt 3.4.2) und mechatronischen Systemen (vgl. Abschnitt 3.3.2) analysiert. Einen ersten zentralen Aspekt bei der Modellierung von HMIs bildet die Abbildung von Benutzeraufgaben. Die eingesetzten Modelle⁷ sind meist auf diesen Kontext zugeschnitten und lassen sich nur bedingt mit mechatronischen Modellierungstechniken, wie der SysML oder CONSENS, abbilden. Demgegenüber lassen sich die Modelle zur Dialogbeschreibung auch mit gängigen Verhaltensbeschreibungen wie Sequenz- oder Zustandsdiagrammen darstellen, die in nahezu allen modellbasierten Ansätzen der Mechatronik vorhanden sind. Eine dritte Säule bei der Modellierung von Mensch-Maschine-Schnittstellen bilden die Präsentationsmodelle, die kaum mit mechatronischen Modellen beschrieben werden können. Zwar können abstrakte Interaktionsobjekte, wie bei BECKER ET AL. (2011), in einfachen Strukturdiagrammen beschrieben werden, allerdings stoßen diese Modelle bei der Abbildung von graphischen Inhalten an ihre Grenzen, so dass in den meisten Ansätzen eigene Beschreibungsmöglichkeiten erarbeitet wurden.

⁷ z. B. Benutzungsmodell von MEIXNER (2010) oder „ConcurTaskTree“-Notation von PATERNO ET AL. 1997

3.5 Identifikation des Handlungsbedarfs

Bezug nehmend auf die durchgeführten Analysen des Stands der Technik in den Abschnitten 3.3.3 und 3.4.3 lässt sich der Handlungsbedarf für die vorliegende Arbeit herausstellen. Auf der einen Seite wurde aufgezeigt, dass in den zahlreichen Herangehensweisen zur Entwicklung mechatronischer Systeme Mensch-Maschine-Schnittstellen entweder nicht oder nur als marginaler Bestandteil des Gesamtsystems berücksichtigt werden. Der Ansatz von HENSEL (2011) bildet diesbezüglich zwar eine Ausnahme, jedoch liegt auch hier kein Fokus auf einem nutzerzentrierten Vorgehen sowie einer Umsetzung innovativer Interaktionstechnologien. Auf der anderen Seite haben sich, vor allem in der Informationstechnik, zahlreiche Ansätze etabliert, welche eben diese Aspekte adressieren und sich der zielgerichteten Entwicklung von Mensch-Maschine-Schnittstellen zum Erlangen einer hohen Gebrauchstauglichkeit widmen. Einige dieser Ansätze, wie bspw. ZÜHLKE (2012), beziehen sich dabei sogar direkt auf die HMIs von technischen Systemen (wie Maschinen oder Anlagen), allerdings teilen alle Konzepte die Gemeinsamkeit, dass die HMI-Entwicklung als in sich geschlossene Aktivität aufgefasst wird, innerhalb derer eine Integration in einen übergeordneten Prozess nur unzureichend berücksichtigt wird.

Hieraus resultiert der zentrale Handlungsbedarf der vorliegenden Arbeit. Die bisher weitestgehend losgelöst voneinander betrachteten Themenfelder der Entwicklung von mechatronischen Systemen und Mensch-Maschine-Schnittstellen sollen kombiniert und in einer neuen Entwicklungsmethodik zusammengeführt werden, um eine integrative Konzeption und Umsetzung von HMIs im Kontext mechatronischer Systeme des Maschinen- und Anlagenbaus zu ermöglichen. Den Rahmen bilden hierfür die in Abschnitt 3.3 angeführten Entwicklungsansätze für mechatronische Systeme, wobei in Kapitel 4 ein konkreter Ansatz als Forschungsgrundlage ausgewählt wird. In diesen können einzelne, ausgewählte Aspekte zur HMI-Entwicklung integriert werden. Dabei wird bewusst auf das gesamte Spektrum an verfügbaren Ansätzen aus Abschnitt 3.4 zurückgegriffen. Die für die vorliegende Problemstellung geeigneten Konzepte, Methoden und Modelle werden ausgewählt, adaptiert und in den übergeordneten Entwicklungsansatz integriert. Dadurch soll sichergestellt werden, dass die HMI-Entwicklung in den gesamten Prozess eingebettet ist und somit frühzeitig berücksichtigt werden kann.

In der anschließenden Anforderungsanalyse in Kapitel 4 wird der eben aufgezeigte Handlungsbedarf erneut aufgegriffen und in konkrete Anforderungen an eine Entwicklungsmethodik für Mensch-Maschine-Schnittstellen überführt.

4 Anforderungsanalyse

Um sicherzustellen, dass die zu erarbeitende Entwicklungsmethodik der Motivation sowie dem zuvor dargestellten Handlungsbedarf gerecht wird, wurden konkrete Anforderungen identifiziert, die als Basis für die spätere Bewertung des Ansatzes verwendet werden (vgl. Abschnitt 4.1). Da im Handlungsbedarf bereits herausgestellt wurde, dass für die Integration der HMI-Entwicklung ein konkretes mechatronisches Vorgehensmodell als Basis verwendet werden soll, wurde im zweiten Teil der Anforderungsanalyse einer der vorgestellten Ansätze zur Entwicklung mechatronischer Systeme als Forschungsgrundlage ausgewählt. Die dafür herangezogenen Kriterien und das Bewertungsergebnis werden in Abschnitt 4.2 vorgestellt.

4.1 Anforderungen an die Entwicklungsmethodik

Zur Ermittlung der aus dem Handlungsbedarf abzuleitenden Anforderungen wurden in dieser Arbeit Literaturrecherchen und Experteninterviews (vgl. Abschnitt 6.1) durchgeführt. Die resultierenden Anforderungen lassen sich generell in drei Kategorien gliedern. Unter den *allgemeinen Anforderungen* werden zunächst diejenigen Anforderungen subsummiert, die zwar im Hinblick auf diese Arbeit auf die HMI-Entwicklung angewendet werden, sich aber auch auf andere Teilbereiche von Entwicklungsprozessen übertragen lassen. Mit den *Anforderungen im Kontext der Mechatronik* wird weiterhin die interdisziplinäre Zusammenarbeit während der Entwicklung adressiert. Die *Anforderungen im Kontext der Mensch-Maschine-Schnittstelle* werden ausschließlich im Hinblick auf HMIs angewendet und widmen sich der Problemstellung, dass Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau oft nicht an den Bedürfnissen der Nutzer ausgerichtet sind.

4.1.1 Allgemeine Anforderungen

Ganzheitliche Prozessbetrachtung: Wie im Stand der Wissenschaft und Technik aufgezeigt, beschränken sich aktuelle Entwicklungsansätze auf einzelne Aspekte der HMI-Entwicklung, wie z. B. die Modellbildung. Um Anwendern eine durchgängige Berücksichtigung von Mensch-Maschine-Schnittstellen zu ermöglichen, bedarf es allerdings einer ganzheitlichen Herangehensweise, mit der HMIs im Maschinen- und Anlagenbau von der anfänglichen Anforderungsanalyse bis zur finalen Implementierung berücksichtigt werden können. Dabei soll die zu erarbeitende Methodik nahtlos in den übergeordneten Entwicklungsprozess eines mechatronischen Systems eingebettet und nicht als nachgelagerter Prozessschritt betrachtet werden.

Konkrete Aktivitäten und Ergebnisse: Die zu erarbeitende Entwicklungsmethodik soll für Anwender einen einfach anzuwendenden Leitfaden bilden, wie Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau nutzerzentriert entwickelt werden können. Dazu soll sich der Ansatz in einzelne Aktivitäten und jeweils resultierende (Zwischen-)Ergebnisse unterteilen lassen und dabei auch die logische Reihenfolge dieser Aktivitäten definieren. Dadurch grenzt sich der Ansatz von agilen Vorgehensmodellen (z. B. Scrum) ab, in denen lediglich beschrieben wird, wie einzelne Aktivitäten auszuführen sind (z. B. Entwicklungszyklen). Aufgrund der zunehmenden Bedeutung dieser Herangehensweisen soll der Ansatz aber generell nicht im Widerspruch zu einer iterativen und inkrementellen Herangehensweise stehen. Vielmehr sollen die hier definierten Aktivitäten auch in einem agilen Entwicklungsprozess sinnvoll durchgeführt werden können.

Schlankes Vorgehen: In der Motivation dieser Arbeit wurde bereits herausgestellt, dass HMIs zukünftig eine größere Rolle im Maschinen- und Anlagenbau spielen müssen, was durch eine adäquate Berücksichtigung in den Entwicklungsprozessen realisiert werden kann. Allerdings sind die Entwicklungskapazitäten im Maschinen- und Anlagenbau eingeschränkt (SCHRÖDER 2003, S. 3), weswegen umfangreiche Vorgehensmodelle für eine nutzerzentrierte HMI-Entwicklung (vgl. Abschnitt 3.4) meist nicht angewendet werden können. Neben der frühzeitigen Integration von Mensch-Maschine-Schnittstellen in den mechatronischen Entwicklungsprozess soll die zu erarbeitende Methodik daher eine schlanke Herangehensweise fokussieren, die zwar die Umsetzung gebrauchstauglicher Mensch-Maschine-Schnittstellen unterstützt, aber auch für Unternehmen mit geringen Entwicklungskapazitäten einsetzbar ist.

Nutzung bestehender Ansätze: In Kapitel 3 wurde aufgezeigt, dass sowohl für mechatronische Systeme im Allgemeinen als auch Mensch-Maschine-Schnittstellen im Speziellen bereits vielfältige Entwicklungsansätze existieren. Der Handlungsbedarf für die vorliegende Arbeit liegt somit in der zielgerichteten Verknüpfung dieser bisher losgelöst voneinander betrachteten Themenfelder, wobei in beiden Bereichen auf bestehenden Konzepten aufgebaut werden soll. Dazu soll einerseits eine konkrete mechatronische Herangehensweise ausgewählt werden, die als Basis für die Integration der HMI-Entwicklung fungiert. Andererseits sollen auch die Vorarbeiten hinsichtlich Mensch-Maschine-Schnittstellen genutzt werden, wobei geeignete Aspekte des Stands der Technik zu selektieren und für die vorliegende Problemstellung zu adaptieren und weiterzuentwickeln sind. Neue Konzepte sollen sich auf Bereiche konzentrieren, in denen bisherige Arbeiten nicht den Spezifika des Maschinen- und Anlagenbaus gerecht werden (vgl. Abschnitt 4.1.3).

4.1.2 Spezielle Anforderungen im Kontext der Mechatronik

Interdisziplinäre Zusammenarbeit: Ein wesentliches Augenmerk von Entwicklungsansätzen für mechatronische Systeme liegt auf der Förderung der interdisziplinären Zusammenarbeit zwischen den beteiligten Gewerken (ISERMANN 2007, S. 33). Die angestrebte Integration von Mensch-Maschine-Schnittstellen in den Entwicklungskontext mechatronischer Systeme macht es erforderlich, diesen Aspekt auch bei der hier zu erarbeitenden Methodik zu berücksichtigen. Es soll explizit beschrieben werden, welche Wechselwirkungen und Abhängigkeiten zwischen der Mensch-Maschine-Schnittstelle und den weiteren Disziplinen über den Entwicklungsverlauf bestehen und wie eine zielgerichtete Zusammenarbeit unterstützt werden kann.

Zentrales Systemmodell: Insbesondere in der frühen Entwicklungsphase kann die Zusammenarbeit zwischen den einzelnen Gewerken vom Einsatz eines zentralen interdisziplinären Systemmodells und somit einer modellbasierten Vorgehensweise profitieren (HACKENBERG ET AL. 2015, S. 15). Vor diesem Hintergrund soll auch der hier zu erarbeitenden Entwicklungsmethodik ein zentrales Systemmodell zugrunde gelegt werden, das um die relevanten Aspekte einer Mensch-Maschine-Schnittstelle zu erweitern ist. Die Relevanz ergibt sich dabei aus der Fragestellung, welche Inhalte eines HMIs bereits in einer frühen Entwicklungsphase festzulegen und mit den weiteren Disziplinen abzustimmen sind. Dabei ist insbesondere darauf zu achten, bestehende Modellierungstechniken nach Möglichkeit zu nutzen und nur dort zu erweitern, wo die Spezifika von HMIs nicht mit bestehenden Mitteln sinnvoll abgebildet werden können.

Mechatronische Modularisierung: Eine wichtige Einflussgröße für den Maschinen- und Anlagenbau, um steigenden Ansprüchen hinsichtlich Kundenindividualität oder kürzer werdenden Entwicklungszyklen zu begegnen, ist die Modularisierung (NICKLAS 2003, S. 13). In Abschnitt 3.1 wurde bereits aufgezeigt, dass sich aktuelle Ansätze in diesem Bereich zunehmend einer ganzheitlichen mechatronischen Modularisierung widmen, in denen funktionale Module gebildet werden, die sich über alle beteiligten Disziplinen erstrecken. Diesem Gedanken folgend soll auch die hier zu erarbeitende Entwicklungsmethodik eine ganzheitliche Modularisierung unterstützen und eine Zuordnung einzelner Funktionen oder Bestandteile einer Mensch-Maschine-Schnittstelle zu mechatronischen Modulen ermöglichen. Dies soll insbesondere zu einer verbesserten Wiederverwendbarkeit einzelner Systembestandteile im Maschinen- und Anlagenbau beitragen und somit eine Reduktion von Entwicklungszeiten und -kosten ermöglichen.

4.1.3 Spezielle Anforderungen im Kontext der Mensch-Maschine-Schnittstelle

Nutzerzentrierte Herangehensweise: In Kapitel 1 wurde bereits aufgezeigt, dass die steigende Komplexität von Maschinen und Anlagen eine intuitive Interaktion mit diesen Systemen erforderlich macht. Um dies zu gewährleisten, soll die zu erarbeitende Methodik die Benutzer einer Maschine oder Anlage ins Zentrum der HMI-Entwicklung stellen, um die Mensch-Maschine-Schnittstelle bestmöglich an deren Aufgaben, Erwartungen und Bedürfnissen ausrichten zu können. Dazu kann der Ansatz insbesondere auf den Konzepten zur nutzerzentrierten HMI-Entwicklung aufbauen, die sich bislang vor allem in der Informatik etablieren konnten (vgl. Abschnitt 3.4). Ferner soll die Nutzerfokussierung auch eine Abgrenzung der Arbeit zu Ansätzen aus dem Bereich der kundenorientierten Produktentwicklung ermöglichen, die im Rahmen dieser Arbeit lediglich im Ausblick hinsichtlich einer möglichen Kombination mit der hier erarbeiteten Methodik betrachtet werden.

Simulationsbasiertes Testen: Innerhalb der Ansätze zur nutzerzentrierten HMI-Entwicklung nimmt der Test von Mensch-Maschine-Schnittstellen durch Benutzer eine zentrale Rolle ein (OPPERMANN & REITERER 1992, S. 285). Vor diesem Hintergrund soll auch an die hier zu erarbeitende Entwicklungsmethodik explizit die Anforderung gestellt werden, HMIs frühzeitig und kontinuierlich im Entwicklungsprozess mit Nutzern erproben zu können. Dies soll insbesondere durch eine Simulation sichergestellt werden, in der bereits frühzeitig ablauffähige HMI-Modelle im Kontext des interdisziplinären Systemmodells evaluiert werden können.

HMI-Spezifika im Maschinen- und Anlagenbau: Neben der generellen Fokussierung auf die Benutzer soll der Ansatz den Spezifika gerecht werden, die Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau mit sich bringen. Unter diesen ist zunächst die häufig anzutreffende Aufteilung der gesamten Benutzerinteraktion auf verschiedene Hardware-Komponenten zu nennen, wie z. B. mehrere graphische HMI-Panels bei großen Anlagen. Aber auch die räumliche Anordnung dieser Komponenten im Kontext der gesamten Maschine bzw. Anlage ist für das Nutzungserlebnis von großer Bedeutung (GERKE 2015, S. 14 ff.). Darüber hinaus sind Fragestellungen der Systemsicherheit, der Langlebigkeit oder Robustheit explizit bei der Entwicklung zu beachten. Schließlich ist zu berücksichtigen, dass das Aufgabenspektrum der Benutzer trotz der Produktvielfalt im Maschinen- und Anlagenbau meist auf standardisierte Abläufe (bspw. zur Fehlerbehandlung oder Parametereinstellung) zurückzuführen ist, was insbesondere die Gestaltung von Baukastensystemen vereinfachen kann.

Innovative Interaktionstechnologien: Neben der Nutzerzentrierung können auch innovative Interaktionstechnologien (vgl. Abschnitt 2.2) zur Umsetzung von gebrauchstauglichen Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau beitragen (PEISSNER & HIPPEL 2013, S. 36 f.). Die hier zu erarbeitende Methodik soll daher aktuelle technologische Möglichkeiten und zukünftige Trends berücksichtigen und dabei die Auswahl geeigneter Technologien für einen konkreten Anwendungsfall unterstützen. Weiterhin muss auch der Entwicklungsprozess so gestaltet sein, dass dieser die Umsetzung unterschiedlicher Technologien unterstützt und sich nicht allein auf graphische Mensch-Maschine-Schnittstellen beschränkt.

4.2 Auswahl einer mechatronischen Entwicklungsmethodik als Forschungsgrundlage

Wie bereits bei der Identifikation des Handlungsbedarfs aufgezeigt und im ersten Teil der Anforderungsanalyse konkretisiert, soll die zu erarbeitende Entwicklungsmethodik für Mensch-Maschine-Schnittstellen auf bestehenden Ansätzen aufbauen und sich in eine übergeordnete mechatronische Herangehensweise einbetten. Die Diversität der in Abschnitt 3.3 vorgestellten Ansätze macht es dabei erforderlich, eine konkrete Methodik auszuwählen und die HMI-Entwicklung in diesen Ansatz zu integrieren.⁸ Diese Auswahl soll dazu beitragen, der Anforderung nach einer interdisziplinären Zusammenarbeit über den Entwicklungsverlauf bestmöglich gerecht zu werden. Gleichzeitig soll bei der Erarbeitung der Methodik aber darauf geachtet werden, einzelne Bausteine nach Möglichkeit so zu gestalten, dass diese auch bei Unternehmen eingesetzt werden können, die ihre Entwicklungsprozesse nicht an dem konkreten, hier ausgewählten Ansatz ausrichten möchten.

Die Basis zur Auswahl eines mechatronischen Entwicklungsansatzes bilden die zuvor definierten Anforderungen. Diese wurden zum Teil direkt als Bewertungskriterien für die zu analysierenden Ansätze übernommen, teilweise wurden aber auch Kriterien formuliert, die aus einzelnen Anforderungen abgeleitet wurden. Als Beispiel sei das Bewertungskriterium *Simulationsunterstützung* genannt, um ein simulationsbasiertes Testen von Mensch-Maschine-Schnittstellen zu ermöglichen. Weiterhin ist zu erwähnen, dass die formulierte Anforderung nach der Verwendung eines zentralen Systemmodells nicht als Bewertungskriterium auftritt, da diese bereits im Vorfeld der Analyse zur Eingrenzung des Betrachtungsraums auf

⁸ Der umgekehrte Weg zur Auswahl eines HMI-Entwicklungsansatzes (vgl. Abschnitt 3.4) wurde ausgeschlossen, da diese Ansätze in sich geschlossen sind und keine Integration mehrerer Disziplinen vorsehen.

modellbasierte mechatronische Entwicklungsansätze genutzt wurde. Insgesamt ergeben sich neun Bewertungskriterien, anhand derer die in Abschnitt 3.3.2 detailliert vorgestellten Ansätze bewertet wurden. Die Bewertung erfolgte dabei qualitativ anhand der über die einzelnen Ansätze verfügbaren Quellen und Zusatzmaterialien (z. B. Entwicklungswerkzeuge). Tabelle 4-1 zeigt die Gegenüberstellung der formulierten Bewertungskriterien und den analysierten modellbasierten Entwicklungsansätzen sowie die jeweils resultierende qualitative Bewertung.

Tabelle 4-1: Bewertung der modellbasierten mechatronischen Entwicklungsansätze

		Analysierte Ansätze				
		EIGNER (2013)	GAUSEMEIER ET AL. (2014)	VDMA (2010)	HENSEL (2011)	HACKENBERG ET AL. (2015)
Bewertungskriterien	Ganzheitliche Prozessbetrachtung	●	●	◐	◑	●
	Konkrete Aktivitäten und Ergebnisse	○	●	◑	◑	◑
	Schlankes Vorgehen	◑	◑	◑	◑	◑
	Interdisziplinäre Zusammenarbeit	◑	●	◑	◑	●
	Mechatronische Modularisierung	○	◑	◑	◑	●
	Anforderungsmodellierung und -management	◑	●	○	◑	◑
	Fehlermodellierung und -management	○	◑	○	●	●
	Simulationsunterstützung	◑	◑	●	●	●
	Berücksichtigung von HMIs	○	○	○	◑	○

Anhand dieser Bewertung wird deutlich, dass der IMoMeSA-Entwicklungsansatz von HACKENBERG ET AL. (2015) eine geeignete Forschungsgrundlage für die vorliegende Arbeit bildet. Im Unterschied zu HENSEL (2011) werden HMIs hier zwar nicht berücksichtigt, was einen höheren Aufwand für die Integration vermuten lässt, allerdings wurden vom Autor der vorliegenden Arbeit bereits bei der Ausarbeitung des IMoMeSA-Ansatzes wichtige Grundlagen geschaffen (bspw. im Anforderungsmanagement), um die Integration von Mensch-Maschine-Schnittstellen zu vereinfachen. Aufgrund der besonderen Bedeutung des IMoMeSA-Ansatzes für diese Arbeit sind dessen zentrale Inhalte im Anhang A1 beschrieben, die zum Verständnis der im folgenden Kapitel eingeführten Methodik zur modellbasierten Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau einen wichtigen Beitrag leisten können.

5 Modellbasierte Entwicklungsmethodik für Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau

Basierend auf den formulierten Anforderungen und der ausgewählten Forschungsgrundlage widmet sich dieses Kapitel der Darstellung der erarbeiteten Methodik zur modellbasierten Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau. In Abschnitt 5.1 wird dazu zunächst ein Gesamtüberblick über die Entwicklungsmethodik gegeben. Dieser zeigt auf, aus welchen Lösungsbausteinen bzw. Aktivitäten die Methodik zusammengesetzt ist und wie sich diese innerhalb der gewählten Forschungsgrundlage einordnen. Die nachfolgenden Abschnitte 5.2 bis 5.5 widmen sich anschließend der detaillierten Vorstellung und Erklärung dieser Aktivitäten. Dazu haben die einzelnen Abschnitte einen jeweils vergleichbaren Aufbau, in dem zunächst die technischen Lösungsbausteine eingeführt werden, bevor der methodische Prozessablauf zur Anwendung dieser Bausteine adressiert wird. In diesem Kontext werden auch die Wechselwirkungen mit dem übergeordneten mechatronischen Entwicklungsansatz dargelegt, um so explizit die interdisziplinäre Zusammenarbeit zu stärken. Abgerundet wird das Kapitel durch eine kurze Zusammenfassung in Abschnitt 5.6.

5.1 Gesamtüberblick über die Entwicklungsmethodik

Wie bei der Vorstellung des IMoMeSA-Ansatzes in Abschnitt 3.3.2 beschrieben (vgl. auch Anhang A1), gliedert sich die Entwicklungsmethodik von HACKENBERG ET AL. (2015) in die drei Phasen der Konzeption, der Verfeinerung und des physikalischen Aufbaus. Diese übergeordneten Phasen können auch nach der Erweiterung des Ansatzes um Mensch-Maschine-Schnittstellen beibehalten werden. In die Phasen der *Konzeption* und der *Verfeinerung* wurden für die HMI-Erweiterung vier neue Aktivitäten eingebettet, die in Abbildung 5-1 dargestellt und auf einer qualitativen Ebene zeitlich im Entwicklungsverlauf eingeordnet sind. Im Bereich des *physikalischen Aufbaus* kann für die HMI-Entwicklung auf die bereits im ursprünglichen IMoMeSA-Ansatz beschriebenen Aktivitäten zurückgegriffen werden, weswegen diese Phase nachfolgend nicht näher betrachtet wird. Dies lässt sich auch durch den hohen Anteil an Software in der HMI-Entwicklung begründen, die gemäß dem hier erarbeiteten Ansatz bereits zum Ende der Verfeinerung vollständig vorliegt. Die Aktivitäten während des physikalischen Aufbaus beschränken sich aus HMI-Sicht somit auf die Aktivitäten des Hardwareaufbaus und des Systemtests, die bereits im ursprünglichen IMoMeSA-Ansatz berücksichtigt sind.

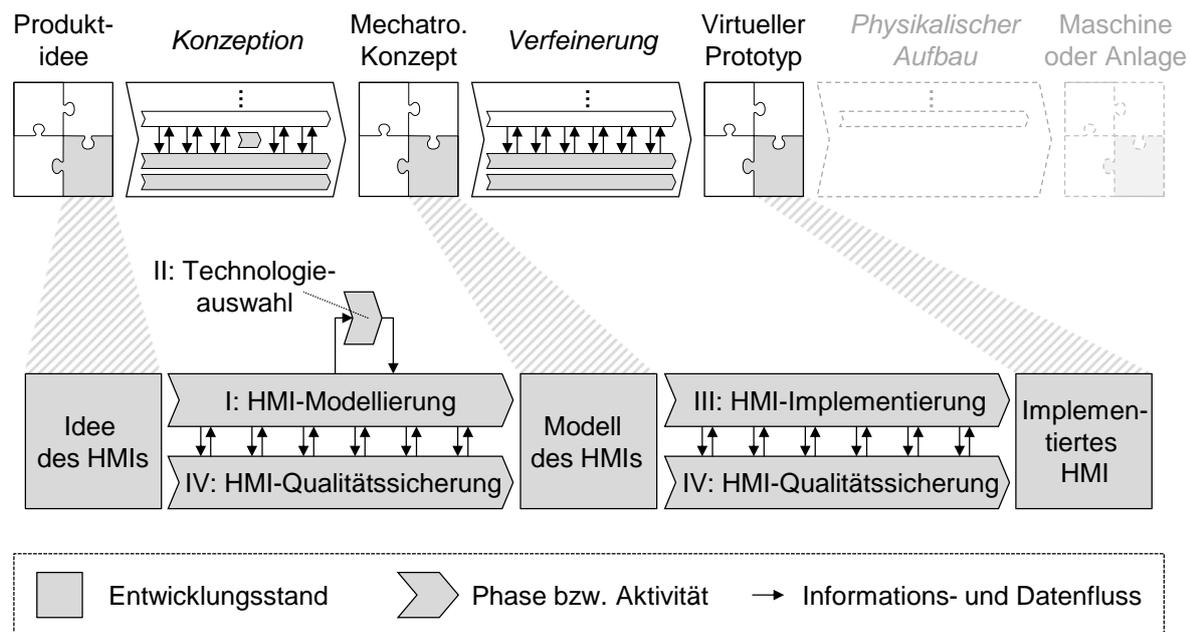


Abbildung 5-1: Gesamtüberblick der erarbeiteten Entwicklungsmethodik sowie der HMI-spezifischen Erweiterungen (grau hervorgehoben)

In der Phase der Konzeption ist die *HMI-Modellierung* (I) die zentrale Aktivität, um eine Idee des HMIs (als Teil der gesamten Produktidee) in ein Modell des HMIs (als Teil des mechatronischen Konzepts) zu überführen. Zur Durchführung dieser Aktivität wird als Lösungsbaustein eine Modellierungstechnik benötigt, die auf dem ursprünglichen IMoMeSA-Metamodell (vgl. Anhang A1) aufbaut, dieses allerdings um HMI-spezifische Inhalte erweitert, um alle für die Konzeption relevanten Bestandteile einer Mensch-Maschine-Schnittstelle abbilden zu können. Da die HMI-Modellierung von einer starken Wechselwirkung mit der Modellierung des gesamten mechatronischen Systems geprägt ist, wird weiterhin ein Prozessablauf benötigt, welcher den Einsatz der Modellierungstechnik während der Konzeption beschreibt. Dabei werden insbesondere vorhandene Abhängigkeiten in der Systemmodellierung identifiziert, die eine Festlegung der generellen Abfolge der einzelnen Modellierungsschritte ermöglichen.

Während der Modellierung einer Mensch-Maschine-Schnittstelle ist es zu einem konkreten Zeitpunkt erforderlich, eine *Technologieauswahl* (II) zu treffen. Die mittlerweile vorhandene Vielfalt an Interaktionstechnologien (vgl. Abschnitt 2.2) sowie die Menge an unterschiedlichen Einflussfaktoren auf deren Auswahl machen es erforderlich, diese Aufgabe als eigenständige, in sich geschlossene Aktivität im Entwicklungsprozess zu berücksichtigen. Als zentralen Lösungsbaustein liefert die vorliegende Arbeit dazu eine Gegenüberstellung von wichtigen Auswahlfaktoren

und Merkmalen von Interaktionstechnologien in Form einer Einflussmatrix. Diese ermöglicht es Entwicklern, auf Basis ihres vorliegenden Anwendungsfalls wichtige Merkmale zu identifizieren, die eine Interaktionstechnologie für einen zielführenden Umgang mit dem HMI aufweisen sollte. Da neben diesen Merkmalen weitere Rahmenbedingungen, wie bspw. eine Investitionsrechnung, in die Entscheidung einzubeziehen sind, wird weiterhin ein Prozessablauf definiert, der beschreibt, wie die Einflussmatrix bei der Technologieauswahl einzusetzen ist. Dabei wird auch darauf eingegangen, wie die bis hierhin gesammelten Erkenntnisse aus der HMI-Modellierung verwendet werden können und wie das Ergebnis der Technologieauswahl in die weiteren Entwicklungsaktivitäten einfließt. Generell ist die Aktivität der Technologieauswahl jedoch weitestgehend losgelöst von der übergeordneten Entwicklungsmethodik einsetzbar.

Basierend auf dem Modell einer Mensch-Maschine-Schnittstelle als Teil des mechanischen Konzepts erfolgt innerhalb der Verfeinerung die *HMI-Implementierung* (III). Dazu ist zunächst eine Transformation des HMI-Modells in eine ausgewählte Entwicklungsumgebung notwendig, um Mehraufwände in der Entwicklung zu vermeiden und eine Konsistenz zwischen Modell und zugehörigem HMI-Programm sicherzustellen. Die erarbeiteten Methoden für diese Modelltransformation bilden den Lösungsbaustein dieser Aktivität. Auf Basis der initialen HMI-Generierung kann die Implementierung der Mensch-Maschine-Schnittstelle generell parallel zur Verfeinerung der weiteren Disziplinen durchgeführt werden, wobei das Systemmodell weiterhin zur Konsistenzsicherung zum Einsatz kommt. Für diese Aufgabe wird ein entsprechender Prozessablauf definiert, der es vorsieht, alle interdisziplinär relevanten Entwicklungsschritte zunächst im Systemmodell abzubilden, um automatisiert die Änderungen an der Mensch-Maschine-Schnittstelle an die weiteren beteiligten Disziplinen zu kommunizieren. Über erneute Modelltransformationen wird sichergestellt, dass die Konsistenz zwischen HMI-Modell und HMI-Programm im Entwicklungsverlauf erhalten bleibt.

Sowohl während der Implementierung als auch bereits bei der Modellierung ist es notwendig, die Funktionalität und Gebrauchstauglichkeit einer Mensch-Maschine-Schnittstelle abzusichern. Zu diesem Zweck sieht der hier entwickelte Ansatz die Aktivität *HMI-Qualitätssicherung* (IV) vor, wofür im Rahmen dieser Arbeit zwei unterschiedliche Simulationstechniken zum Einsatz kommen. In die mit der Simulation ausgeführten Tests sind auch die späteren Benutzer des Systems eingebunden, um sicherzustellen, dass ein HMI an deren Erwartungen und Bedürfnissen ausgerichtet ist. Wie diese Einbindung erfolgt und wie die Simulation generell während des gesamten Entwicklungsverlaufs möglichst zielgerichtet eingesetzt

werden kann, wird in einem zugehörigen Prozessablauf festgelegt. Dabei werden auch die Unterschiede bezüglich der Verwendung der Simulation herausgearbeitet, die zwischen den Phasen der Konzeption und der Verfeinerung bestehen.

5.2 Aktivität I: HMI-Modellierung

Wie bereits im letzten Abschnitt dargelegt, basiert die Modellierung von Mensch-Maschine-Schnittstellen auf der Modellierungstechnik für mechatronische Systeme von HACKENBERG ET AL. (2015). Zur Erweiterung dieser Technik konnten die verfügbaren Ansätze zur HMI-Modellierung aus dem Stand der Wissenschaft und Technik (vgl. Abschnitte 2.4.2 und 3.4.2) genutzt werden, wobei die jeweils relevanten Aspekte dieser Ansätze gemäß der vorliegenden Zielsetzung adaptiert und in die Modellierungstechnik integriert wurden. In Abschnitt 5.2.1 werden die resultierenden Elemente der Modellierungstechnik spezifiziert und erläutert. Darauf aufbauend widmet sich Abschnitt 5.2.2 der Beschreibung des Prozessablaufs, um die Modellierungstechnik während der Konzeption von mechatronischen Systemen zielführend einzusetzen. Dabei werden auch die Wechselwirkungen zwischen HMI-Entwicklung und dem mechatronischen Systementwurf herausgearbeitet.

5.2.1 Erweiterte Systemmodellierungstechnik zur Abbildung von HMIs

Die von HACKENBERG ET AL. (2015) erarbeitete Modellierungstechnik umfasst acht unterschiedliche Elemente, um mechatronische Systeme abzubilden. Gemäß der formulierten Anforderung, bestehende Techniken nach Möglichkeit für die HMI-Modellierung zu nutzen, wurden diese Elemente zunächst dahingehend analysiert, inwiefern diese zur Abbildung von Mensch-Maschine-Schnittstellen verwendet werden können. Dabei zeigte sich, dass das Element *Verhalten* ohne Anpassungen für die Modellierung der HMI-Logik genutzt werden kann. Auch *Anforderungen*, *Ports*, *Szenarien* und *Komponenten* können prinzipiell zum Einsatz kommen, müssen allerdings sowohl semantisch als auch teilweise syntaktisch adaptiert werden. Darüber hinaus sind auch neue Modellierungselemente (*Benutzerrollen* und *Interaktionselemente*) notwendig, um alle für die Konzeption relevanten Aspekte einer Mensch-Maschine-Schnittstelle abbilden zu können. Abbildung 5-2 gibt einen Überblick über die erweiterte Modellierungstechnik, in welcher die für die HMI-Entwicklung einzusetzenden Elemente grau hervorgehoben sind. Grundsätzlich lässt sich für die Anwendung dieser Elemente eine generelle Reihenfolge und Einordnung innerhalb der bestehenden Modellierungselemente bilden, wie auch in Abbildung 5-2 deutlich wird. Diese Reihenfolge wird insbesondere bei der

Ausarbeitung des Prozessablaufs in Abschnitt 5.2.2 erneut aufgegriffen, um den Einsatz der HMI-Modellierungselemente im Verlauf der Konzeption zu definieren. Nachfolgend werden die einzelnen Elemente im Detail vorgestellt. Dabei richten sich die Erläuterungen an potenzielle Anwender aus dem Maschinen- und Anlagenbau. Eine detaillierte Spezifikation auf Metamodell-Ebene für die Entwickler von Softwarewerkzeugen findet sich stattdessen im Anhang A2 dieser Arbeit.

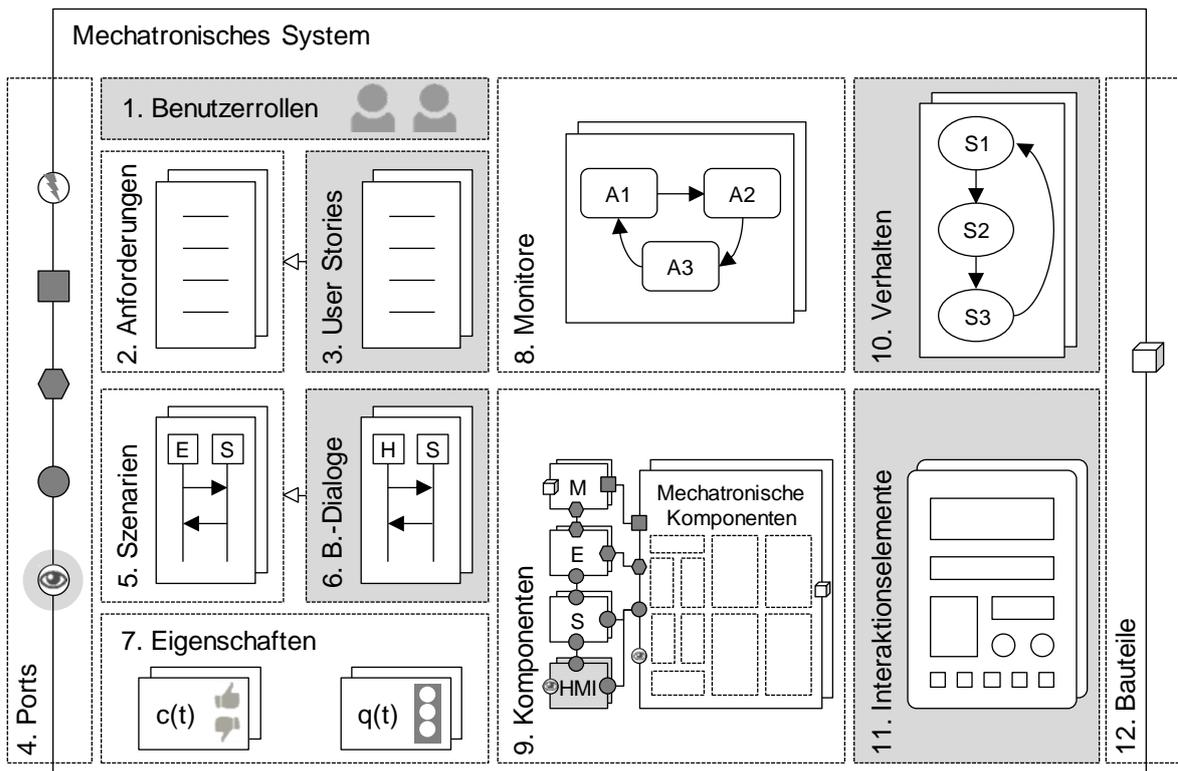


Abbildung 5-2: Erweiterte Modellierungstechnik zur Abbildung von HMIs in der Konzeption mechatronischer Systeme; HMI-Elemente grau hervorgehoben

Benutzerrollen

Das erste Modellierungselement zur Abbildung von HMIs adressiert die Festlegung der einzelnen Benutzer, die mit dem mechatronischen System interagieren. Die Modellierung erfolgt im Rahmen dieser Arbeit auf der Ebene abstrakter Benutzerrollen, wie z. B. Schichtleiter oder Administrator, die im Maschinen- und Anlagenbau typischerweise jeweils eigene Aufgaben im Umgang mit dem mechatronischen System besitzen. Durch Kombination mit weiteren Modellierungselementen, wie bspw. User Stories, wird eine Möglichkeit geschaffen, Benutzern ihre jeweiligen Aufgaben und Verantwortlichkeiten im Umgang mit dem HMI einfach zuzuordnen.

Um diese Ziele zu erreichen, setzt sich jede modellierte Benutzerrolle aus einem eindeutigen Namen und einer optionalen textuellen Beschreibung des jeweiligen Rollenverständnisses zusammen. Weiterhin besteht die Möglichkeit, konkrete Benutzerrollen aus übergeordneten Benutzerrollen abzuleiten, um dadurch ein hierarchisches Schema für deren Strukturierung aufzubauen. Dies vereinfacht auch die spätere Zuordnung von Aufgaben und Verantwortlichkeiten, die für mehrere Benutzerrollen relevant sind. Dabei können Aufgaben auch übergeordneten Rollen zugeordnet werden, wodurch sie im Sinne einer Vererbung auch für abgeleitete Rollen gültig sind. Generell ist die Abbildung von Benutzern mit dem vorgeschlagenen Schema aus einem eindeutigen Namen, einer textuellen Beschreibung und einer hierarchischen Strukturierung bewusst auf das Wesentliche konzentriert, um die Modellierungstechnik schlank zu halten. Weiterführende Konzepte der Benutzermodellierung, wie bspw. Personas (vgl. GOTHELF & SEIDEN 2013), sind nicht Teil dieses Ansatzes, können aber bei Bedarf mit geringem Aufwand in die Technik integriert werden.

Hinsichtlich ihrer Einbettung in die gesamte Modellierungstechnik sind Benutzerrollen als Bestandteile von Komponenten definiert. In den meisten Fällen werden Benutzerrollen zwar direkt auf Systemebene modelliert, jedoch berücksichtigt der gewählte Ansatz auch den Fall, dass spezielle Benutzerrollen nur mit Teilen des gesamten Systems bzw. einzelnen Komponenten interagieren sollen oder dürfen. Auch das im letzten Absatz definierte Vererbungsprinzip für Benutzerrollen lässt sich auf die Zuordnung zu einzelnen Komponenten anwenden. So kann bspw. auf Systemebene eine generische Benutzerrolle „Systembediener“ mit übergreifenden Aufgaben und Verantwortlichkeiten definiert werden, aus der auf Komponentenebene konkretere Benutzerrollen, wie z. B. „Bediener Modul A“, mit speziellen Aufgaben und Rechten abgeleitet werden.

User Stories

Um in einer frühen Entwicklungsphase die Aufgaben einzelner Benutzerrollen im Umgang mit einem mechatronischen System schnell und einfach modellieren zu können, wird der Einsatz des Elements „User Stories“ vorgeschlagen. Diese ermöglichen eine informelle, aber strukturierte Abbildung von Benutzeraufgaben, die den funktionalen Anforderungen an eine Mensch-Maschine-Schnittstelle entsprechen. Für nicht-funktionale Aspekte, wie ein zu verwendendes Farbschema, kann hingegen auf das im ursprünglichen Meta-Modell vorhandene Element „Anforderungen“ zurückgegriffen werden.

User Stories haben ihren Ursprung in der agilen Softwareentwicklung und werden hier für die Abbildung und Strukturierung von Kundenanforderungen verwendet. Der von COHN (2004) vorgeschlagene standardisierte Aufbau einer User Story wird im Rahmen dieser Arbeit aufgegriffen und ermöglicht eine Verknüpfung der jeweiligen Aufgabe mit einer Benutzerrolle und dem angestrebten Nutzen. Insbesondere der Nutzen ist aus Sicht der mechatronischen Zusammenarbeit besonders wichtig, um den Bezug einer Aufgabe zu dem produktionstechnischen System abbilden zu können. Abbildung 5-3 zeigt den resultierenden Aufbau einer User Story und veranschaulicht diesen anhand eines einfachen Beispiels. Während in den Feldern „Ziel“ und „produktionstechnischer Nutzen“ beliebige textuelle Eingaben möglich sind, sieht das zugehörige Meta-Modell (vgl. Anhang A2) ein Referenzieren von modellierten Benutzerrollen vor. Weiterhin wird im Meta-Modell eine Verbindung von User Stories mit beliebigen Modellierungselementen vorgeschlagen. Neben einer Verknüpfung von User Stories untereinander, um bspw. auf Abhängigkeiten hinzuweisen, kann somit auch abgebildet werden, welche HMI-Komponenten oder Interaktionselemente zu der Realisierung einer User Story beitragen.

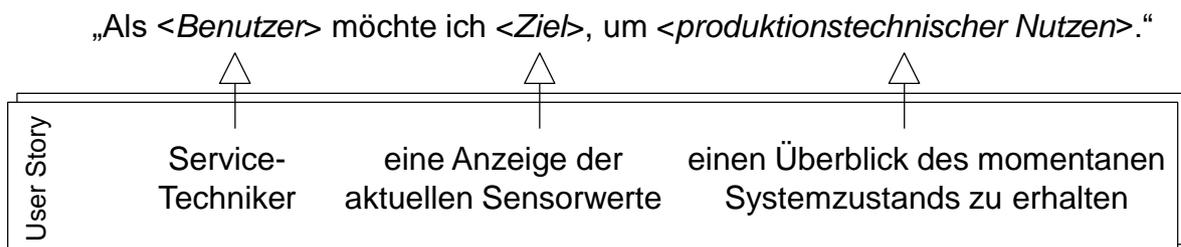


Abbildung 5-3: Aufbau und Beispiel einer User Story

User Stories können allgemein als eine Konkretisierung der Anforderungen aus dem ursprünglichen Meta-Modell verstanden werden. So wird bspw. die Möglichkeit der Verknüpfung mit beliebigen Modellierungselementen übernommen, während die freie textuelle Eingabe bei Anforderungen durch den standardisierten Aufbau einer User Story ersetzt wird. Wie Anforderungen können auch User Stories innerhalb von beliebigen Komponenten modelliert werden. Typischerweise werden dabei zunächst allgemeine User Stories (vgl. Abbildung 5-3) auf Systemebene spezifiziert, die dann durch detailliertere User Stories innerhalb einzelner Teilkomponenten konkretisiert werden. Der genaue Modellierungsablauf für User Stories wird in Abschnitt 5.2.2 noch detailliert erläutert.

HMI-Ports

Eine erste Möglichkeit für die Formalisierung von Benutzeraufgaben, die in den User Stories spezifiziert sind, bietet das Modellierungselement „HMI-Ports“. Dieses bildet die vorhandenen Interaktionsmöglichkeiten eines Benutzers mit einem mechatronischen System bzw. seinen Komponenten aus einer funktionalen Perspektive ab, ohne sich dabei bereits auf mögliche technologische Lösungen einzuschränken. Da sich die Aufgaben der Benutzer meist nicht auf die Interaktion mit der Mensch-Maschine-Schnittstelle selbst beschränken, wie bspw. Parameter-einstellungen, sondern auch einen physischen Eingriff in das mechatronische System erfordern, wie bspw. eine Materialeingabe, dürfen sich auch HMI-Ports nicht auf eine reine Ein- und Ausgabe von Daten beschränken.

Daher sind HMI-Ports im vorliegenden Ansatz nicht als eigenständiger Porttyp definiert, sondern als eine spezielle Ausprägung der bereits vorhandenen Porttypen für Material, Energie und Daten. Dafür verfügt jeder Port über ein Flag, über das er als HMI-Port gekennzeichnet und im weiteren Modellierungsverlauf als solcher genutzt werden kann. Mittels HMI-Materialports werden dabei im Wesentlichen Übergabepositionen definiert, an denen Benutzer bspw. Bauteile in eine Maschine einlegen können. Über HMI-Energieports werden demgegenüber alle weiteren Interaktionsmöglichkeiten spezifiziert, die keiner speziellen Datenverarbeitung bedürfen. Sie repräsentieren in der Regel manuelle Einstellungen an Sensoren und Aktoren, wie bspw. zum Öffnen von Ventilen. HMI-Datenports befassen sich schließlich mit den Interaktionsmöglichkeiten, die über die zu realisierende Mensch-Maschine-Schnittstelle bereitgestellt werden. Das Spektrum möglicher HMI-Datenports reicht von einfachen binären Signalen, wie zum Start und Stopp von Bearbeitungsvorgängen, bis zu komplexen graphischen Anzeigen. Unabhängig vom konkreten Typ können alle HMI-Ports mit Benutzerrollen assoziiert werden, um auf Modellebene grundsätzliche Verantwortlichkeiten zu spezifizieren, die in der späteren HMI-Implementierung in konkrete Benutzerrechte überführt werden.

Als Spezialform allgemeiner Ports haben HMI-Ports analoge Eigenschaften und werden somit an der Schnittstelle von beliebigen Komponenten modelliert. Dabei besteht grundsätzlich die Möglichkeit, HMI-Ports (außer Material) mit Kanälen (vgl. HACKENBERG ET AL. 2015, S. 29) zwischen Komponenten und über Hierarchieebenen hinweg zu verbinden. Dies ist aber nicht zwingend notwendig, da Benutzerdialoge und Interaktionselemente (vgl. nachfolgende Abschnitte) auch dezentral auf HMI-Ports zugreifen können, wodurch eine Bereitstellung aller HMI-Ports auf Systemebene nicht erforderlich ist.

Benutzerdialoge

Gemeinsam mit den HMI-Ports werden Benutzerdialoge verwendet, um eine spezifizierte User Story bzw. eine Benutzeraufgabe formal abzubilden. Dabei wird festgelegt, welche Eingaben ein Benutzer bei der Durchführung einer Aufgabe tätigt und welche Ausgaben vom System erwartet werden. Zudem werden die einzelnen Interaktionen in eine qualitative zeitliche Reihenfolge gebracht. Ein Benutzerdialog kann somit als ein konkreter Ausschnitt des Nutzerverhaltens interpretiert werden. Innerhalb eines solchen Dialogs können alle definierten HMI-Ports (Material, Energie, Daten) auftreten, um auch Aufgaben abbilden zu können, die über eine reine Interaktion an der Mensch-Maschine-Schnittstelle hinausgehen.

Sowohl syntaktisch als auch semantisch sind Benutzerdialoge als Spezialform der allgemeinen Szenarien zu betrachten. An die Stelle der allgemeinen Umgebung treten konkrete Benutzerrollen, die Struktur beider Modellierungselemente ist allerdings analog. So kommen auch bei Benutzerdialogen Nachrichten-Reihenfolge-Diagramme zum Einsatz, um den Ablauf einer Aufgabe anhand der modellierten HMI-Ports festzulegen. Als Entitäten treten dabei zuvor modellierte Benutzerrollen und beliebige Komponenten auf. Während bei einem Szenario allerdings die Anzahl der Entitäten auf eine betrachtete Komponente und deren Umgebung beschränkt ist, ermöglicht ein Benutzerdialog auch die gleichzeitige Modellierung mehrerer Benutzerrollen und unterschiedlicher Komponenten. Damit soll einerseits die Abbildung von Aufgaben ermöglicht werden, in deren Durchführung mehrere Benutzer involviert sind, andererseits wird der Tatsache Rechnung getragen, dass ein Benutzerdialog auch auf HMI-Ports verschiedener Komponenten zugreifen kann. Abbildung 5-4 zeigt exemplarisch einen Benutzerdialog, der die Aufgabe zum Einlegen und Entnehmen eines Bauteils in bzw. aus einer Werkzeugmaschine sowie zum Start der Bearbeitung nach voriger Programmauswahl formal erfasst.

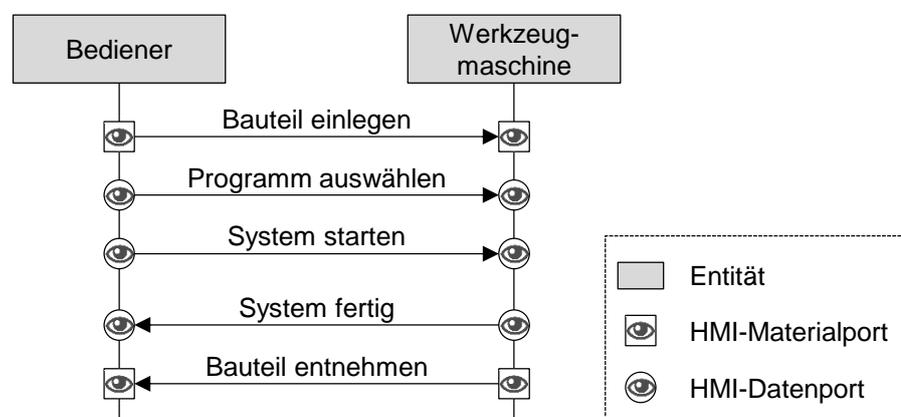


Abbildung 5-4: Exemplarischer Benutzerdialog

Wie Szenarien sind auch Benutzerdialoge als Bestandteile beliebiger Komponenten definiert. Dies ermöglicht eine Verteilung von Benutzerdialogen über die gesamte Komponentenhierarchie, um so komplexe Aufgaben, wie bspw. eine Fehlersuche und -behandlung, auf Systemebene abzubilden, während einfache Aufgaben, wie bspw. das Auslesen eines Sensorwertes, unmittelbar einzelnen Teilkomponenten zugeordnet werden können. Ein modellierter Benutzerdialog adressiert dabei primär die Komponente, in welcher er modelliert wird, gleichzeitig können aber auch alle abgeleiteten Teilkomponenten als weitere Entitäten hinzugefügt werden. Dies ist gerade bei komplexen Aufgaben erforderlich, bei denen Benutzer oft mit einer Vielzahl an HMI-Ports unterschiedlicher Komponenten interagieren müssen.

HMI-Komponenten

Während sich die bisherigen Modellierungselemente primär für die Abbildung von Anforderungen an eine Mensch-Maschine-Schnittstelle eignen, adressieren HMI-Komponenten die Abbildung der notwendigen Funktionalität, um spezifizierte Benutzerdialoge realisieren zu können. HMI-Komponenten fungieren in erster Linie als Bindeglied zwischen den HMI-Ports einer mechatronischen Komponente und ihren relevanten Teilkomponenten, wie bspw. Sensoren oder Aktoren. Sie verantworten dabei im Wesentlichen die Verarbeitung und Weiterleitung von Benutzereingaben an HMI-Eingabepports sowie in umgekehrter Richtung die korrekte Anzeige des aktuellen Systemzustands an HMI-Ausgabepports.

Um dieses Ziel zu erreichen, sind HMI-Komponenten als Bestandteile beliebiger mechatronischer Komponenten definiert. Jede HMI-Komponente besitzt eigene Ports, die einerseits mit den HMI-Ports der übergeordneten mechatronischen Komponente und andererseits mit den Ports der relevanten Teilkomponenten (Sensoren, Aktoren, etc.) über Kanäle verbunden werden. Meist wird allerdings nur ein konkreter Ausschnitt der übergeordneten HMI-Ports mit einer HMI-Komponente verknüpft, der aus Sicht der Benutzeraufgaben logisch zusammenhängt. Daher werden in der Regel mehrere HMI-Komponenten innerhalb einer mechatronischen Komponente modelliert, um die insgesamt geforderte Funktionalität abbilden zu können. Diese Herangehensweise vereinfacht im Sinne des „Separation of Concerns“-Konzepts (HÜRSCH & LOPES 1995) den Entwurf der notwendigen Logik, die im gewählten Ansatz mit dem Modellierungselement „Verhalten“ (siehe nächster Abschnitt) spezifiziert wird. Abbildung 5-5 verdeutlicht die resultierende Systemarchitektur anhand einer beispielhaften mechatronischen Komponente.

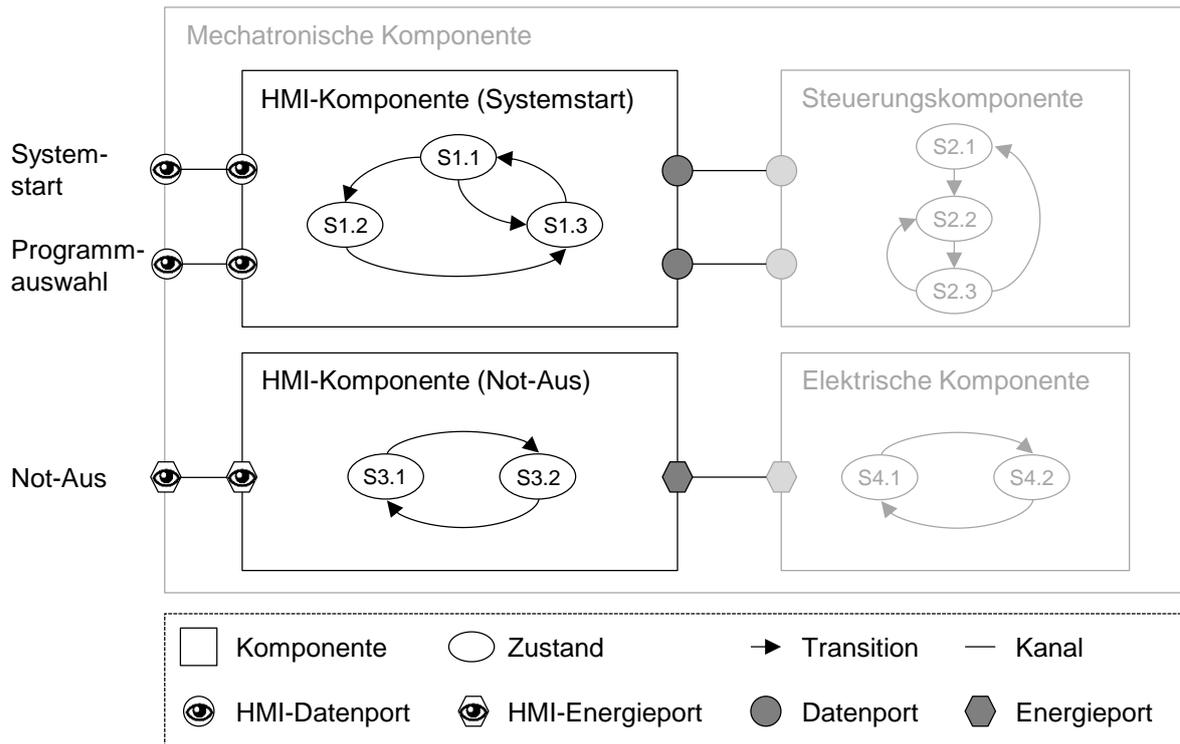


Abbildung 5-5: Einsatz von HMI-Komponenten mit integriertem Verhalten innerhalb einer beispielhaften mechatronischen Komponente

Verhalten

Um zwischen den an einer HMI-Komponente anliegenden Ports eine Kommunikation zu ermöglichen, bedarf es einer entsprechenden Logik. Diese wird im entwickelten Ansatz mit dem Modellierungselement „Verhalten“ abgebildet, das bereits im ursprünglichen IMoMeSA-Metamodell (vgl. Anhang A1) für die Ausgestaltung von beliebigen Komponenten vorgesehen ist.

Dabei kommen endliche, diskrete Zustandsautomaten zum Einsatz, die sich aus Zuständen und Transitionen zusammensetzen. Über mathematische Ausdrücke werden in Abhängigkeit der aktuellen Werte an den Eingabeports die Bedingungen definiert, die zu einem Zustandswechsel führen. In einzelnen Zuständen oder bei Zustandsübergängen können Aktionen modelliert werden, die ebenfalls mittels mathematischer Ausdrücke definieren, mit welchen Werten die Ausgabeports einer HMI-Komponente aktuell belegt werden sollen. Zudem können interne Variablen definiert und verarbeitet werden, um bspw. aktuelle Portwerte oder Berechnungsergebnisse zwischenspeichern zu können.

Die Verwendung von endlichen, diskreten Zustandsautomaten mit den eben beschriebenen Eigenschaften und Inhalten ist für die Abbildung der Logik von

HMI-Komponenten zielführend. Zwar sind die generellen Möglichkeiten von Zustandsautomaten verglichen mit der Programmierung in einer Hochsprache eingeschränkt, jedoch ist die vereinfachte Nachbildung der Logik in der Regel hinreichend, um in der Phase der Konzeption ein interdisziplinäres Verständnis für das zu entwickelnde mechatronische System aufzubauen. Abschließend sei erwähnt, dass die mit dem Modellierungselement „Verhalten“ abgebildete Logik einer Mensch-Maschine-Schnittstelle vollständig losgelöst von eingesetzten Interaktionselementen ist (vgl. nachfolgender Abschnitt). Dadurch wird eine grundsätzliche Trennung von Funktion und Design bereits auf Modellebene adressiert, die sich generell in den vergangenen Jahren bei der HMI-Entwicklung bewährt hat (GODERIS 2008, S. 5 f.).

Interaktionselemente

Erst in einem letzten Schritt des hier entwickelten Ansatzes werden die sog. Interaktionselemente definiert, welche die technologische Lösung bzw. die Darstellung der zu entwickelnden Mensch-Maschine-Schnittstelle auf Modellebene abbilden. Die Modellierung von Interaktionselementen schließt sich typischerweise an die Technologieauswahl (Aktivität II) an, wodurch sich die qualitative zeitliche Einordnung dieser Aktivität in Abbildung 5-1 erklärt. Allgemein fokussiert der Ansatz an dieser Stelle haptische und graphische Interaktionstechnologien, während bspw. keine auditiven HMIs oder 3D-Gestensteuerungen abgebildet werden. Diese Einschränkung lässt sich einerseits mit der auch in absehbarer Zukunft vorhandenen Vorreiterrolle haptischer und graphischer Interaktionstechnologien im Maschinen- und Anlagenbau begründen (ZÜHLKE 2012, S. 70-72), andererseits sind weiterführende Interaktionstechnologien auf einer Modellebene nur bedingt darzustellen. Für den interdisziplinären Austausch und erste Nutzerevaluationen (vgl. Aktivität IV), die im Fokus der Konzeption von Mensch-Maschine-Schnittstellen stehen, ist es ausreichend, diese mit den vorgesehenen Möglichkeiten für haptische und graphische HMIs nachzubilden. Durch den modularen Aufbau der Modellierungstechnik besteht aber prinzipiell die Möglichkeit, zukünftig auch zusätzliche Interaktionselemente für bspw. Sprach- oder 3D-Gestensteuerungen zu integrieren.

Die Abbildung von Interaktionselementen erfolgt in der erweiterten IMoMeSA-Modellierungstechnik mit zwei wesentlichen Konzepten (vgl. Abbildung 5-6). Einerseits werden einfache Objekt-Repräsentationen eingesetzt, um die generelle Positionierung von einzelnen Interaktionselementen, wie bspw. Panel-PCs, Tastenfelder oder Bildschirme, im Kontext der 3D-Maschinengeometrie abzubilden. Andererseits können für modellierte Interaktionselemente auch konkrete Anzeige-

und Bedienelemente, wie Buttons oder Textfelder, in entsprechenden Screen-Repräsentationen abgebildet werden. Die Modellierung erfolgt dabei mit einfachen, schematischen Zeichnungen in Anlehnung an die in Abschnitt 2.4.2 eingeführten Wireframes. Der geringe Detaillierungsgrad dieser Beschreibungsform ist dabei für die Phase der Konzeption im Sinne einer einfachen Modellierung und Qualitätssicherung zielführend. Während für die Abbildung eines Interaktionselements eine einzelne Objekt-Repräsentation ausreichend ist, kommen in der Regel mehrere Screen-Repräsentationen zum Einsatz, wodurch die Modellierung unterschiedlicher Ansichten bei graphischen Mensch-Maschine-Schnittstellen ermöglicht wird. Innerhalb der Interaktionselemente wird weiterhin das Navigationskonzept spezifiziert, welches den Übergang zwischen den einzelnen Screen-Repräsentationen festlegt. Dazu können spezielle Navigationselemente modelliert werden, anhand derer der nächste aufzurufende Screen definiert wird. Die konkrete technische Spezifikation von Interaktionselementen kann dem Metamodell der Modellierungstechnik im Anhang A2 entnommen werden.

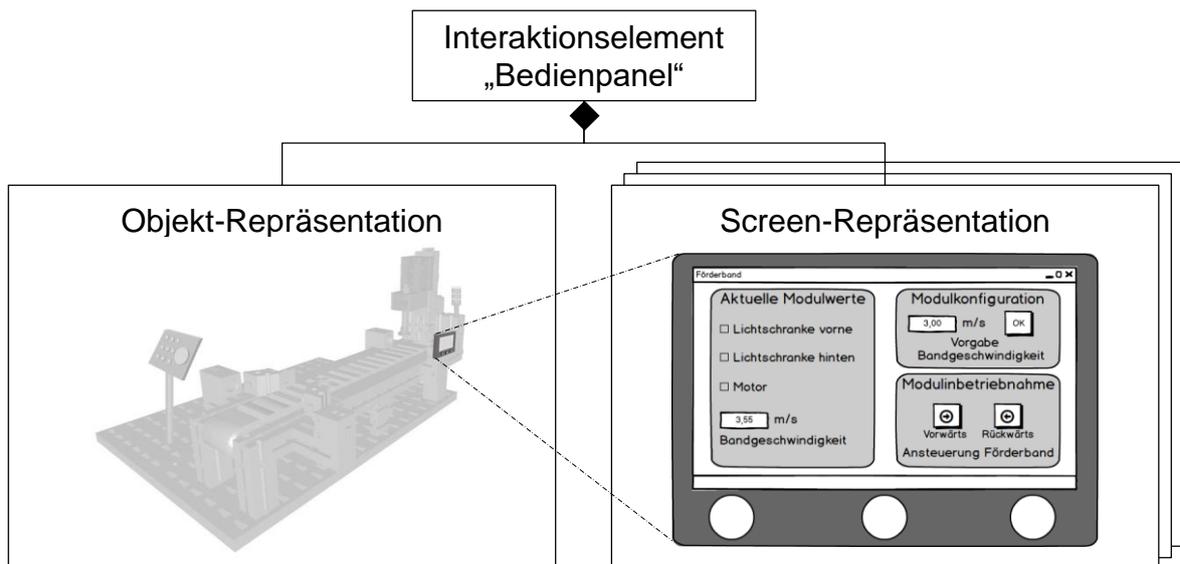


Abbildung 5-6: Aufbau von Interaktionselementen am Beispiel eines Bedienpanels

Analog zu den bisher vorgestellten Bestandteilen der Modellierungstechnik können Interaktionselemente innerhalb beliebiger mechatronischer Komponenten modelliert werden. Dies ermöglicht die Abbildung von Interaktionselementen, die nur für einzelne Systembestandteile zuständig sind, was vor allem im Anlagenbau auftreten kann. Die Bedien- und Anzeigeelemente innerhalb der Screen-Repräsentationen können ferner mit den HMI-Ports der zugehörigen mechatronischen Komponente sowie möglicher Teilkomponenten verknüpft werden, um die lösungsneutralen Ein- und Ausgabepoints den eingesetzten Interaktionselementen zuzuordnen.

5.2.2 Prozessablauf für den Einsatz der Modellierungstechnik während der Konzeption von mechatronischen Systemen

Die im letzten Abschnitt gewählte Reihenfolge für die Vorstellung der einzelnen Modellierungselemente spiegelt auf einer hohen Abstraktionsebene den Ablauf wider, wie diese Elemente während der Konzeption von mechatronischen Systemen eingesetzt werden. Betrachtet man das Vorgehen allerdings detaillierter, werden die einzelnen Elemente typischerweise nicht rein sequenziell genutzt. Vielmehr gibt es notwendige Iterationen und Rücksprünge, die nicht zuletzt von der Entwicklung des übergeordneten mechatronischen Systems abhängen. Vor diesem Hintergrund wird im Folgenden beschrieben, wie die erarbeitete Modellierungstechnik für die Abbildung von HMIs zielführend anzuwenden ist.

Dabei lässt sich die Aktivität der HMI-Modellierung in die drei Schritte unterteilen, die auch übergeordnet zur Gliederung der Konzeptionsphase von mechatronischen Systemen eingesetzt werden (vgl. Abschnitt 3.3.2 und Anhang A1). Dabei wird mit dem *Aufbau der mechatronischen Komponentenstruktur* begonnen, in dessen Rahmen aus Sicht der Mensch-Maschine-Schnittstelle primär Benutzeraufgaben im Umgang mit dem HMI innerhalb der mechatronischen Komponenten spezifiziert werden. Hierauf folgt die *Ausgestaltung der modellierten Komponenten*, die im Kontext der Mensch-Maschine-Schnittstelle anhand der HMI-Komponenten und deren integriertem Verhalten durchgeführt wird. Der letzte Schritt adressiert die *Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten*. Aus HMI-Sicht wird dabei die bestehende Modellierung erweitert, um u. a. robuster auf mögliches Fehlverhalten des Menschen reagieren zu können. Ferner erfolgt innerhalb dieses Schritts auch die Modellierung von Interaktionselementen, die zur Umsetzung der Benutzeraufgaben eingesetzt werden. Abbildung 5-7 fasst die eben eingeführten Schritte der HMI-Modellierung zusammen, die in den nachfolgenden Teilabschnitten noch im Detail betrachtet werden.

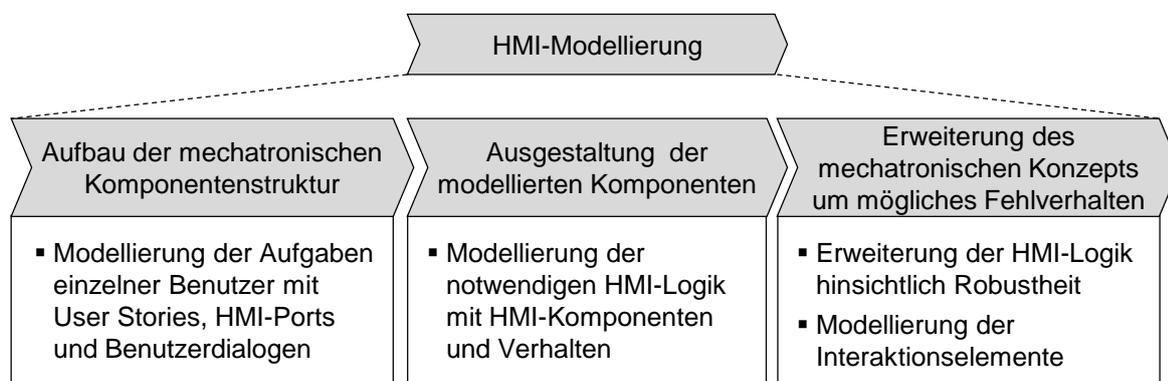


Abbildung 5-7: Prozessablauf zur HMI-Modellierung

Aufbau der mechatronischen Komponentenstruktur

Die Konzeptionsphase beginnt im Allgemeinen mit dem Anlegen einer Wurzelkomponente des mechatronischen Systems, innerhalb derer erste Anforderungen, Ports, Szenarien oder geforderte Eigenschaften abgebildet werden. Hinsichtlich des HMIs können zu diesem Zeitpunkt erste Benutzerrollen und User Stories modelliert werden, um übergreifende, meist noch generische Aufgaben (vgl. Abbildung 5-3) einzelner Benutzer zu spezifizieren. Weiterhin besteht die Möglichkeit, diese Aufgaben bereits mit HMI-Ports und Benutzerdialogen zu formalisieren, sofern alle für einen Dialog benötigten Ports direkt auf Systemebene modelliert werden und nicht einzelnen Komponenten zuzuordnen sind.

Gemäß dem ursprünglichen IMoMeSA-Ansatz wird weiterhin der funktionale Ablauf des zu entwickelnden Systems mit dem Modellierungselement „Monitor“ abgebildet. Dies dient als Basis, um das System in mechatronische Komponenten zu strukturieren. Für alle hierbei abgeleiteten Komponenten können anschließend erneut die Modellierungselemente der Analysephase eingesetzt werden, um bspw. konkretere Anforderungen an einzelne Komponenten abzubilden. Hinsichtlich der Benutzerinteraktion kann dieser Schritt analog durchgeführt werden. Dies beinhaltet zunächst die Abbildung von User Stories für Benutzeraufgaben, die speziell die Interaktion mit einer betrachteten Komponente betreffen. Auch diese Aufgaben können mit HMI-Ports und Benutzerdialogen formalisiert werden. Ggf. können die hierbei modellierten HMI-Ports auch für die Modellierung von Benutzerdialogen auf Systemebene eingesetzt werden, um User Stories zu formalisieren, für welche die notwendigen Ports zuvor noch nicht zur Verfügung standen.

Da jede modellierte Komponente in einzelne mechatronische (Sub)-Komponenten unterteilt werden kann, wiederholt sich das eben beschriebene Vorgehen so lange, bis ein zu entwickelndes System in hinreichend kleine Einheiten zerlegt wurde (vgl. Anhang A1). Als Ergebnis resultiert eine mechatronische Komponentenstruktur, in der aus HMI-Sicht alle Aufgaben der einzelnen Benutzerrollen durch User Stories, HMI-Ports und Benutzerdialoge spezifiziert und den jeweiligen Komponenten zugeordnet sind. Innerhalb der Hierarchie können User Stories untereinander oder mit Benutzerdialogen verknüpft werden, um etwaige Abhängigkeiten abbilden zu können. Weiterhin können HMI-Ports mit Benutzerrollen assoziiert werden, um auf Modellebene bereits ein Konzept für die spätere Verwaltung von Benutzerrechten definieren zu können. Abbildung 5-8 zeigt zusammenfassend eine mechatronische Komponentenstruktur, in der die HMI-spezifischen Bestandteile schematisch in den einzelnen Komponenten visualisiert sind.

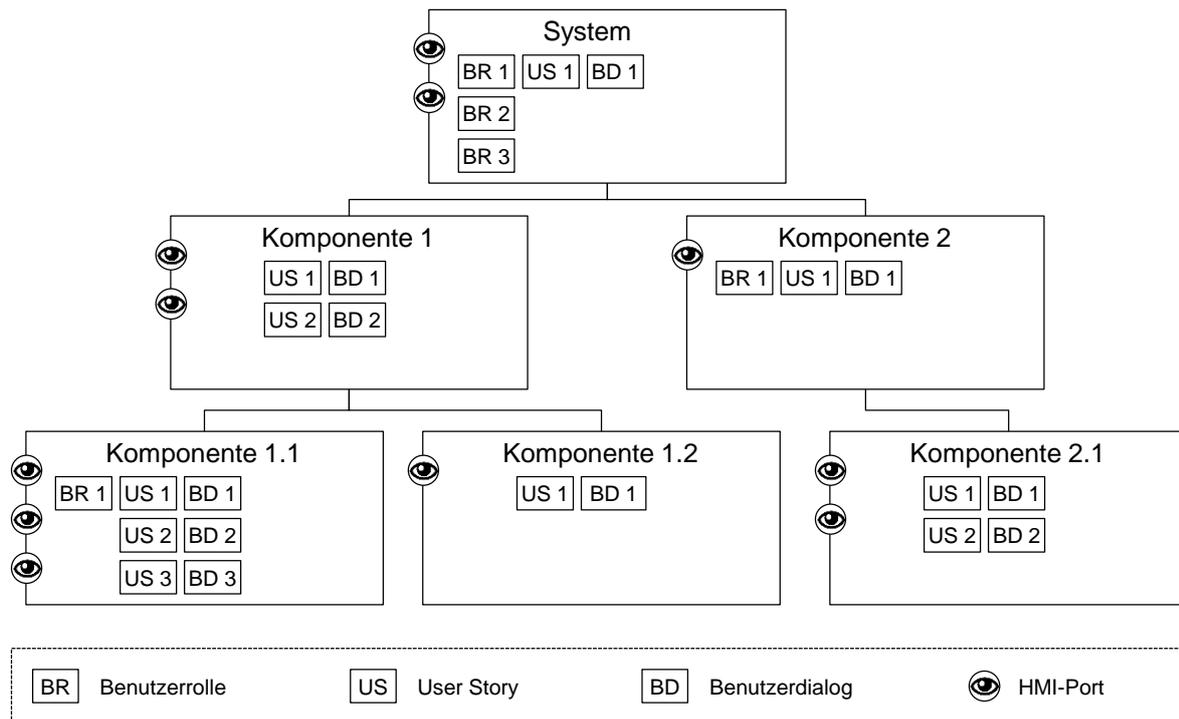


Abbildung 5-8: Exemplarisches Ergebnis einer mechatronischen Komponentenstruktur nach deren Aufbau (HMI-Sicht)

Das eben skizzierte Vorgehen zum Aufbau einer mechatronischen Komponentenstruktur adressiert die vollständige Neukonstruktion einer Maschine bzw. Anlage. Im Sinne der Wiederverwendung bietet es sich in der Praxis jedoch meist an, bei der Entwicklung auch auf bestehende Komponenten zurückzugreifen und diese zu einer Maschine bzw. Anlage zu komponieren. Durch den Aufbau von mechatronischen Systemen in voneinander gekapselte Komponenten wird dieser Gedanke bereits im ursprünglichen IMoMeSA-Ansatz berücksichtigt und lässt sich auf die Mensch-Maschine-Schnittstelle übertragen. Daher werden bei der Wiederverwendung einer mechatronischen Komponente auch die zugehörigen Benutzerrollen, User Stories, HMI-Ports und Benutzerdialoge, aber auch die HMI-Komponenten, Verhalten und Interaktionselemente in die zu entwickelnde Maschine integriert. Ferner besteht die Möglichkeit, nur einzelne Elemente, wie bspw. eine Benutzerrolle, innerhalb eines Baukastens bereitzustellen und beim Aufbau einer mechatronischen Komponentenstruktur zu verwenden. Hierbei ist allerdings zu beachten, dass dies im Hinblick auf Interaktionselemente nur für ein gesamtes Element gilt, während einzelne Screen-Repräsentationen nicht als Baukastenelement zur Verfügung stehen. Dies ist darauf zurückzuführen, dass die Inhalte eines Interaktionselements vor dem Hintergrund aller Benutzeraufgaben eines konkreten Anwendungsfalls zu erarbeiten sind, um ein gebrauchstaugliches HMI zu realisieren (MEIXNER & GÖRLICH 2008, S. 2).

Eine Besonderheit bei der Modellierung von Benutzeraufgaben im Rahmen des Aufbaus einer mechatronischen Komponentenstruktur ergibt sich durch den Aufbau der Modellierungstechnik. Da User Stories, HMI-Ports und Benutzerdialoge als spezielle Ausprägung von Anforderungen, Ports und Szenarien spezifiziert wurden, muss nicht zwingend im Voraus entschieden werden, ob eine konkrete Aufgabe von einem Menschen oder bspw. einer Automatisierungslösung übernommen wird. In diesem Fall kann eine Aufgabe zunächst informell als Anforderung oder formal mit Ports und Szenarien modelliert werden. Wenn während der Entwicklung festgelegt wird, dass eine modellierte Aufgabe durch einen Menschen zu realisieren ist, kann diese aufwandsarm in eine User Story bzw. einen Benutzerdialog überführt werden, da die jeweiligen Modellierungselemente auf einem ähnlichen Metamodell basieren (siehe Anhang A2). Ein typisches Beispiel einer solchen Aufgabe wäre bspw. das Einlegen von Material in eine Maschine, für die nicht a priori festgelegt werden muss, ob die Aufgabe von einem Menschen oder einem Roboter durchgeführt wird.

Zusammenfassend lässt sich festhalten, dass beim Aufbau einer mechatronischen Komponentenstruktur aus HMI-Sicht die Abbildung von Benutzeraufgaben im Fokus steht, um in der frühen Entwicklungsphase ein Verständnis für den Nutzungskontext zu generieren. Erst nach dessen interdisziplinärer Abstimmung und Freigabe im Sinne eines Quality Gates kann mit der Ausgestaltung der modellierten Komponenten fortgefahren werden. Dieser Schritt ist für die Realisierung einer gebrauchstauglichen Mensch-Maschine-Schnittstelle essenziell (vgl. BURMESTER 2008, S. 332), weshalb die Modellierung von Benutzeraufgaben durch Methoden des Usability Engineerings, wie bspw. eine Nutzerbeobachtung, unterstützt werden sollte. Ein Überblick relevanter Methoden und Anwendungsbereiche ist u. a. in ZÜHLKE (2012) oder RICHTER & FLÜCKINGER (2013) zu finden.

Ausgestaltung der modellierten Komponenten

Nach dem Aufbau einer mechatronischen Komponentenstruktur geht es in einem zweiten Schritt um die Ausgestaltung der modellierten Komponenten. Diese Aufgabe beginnt bei den mechatronischen Komponenten der untersten Hierarchieebene, die in ihre atomaren Bestandteile (Sensoren, Aktoren, Steuerungskomponenten, etc.) zerlegt werden. Diese werden wiederum mit Ports, Verhalten und Bauteilen modelliert und über Kanäle miteinander verbunden. Sobald alle mechatronischen Komponenten der untersten Hierarchieebene dadurch umgesetzt sind, können diese auf der nächsthöheren Ebene integriert werden. Neben der Verknüpfung der jeweiligen Ports mit Kanälen kann es bei diesem Schritt auch notwendig werden, neue atomare Bestandteile einzuführen, wenn bspw. für die Kommunikation zweier

Komponenten eine zusätzliche Gesamtsteuerung benötigt wird oder die mechanische Montage zweier Komponenten eine gemeinsame Grundplatte erfordert. Das resultierende Bottom-up-Vorgehen wird solange wiederholt, bis die letzten mechatronischen Komponenten auf Systemebene integriert sind.

Hinsichtlich der HMI-Entwicklung lässt sich der eben beschriebene Ablauf auf das Modellierungselement der HMI-Komponenten sowie deren Verhalten übertragen. Bei der Zerlegung einer mechatronischen Komponente der untersten Hierarchieebene in ihre atomaren Bestandteile werden somit auch HMI-Komponenten mit Ports und Verhalten modelliert und mit den Komponenten weiterer Disziplinen, wie Steuerungskomponenten (vgl. Abbildung 5-5), verknüpft. Durch diesen Schritt wird die HMI-Logik abgebildet, die zur Umsetzung der Benutzerdialoge der betrachteten mechatronischen Komponente notwendig ist. Wenn alle Komponenten der untersten Hierarchieebene auf diese Weise ausgestaltet sind, können in Analogie zu dem übergeordneten Bottom-up-Vorgehen die Benutzerdialoge auf der nächsthöheren Ebene realisiert werden. Dazu kommen erneut HMI-Komponenten mit Ports und Verhalten zum Einsatz. Da Benutzerdialoge auf die HMI-Ports untergeordneter Komponenten durchgreifen (vgl. Abschnitt 5.2.1), kann es notwendig werden, auch auf tieferen Ebenen neue HMI-Komponenten einzuführen oder das Verhalten bestehender HMI-Komponenten zu adaptieren. Das Vorgehen endet, wenn die Systemebene erreicht ist und die modellierten HMI-Komponenten somit eine Umsetzung aller spezifizierten Benutzerdialoge aus einer funktionalen Sicht ermöglichen. Um dies zu überprüfen, können Maßnahmen zur Qualitätssicherung eingesetzt werden, auf die in Abschnitt 5.5 detailliert eingegangen wird.

Hinsichtlich des eben beschriebenen Vorgehens ist zu erwähnen, dass zwischen mechatronischen Komponenten eine Verknüpfung von HMI-Ports mit Kanälen nicht notwendig ist. Dies liegt daran, dass Benutzerdialoge und Interaktionselemente auf beliebige HMI-Ports entlang der Komponentenstruktur zugreifen können (vgl. Abschnitt 5.2.1). Ein zweiter wichtiger Aspekt bei der Ausgestaltung modellierter Komponenten ist, dass nicht nur die Aufgaben realisiert werden, die während des Aufbaus der mechatronischen Komponentenstruktur mit User Stories und Benutzerdialogen abgebildet wurden. Vielmehr kann es sich ergeben, dass aus der Ausgestaltung auf Komponentenebene neue Aufgaben auf übergeordneten Ebenen resultieren. Ein Beispiel hierfür ist eine Rezeptverwaltung auf Systemebene, die sich erst als Ergebnis der komponentenspezifischen Modellierung als erforderlich herausstellt. Diese oder vergleichbare Aufgaben können mit User Stories, HMI-Ports und Benutzerdialogen abgebildet werden, für die anschließend eine Logik mit HMI-Komponenten und Verhalten modelliert werden kann.

Wie die bisherigen Ausführungen bereits deutlich machen, wird die Modellierung von Interaktionselementen bei der Ausgestaltung von modellierten Komponenten bewusst nicht adressiert. Dies lässt sich dadurch erklären, dass für die Auswahl einer geeigneten Interaktionstechnologie (vgl. Abschnitt 5.3) sowie deren Ausgestaltung alle Benutzeraufgaben berücksichtigt werden müssen (ZÜHLKE 2012, S. 51-53). Dies ist im Kontext der HMI-Modellierung erst nach der Erweiterung des mechatronischen Konzepts um potenzielles Fehlverhalten der Fall, da gerade im Maschinen- und Anlagenbau durch diese Betrachtung noch ein breites Spektrum an Benutzeraufgaben hinzukommen kann (SIEMON 2001, S. 44 ff.).

Vor diesem Hintergrund ergibt sich am Ende der Ausgestaltung der modellierten Komponenten ein um HMI-Komponenten und Verhalten erweitertes mechatronisches Konzept. Diese Komponenten können auch mit User Stories verknüpft werden, um einen Zusammenhang mit modellierten Anforderungen herzustellen. Analog zum Aufbau der Komponentenstruktur bietet es sich an dieser Stelle erneut an, das erarbeitete Modell interdisziplinär abzustimmen und erst nach einer gemeinsamen Freigabe mit der Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten fortzufahren. Abbildung 5-9 zeigt die resultierende Komponentenstruktur für das bereits im letzten Abschnitt angeführte Beispiel, welches neben den HMI-Komponenten auch neu hinzugekommene Benutzeraufgaben in Form von User Stories, HMI-Ports und Benutzerdialogen beinhaltet.

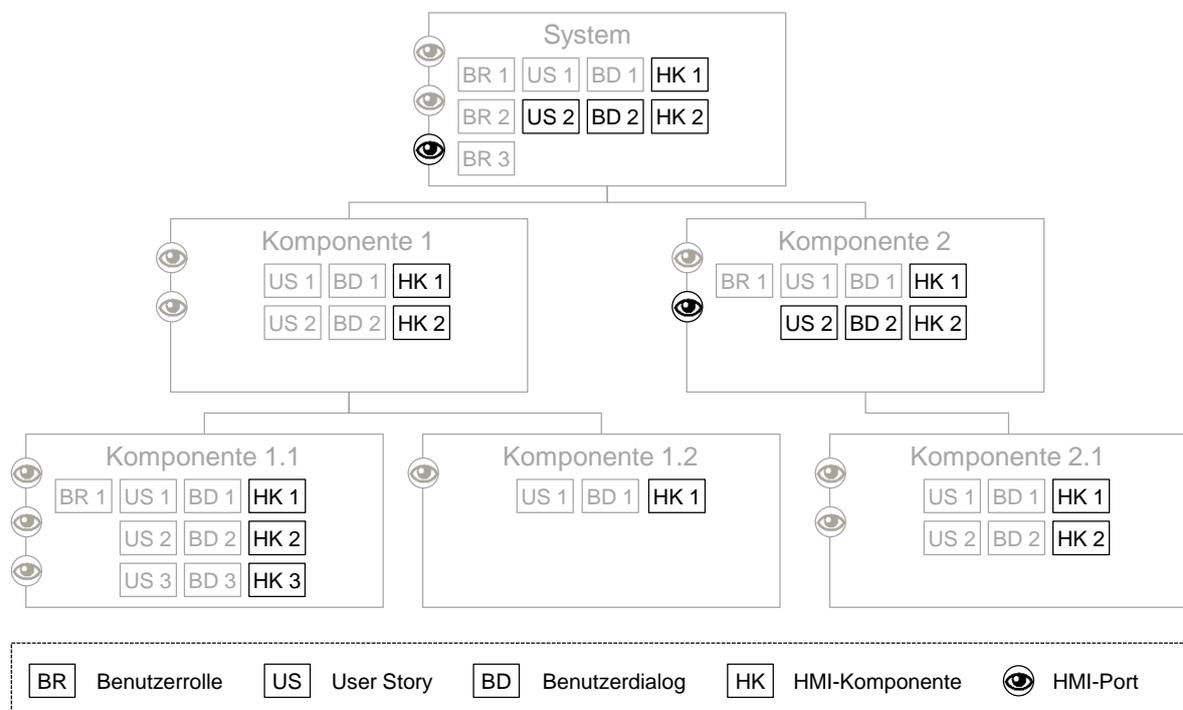


Abbildung 5-9: Exemplarisches Ergebnis einer mechatronischen Komponentenstruktur nach der Ausgestaltung der modellierten Komponenten (HMI-Sicht)

Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten

Im letzten Schritt der Konzeption wird ein mechatronisches Konzept um mögliches Fehlverhalten erweitert, um dadurch seine Robustheit im Betrieb zu erhöhen. Dazu werden zunächst die Verhaltensweisen einzelner mechatronischer Komponenten hinsichtlich potenzieller Fehler, wie z. B. dem Ausfall eines Sensors, adaptiert. Ein solches Fehlverhalten kann anschließend in einem Szenario über einen sog. Eventport ausgelöst werden (vgl. Anhang A1). Die Auswirkungen dieses Fehlers können in einer Simulation beobachtet werden. Auf dieser Basis kann entschieden werden, ob eine Anpassung des Verhaltens notwendig ist, die den eingebrachten Fehler beheben oder das System in einen sicheren Zustand überführen kann.

Da in der industriellen Praxis meist Menschen in die Fehlerbehebung involviert sind, resultiert aus diesem Entwicklungsschritt in der Regel eine Vielzahl neuer Benutzeraufgaben und Interaktionsmöglichkeiten. Um diese auf Modellebene abzubilden und so einen ganzheitlichen Überblick hinsichtlich des Aufgabenspektrums der Benutzer zu erhalten, ist die Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten ein essenzieller Bestandteil. Zur Abbildung der neuen Aufgaben können analog zum bisherigen Vorgehen zunächst User Stories, HMI-Ports und Benutzerdialoge genutzt werden. Auf dieser Basis kann anschließend die notwendige Logik mit HMI-Komponenten und Verhalten modelliert werden.

Neben dem Einbringen von Fehlern in einzelnen Systemkomponenten besteht auch die Möglichkeit, ein Fehlverhalten von Benutzern zu modellieren, um dadurch die Robustheit des HMIs zu erhöhen. Zu diesem Zweck können modellierte Benutzerdialoge bspw. dahingehend variiert werden, dass Benutzer ungültige Eingabewerte an das System übermitteln oder Interaktionen in einer falschen Reihenfolge tätigen. In der Dialogvariante kann dann die gewünschte Reaktion des Systems auf diese fehlerhafte Eingabe definiert werden. Diese kann bspw. darin bestehen, dass die Übermittlung eines Werts an das System verhindert und der Benutzer darüber informiert wird. Zur Umsetzung der definierten Reaktion müssen weiterhin die zugehörigen HMI-Komponenten in ihrem Verhalten adaptiert werden. Mit diesem Vorgehen kann aus Aufwandsgründen immer nur ein Ausschnitt des möglichen Benutzerverhaltens abgebildet werden. Durch den Einsatz von Methoden des Usability Engineerings, wie z. B. Nutzerbeobachtungen, kann allerdings sichergestellt werden, dass dieser Ausschnitt repräsentativ für die jeweilige Anwendung ist. Darüber hinaus sind in der Qualitätssicherung sog. Nutzerevaluationen mit dem HMI-Modell vorgesehen (vgl. Abschnitt 5.5), um auf Bedienfehler reagieren zu können, die mit dem beschriebenen Vorgehen nicht erfasst und behandelt werden.

Erst zum Ende der Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten liegt ein vollständiger Überblick hinsichtlich der Benutzeraufgaben eines Anwendungsfalls vor, der allerdings für die Auswahl geeigneter Technologien (vgl. Abschnitt 5.3) und somit die Abbildung von Interaktionselementen erforderlich ist. Daher werden erst im letzten Schritt der Konzeption Interaktionselemente in die Komponentenstruktur eingebettet. Dabei bietet es sich an, zunächst die Objekt-Repräsentationen abzubilden, anschließend ein Navigationskonzept für die einzelnen Screen-Repräsentationen festzulegen und diese anschließend mit Bedien- und Anzeigeelementen anzureichern.⁹ Schließlich erfolgt deren Verknüpfung mit HMI-Ports, welche für die Nutzerevaluationen (vgl. Abschnitt 5.5) notwendig ist.

Zusammenfassend zeigt Abbildung 5-10 die resultierende Komponentenstruktur des vollständigen mechatronischen Konzepts, welches vor Beginn der Verfeinerung noch ein letztes Mal im gesamten Entwicklungsteam abzustimmen ist. Neben der Integration der Interaktionselemente erkennt man die aus Fehlerbehandlungen hinzugekommenen User Stories und Benutzerdialoge sowie daraus resultierende HMI-Komponenten, aber auch die modellierten Varianten von bestehenden Benutzerdialogen und die diesbezüglichen Adaptionen an den HMI-Komponenten.

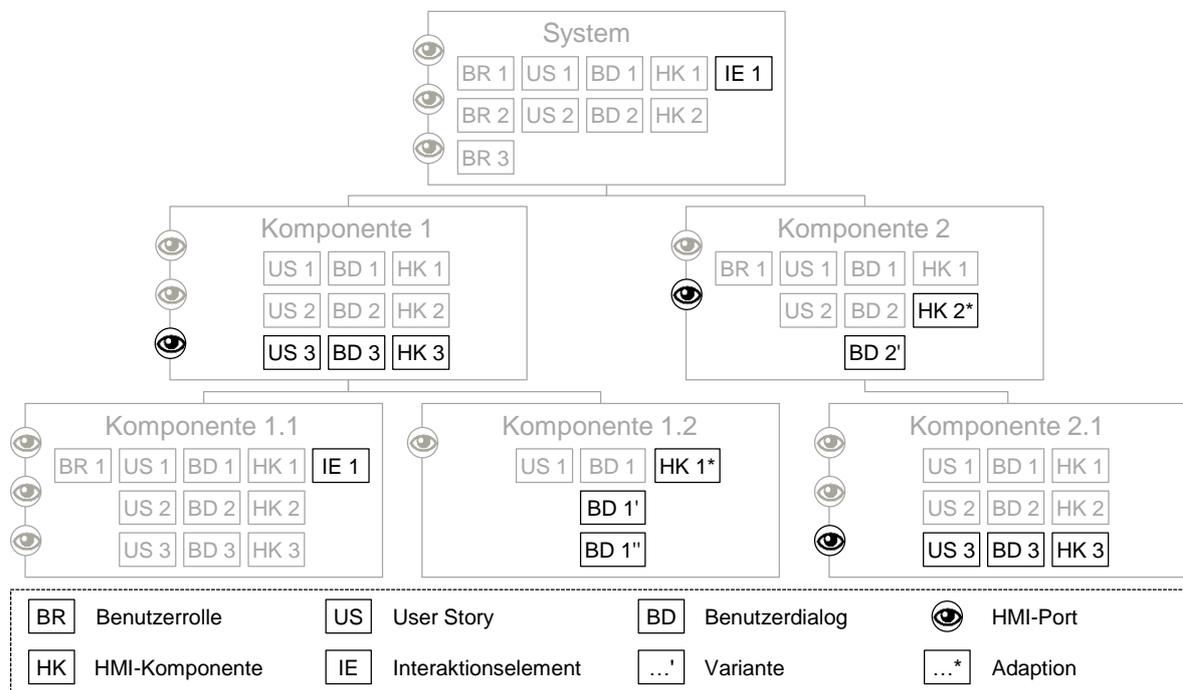


Abbildung 5-10: Exemplarisches Ergebnis einer mechatronischen Komponentenstruktur nach der Erweiterung um mögliches Fehlverhalten (HMI-Sicht)

⁹ Ein detailliertes Vorgehen zum Aufbau und zur Gestaltung der Interaktionselemente wird an dieser Stelle nicht vorgeschlagen, da hierfür auf bestehende Ansätze, wie bspw. von GALITZ (2007), und Grundsätze der Softwareergonomie (vgl. Abschnitt 2.3) zurückgegriffen werden kann.

5.3 Aktivität II: Technologieauswahl

Die Auswahl einer geeigneten Interaktionstechnologie ist für die Umsetzung eines gebrauchstauglichen HMIs ebenso wichtig wie die frühzeitige Fokussierung von Benutzern und deren Aufgaben. Im Umfeld des Usability Engineerings wird hierfür meist die Erprobung unterschiedlicher Technologien mit späteren Nutzern vorgeschlagen. Diese umfangreichen Tests können in den Entwicklungsprozessen des Maschinen- und Anlagenbaus jedoch aus Kapazitäts- und Kostengründen in der Regel nicht durchgeführt werden. Daher wird im Rahmen dieser Arbeit das grundsätzliche Vorgehen gewählt, den jeweils vorliegenden Anwendungsfall anhand eindeutiger Auswahlfaktoren, wie bspw. die Anzahl der Benutzeraufgaben, einzuordnen und hieraus einen Vorschlag hinsichtlich einzusetzender Interaktionstechnologien abzuleiten. Aufgrund der Vielzahl unterschiedlicher Technologien sowie der Abhängigkeit der Technologieauswahl von Aspekten, die über den konkreten Anwendungsfall hinausgehen, wie bspw. eine Investitionsrechnung, wird im Rahmen dieser Arbeit von der Auswahl einer spezifischen Technologie abgesehen. Stattdessen wird das Ziel verfolgt, die aus Sicht eines konkreten Anwendungsfalls geeigneten und notwendigen Merkmale einer Mensch-Maschine-Schnittstelle zu identifizieren, die bei der Technologieauswahl einzubeziehen sind. Das hierfür erarbeitete Schema zur Gegenüberstellung von Auswahlfaktoren und Interaktionstechnologien wird im nachfolgenden Abschnitt 5.3.1 vorgestellt. Auf dieser Basis wird in Abschnitt 5.3.2 erläutert, wie dieses Schema in die Technologieauswahl einfließen kann und wie die gesamte Aktivität in die übergeordnete Entwicklungsmethodik eingebettet ist.

5.3.1 Gegenüberstellung von Auswahlfaktoren und Interaktionstechnologien in einer Einflussmatrix

Zur Gegenüberstellung von Auswahlfaktoren und Interaktionstechnologien bedarf es zunächst einer *Identifikation relevanter Auswahlfaktoren* für einen konkreten Anwendungsfall sowie einer *Klassifikation von Interaktionstechnologien* anhand eindeutiger Unterscheidungsmerkmale. Diese Beschreibung bildet die Basis für eine *Gegenüberstellung in einer Einflussmatrix*. In dieser wird anhand eines festen Bewertungsschemas definiert, inwiefern die Auswahlfaktoren Rückschlüsse hinsichtlich der Merkmale von Interaktionstechnologien ermöglichen. Wenn die für einen konkreten Anwendungsfall zutreffenden Ausprägungen der Auswahlfaktoren ausgewählt werden, können anhand der Einflussmatrix diejenigen Merkmale identifiziert werden, die eine einzusetzende Interaktionstechnologie aufweisen sollte.

Identifikation relevanter Auswahlfaktoren

Nach ZÜHLKE (2012) sind bei der Identifikation relevanter Auswahlfaktoren unterschiedliche Stakeholder und Rahmenbedingungen zu berücksichtigen. Diese haben meist verschiedene und teilweise konträre Anforderungen, die allerdings vollständig in die Technologieauswahl einzubeziehen sind. Vor diesem Hintergrund wurden im Rahmen dieser Arbeit zunächst auf Basis von Literaturrecherchen (vgl. Abschnitte 2.2, 2.3 und 3.4) sowie Expertenworkshops (vgl. Abschnitt 6.1) insgesamt zwölf Stakeholder und Rahmenbedingungen ermittelt, die jeweils eigene Anforderungen an eine einzusetzende Interaktionstechnologie stellen. Abbildung 5-11 gibt einen Überblick dieser Stakeholder und Rahmenbedingungen, wobei die Stakeholder zur einfacheren Einordnung weiterhin in die Kategorien *Hersteller* (= Maschinen- bzw. Anlagenbauer) und *Kunde* unterteilt wurden.

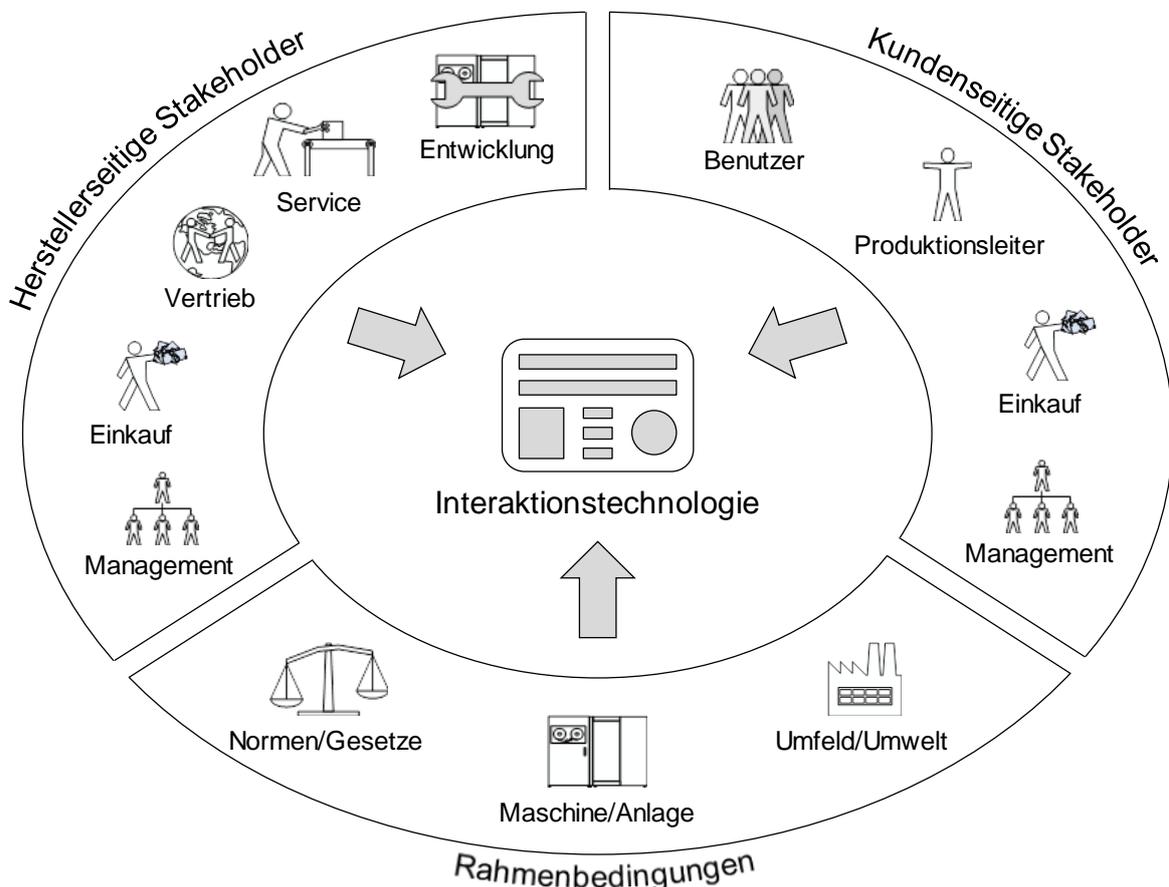


Abbildung 5-11: Stakeholder und Rahmenbedingungen bei der Technologieauswahl

Auf Herstellerseite ergeben sich wichtige Auswahlfaktoren zunächst aus der *Entwicklung*, da u. a. die Größe des Entwicklungsteams oder das Programmier-Know-how einen großen Einfluss auf die einzusetzende Interaktionstechnologie

besitzen. Weiterhin formuliert der *Service* wichtige Anforderungen, bspw. hinsichtlich des Rechtemanagements oder der Notwendigkeit von Fernwartungen. Der *Vertrieb* agiert ebenfalls an der Schnittstelle zum Kunden, wodurch sich u. a. Auswahlfaktoren hinsichtlich der Distributionskanäle oder der kundenspezifischen Konfigurierbarkeit des HMIs ergeben. Aus dem *Einkauf* ergeben sich hingegen Auswahlfaktoren, die u. a. die Verfügbarkeit einer Technologie oder bestehende Lieferantenbeziehungen adressieren. Das *Management* liefert schließlich Auswahlfaktoren, die meist aus der allgemeinen Unternehmensstrategie resultieren, wie bspw. die Innovationsbereitschaft oder zu erreichende Preissegmente.

Auf Seite des Kunden liefern vor allem die *Benutzer* zentrale Faktoren, um eine geeignete Technologie auszuwählen. Das Spektrum der zu berücksichtigenden Faktoren reicht vom jeweiligen Bildungsstand bis zur eingesetzten persönlichen Schutzausrüstung. Neben dem Bedienpersonal liefern auf Kundenseite aber auch *Produktionsleiter* wichtige Anforderungen an das HMI, von denen bspw. ein einheitlicher Technologieeinsatz im Maschinenpark gewünscht werden kann. Analog zum Anbieter nimmt auch der *Einkauf* des Kunden eine wichtige Rolle ein, dessen wesentliche Anforderungen meist auf Kosten- und Qualitätsaspekte zurückzuführen sind. Schließlich ist auch das *Management* auf Kundenseite für die Technologieauswahl ein wichtiger Stakeholder, der indirekte Anforderungen an das HMI bspw. über die geforderte Produktqualität oder Preispolitik formuliert.

Darüber hinaus haben generelle Rahmenbedingungen, wie *Normen* oder *Gesetze*, einen wichtigen Einfluss auf die einzusetzende Interaktionstechnologie. Aus dieser ersten Gruppe können Anforderungen der Datensicherheit oder einzuhaltender Zertifizierungen, wie bspw. die CE-Kennzeichnung, resultieren. Eine zweite zentrale Rahmenbedingung bildet weiterhin die *Maschine* oder *Anlage*, mit der die Benutzer interagieren. Auswahlfaktoren ergeben sich hierbei u. a. aus der generellen Anzahl der durchzuführenden Bedienoperationen oder der Zugänglichkeit und Anbringungsmöglichkeiten für Interaktionstechnologien. Eine letzte Gruppe bildet das *Umfeld* bzw. die *Umwelt*, in der die Interaktionstechnologie eingesetzt wird. So haben bspw. die Lautstärke des Umfelds oder die vorherrschenden Lichtverhältnisse einen signifikanten Einfluss auf die möglichen Interaktionstechnologien.

In den vorangegangenen Absätzen wurden für die jeweiligen Stakeholder und Rahmenbedingungen bereits exemplarische Auswahlfaktoren vorgestellt. In Summe wurden im Rahmen der vorliegenden Arbeit 66 Auswahlfaktoren identifiziert, die in Anhang A3 weiterführend beschrieben sind. Im Rahmen dieses Abschnitts wird nachfolgend die erarbeitete Struktur für die formale Erfassung von Auswahlfaktoren

erläutert (vgl. Abbildung 5-12), welche für eine Gegenüberstellung mit den Klassifikationskriterien zielführend ist. Diese Struktur ermöglicht es, das Spektrum der Auswahlfaktoren zukünftig zu erweitern, wenn für spezifische Anwendungsfälle weitere Faktoren für die Technologieauswahl relevant werden.

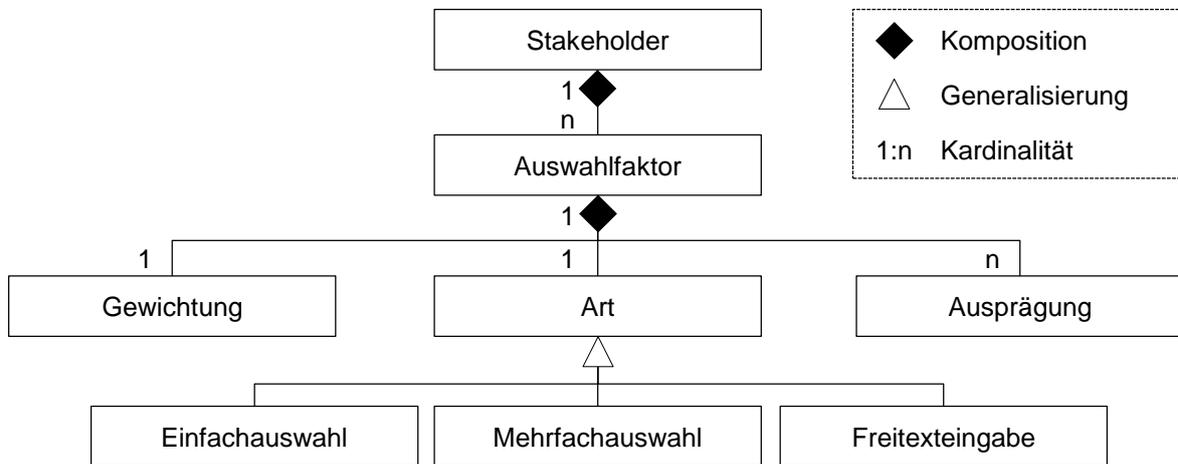


Abbildung 5-12: Struktur zur Erfassung von Auswahlfaktoren

Den Ausgangspunkt der in Abbildung 5-12 dargestellten Struktur bilden die *Stakeholder*, die für die Formulierung der Auswahlfaktoren verantwortlich sind. Die *Auswahlfaktoren* selbst setzen sich aus drei konstituierenden Elementen zusammen. Unter diesen ist zunächst die *Gewichtung* zu nennen, über welche die Relevanz eines Auswahlfaktors in Bezug auf den vorliegenden Anwendungsfall definiert wird. Im Rahmen der vorliegenden Arbeit wurde hierfür ein Schema mit den Abstufungen von 0 (keine Relevanz) bis 3 (hohe Relevanz) gewählt. Für jeden Auswahlfaktor ist weiterhin seine *Art* festzulegen, wobei insgesamt drei Arten unterschieden werden. Die häufigste Form ist dabei die *Einfachauswahl*, bei der aus einer gegebenen Menge an *Ausprägungen* eines Auswahlfaktors die für den Anwendungsfall zutreffende Ausprägung ausgewählt wird. Ein Beispiel ist der Auswahlfaktor „Bildungsstand der Benutzer“, dessen Ausprägungen von „Analphabeten“ bis zu „studiertes Personal“ reichen. Darüber hinaus gibt es auch Auswahlfaktoren bei denen mehrere Ausprägungen für einen konkreten Anwendungsfall zutreffen können. Dies tritt bspw. bei der Bestimmung von notwendigen Zertifizierungen für eine Interaktionstechnologie (CE, CSA, etc.) auf. Schließlich bietet es sich für spezielle Auswahlfaktoren an, gänzlich auf vorformulierte Ausprägungen zu verzichten und stattdessen die für einen Anwendungsfall zutreffende Ausprägung als Freitext zu spezifizieren. Dies ist insbesondere bei der Eingabe von numerischen Grenzwerten, wie bspw. der Umgebungstemperatur für das HMI, zielführend.

Klassifikation von Interaktionstechnologien

Grundsätzliche Möglichkeiten zur Klassifikation von Interaktionstechnologien wurden in der Literatur bereits von unterschiedlichen Autoren (BAUMANN & LANZ (2013), HEINECKE (2012), ZÜHLKE (2012), KRAUB (2003), etc.) vorgestellt. Die verwendeten Kriterien zur Einordnung der Technologien sind jedoch zu generisch, um sie unmittelbar bei der Gegenüberstellung in einer Einflussmatrix verwenden zu können. Vor diesem Hintergrund wurde im Rahmen dieser Arbeit ein auf der Literatur basierendes und in Expertenworkshops (vgl. Abschnitt 6.1) konkretisiertes Schema zur Klassifikation von Interaktionstechnologien erarbeitet.

Dieses Schema sieht in einem ersten Schritt eine hierarchische Unterteilung von Interaktionstechnologien in sog. *Technologieklassen* vor (vgl. Abbildung 5-13). Auf oberster Ebene erfolgt eine grundsätzliche Einteilung von Interaktionstechnologien in Hard- und Software, um Anwendern in beiden Bereichen eine Unterstützung bei der Technologieauswahl anzubieten. Hinsichtlich der Hardware wird weiterhin eine Unterteilung in Interaktionsgeräte und die zugehörigen Recheneinheiten vorgeschlagen.¹⁰ Für die Interaktionsgeräte bietet sich ferner eine Unterteilung anhand der Informationsrichtung an, wobei zwischen Eingabegeräten, Ausgabegeräten sowie kombinierten Ein- und Ausgabegeräten unterschieden wird. Unter letztgenannten werden aktuell ausschließlich sog. direkte Zeigergeräte zusammengefasst, die auch als Touchscreens bezeichnet werden. Somit bietet sich eine weiterführende Unterteilung nur für die Ein- und Ausgabegeräte an, wobei als Differenzierungsmerkmal die Sinnesmodalitäten des Menschen genutzt werden. Dies führt sowohl auf Eingabe- als auch auf Ausgabeseite zu einer Einteilung in visuelle, akustische und haptische Geräte. Innerhalb dieser Gliederung lassen sich auf einer letzten Ebene schließlich spezielle Ausprägungen als jeweils eigene Technologieklassen definieren, wobei aus Übersichtsgründen in Abbildung 5-13 lediglich die Unterteilung der haptischen Eingabegeräte in Positioniergeräte, Tastaturgeräte und Befehlsgeber dargestellt ist. Ein weiteres Beispiel wäre die Unterteilung von visuellen Ausgabegeräten in Bildschirme, 2D- und 3D-Projektionen.

Hinsichtlich der Unterteilung von Software bietet sich eine ähnliche Gliederung wie bei der Hardware an, da für spezifische Hardwarelösungen in der Regel auch spezifische Softwarelösungen eingesetzt werden, wie bspw. für die Verarbeitung von Sprachsignalen. Daher wird zunächst zwischen einer Eingabesoftware und

¹⁰ Diese Unterteilung ist notwendig, da viele Interaktionsgeräte, wie bspw. Tastaturen oder Computermäuse, nicht ohne entsprechende Recheneinheiten funktionieren würden, an die wiederum eigene Anforderungen, wie z. B. im Hinblick auf die Prozessorleistung, formuliert werden können.

einer Ausgabesoftware unterschieden. Auf eine kombinierte Ein- und Ausgabesoftware kann an dieser Stelle verzichtet werden, da bei Software immer zwischen notwendigen Funktionsumfängen für die Verarbeitung von Nutzereingaben und die Bereitstellung von Ausgaben unterschieden werden kann. Innerhalb dieser beiden Technologieklassen erfolgt die weiterführende Unterteilung analog zur Hardware anhand der Sinnesmodalitäten des Menschen, so dass sich jeweils eine Klassifikation in visuelle, akustische und haptische Softwarelösungen ergibt. Eine weitere Gliederung von Software ist nicht mehr erforderlich, da bereits auf dieser Ebene die notwendigen Funktionsumfänge von Softwarelösungen anhand der nachfolgend eingeführten Klassifikationskriterien eindeutig charakterisiert werden können.

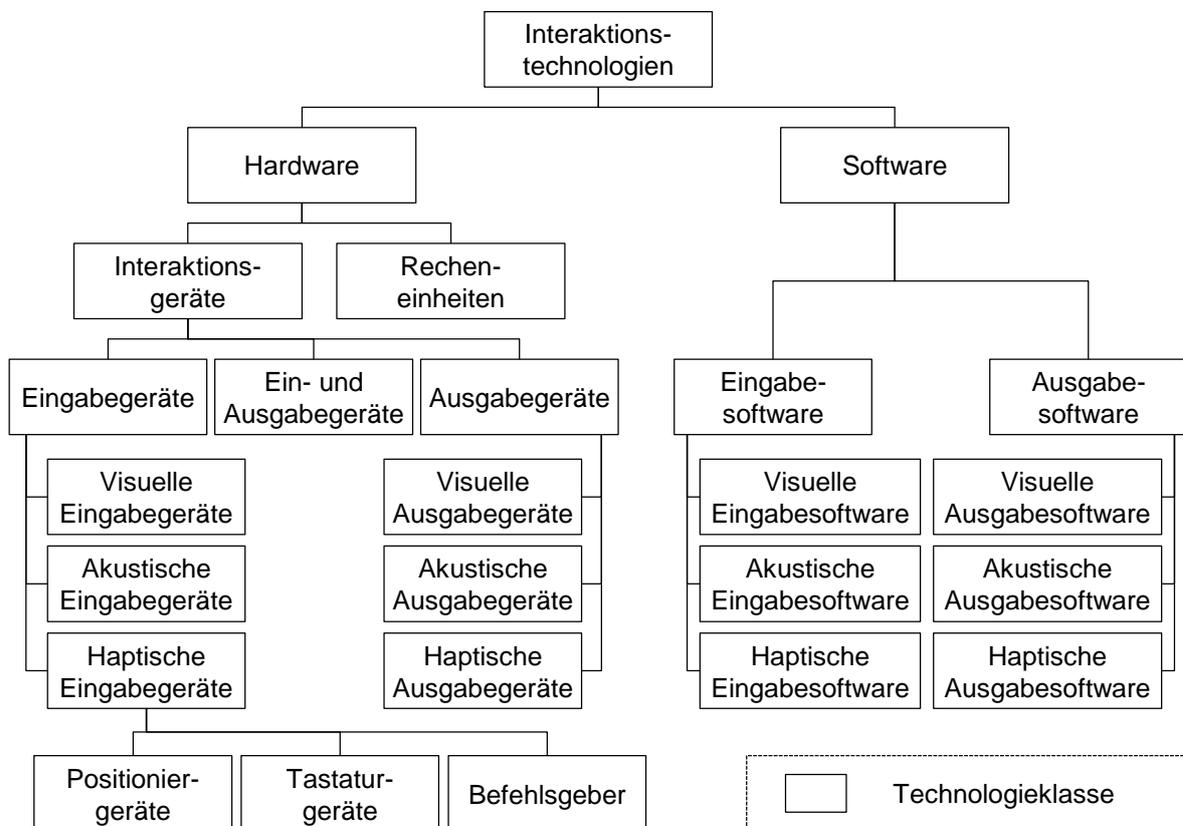


Abbildung 5-13: Technologieklassen für Interaktionstechnologien

Die in Abbildung 5-13 gezeigten Technologieklassen nehmen eine strikte Trennung von Interaktionstechnologien anhand der Sinnesmodalitäten des Menschen vor, was in Zeiten multimodaler Mensch-Maschine-Schnittstellen (vgl. Abschnitt 2.2.2) nicht zielführend erscheint. Allerdings können diese Systeme immer als Aggregation einzelner Interaktionsgeräte interpretiert werden, wie am Beispiel des Smartphones deutlich wird, das eine Spracheingabe über ein Mikrofon mit einer Sprachausgabe über einen Lautsprecher sowie einer visuellen Ausgabe und haptischen Eingabe

über einen Touchscreen kombiniert. Auf die einzelnen Bestandteile eines solchen Interaktionssystems können die nachfolgend eingeführten Klassifikationskriterien angewendet werden, anhand derer bei der späteren Technologieauswahl entschieden wird, ob diese Bestandteile den Anforderungen eines konkreten Anwendungsfalls genügen.

Wie bereits in den vorangegangenen Absätzen deutlich wurde, ist das eingeführte Schema der Technologieklassen alleine noch nicht ausreichend, um die Auswahl einer Interaktionstechnologie zu unterstützen. Hierfür bedarf es weiterhin der bereits erwähnten *Klassifikationskriterien*, die den einzelnen Technologieklassen zugeordnet werden und bei der späteren Gegenüberstellung detaillierte Aussagen hinsichtlich der Eignung einer spezifischen Interaktionstechnologie für einen konkreten Anwendungsfall ermöglichen. Analog zu den Auswahl Faktoren werden die insgesamt 69 Klassifikationskriterien erst im Anhang A3 im Detail vorgestellt, während an dieser Stelle die grundsätzliche Struktur (vgl. Abbildung 5-14) dargestellt und anhand ausgewählter Beispiele erläutert wird.

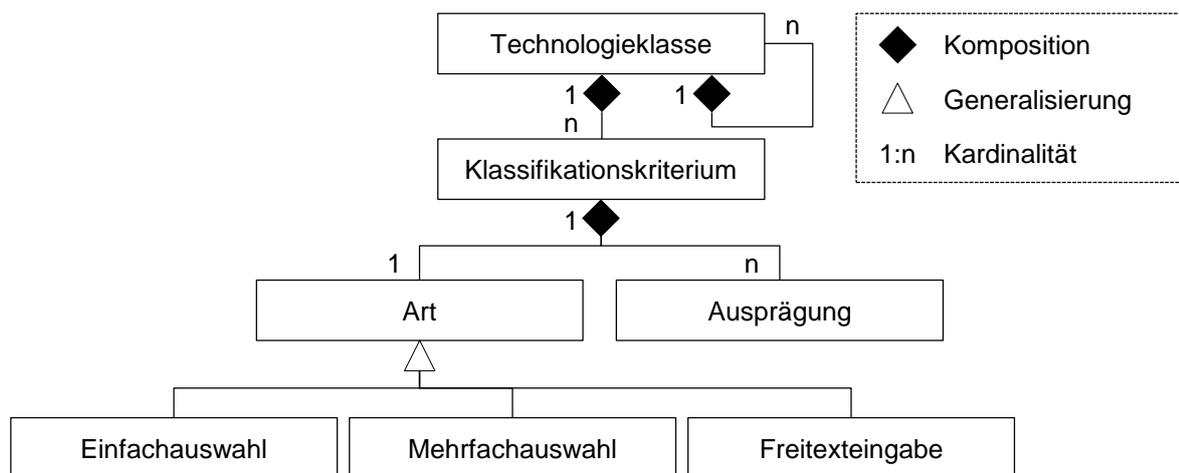


Abbildung 5-14: Struktur zur Erfassung von Klassifikationskriterien

Die Struktur wurde dabei analog zu den Auswahl Faktoren (vgl. Abbildung 5-12) gewählt, was insbesondere für deren spätere Gegenüberstellung von Vorteil ist. Die Basis bilden daher die definierten Technologieklassen, die jeweils auch abgeleitete Technologieklassen beinhalten können (vgl. Abbildung 5-13). Weiterhin setzt sich jede Technologieklasse aus einer bestimmten Anzahl an *Klassifikationskriterien* zusammen. Dabei werden allgemeine Kriterien, wie bspw. der Preis oder das Erscheinungsdatum, die für alle Interaktionstechnologien relevant sind, direkt dieser übergeordneten Klasse zugeordnet. Demgegenüber werden spezifischere Kriterien, wie bspw. die Displaygröße, auch einer spezifischeren Technologieklasse, wie in

diesem Fall der Klasse „Bildschirme“, zugeordnet. Generell wurde die Zuordnung so umgesetzt, dass ein Kriterium, das für alle Technologieklassen einer Hierarchieebene, wie z. B. für alle Eingabegeräte, relevant ist, der jeweils übergeordneten Technologiekategorie zugeordnet wird, um ein mehrfaches Anlegen des gleichen Kriteriums zu vermeiden. Weiterhin ist für jedes Klassifikationskriterium analog zu den Auswahl Faktoren die *Art* entscheidend, die angibt, ob eine Einfach- oder eine Mehrfachauswahl vordefinierter *Ausprägungen* oder eine Freitexteingabe für die Klassifikation einer konkreten Interaktionstechnologie anhand des jeweiligen Kriteriums verwendet wird. Auch bei den Klassifikationskriterien tritt der Fall der Einfachauswahl am häufigsten auf, wie bspw. beim Kriterium „Portabilität“ mit den Ausprägungen „Mobil“ und „Stationär“, das der Technologiekategorie „Hardware“ zugeordnet ist. Im Bereich der Mehrfachauswahl ist demgegenüber das Beispiel „Stromversorgung“ anzuführen, da einzelne Interaktionsgeräte ggf. über mehrere Möglichkeiten, wie bspw. einen Akkumulator und ein Netzkabel, verfügen. Die Freitexteingabe kommt hingegen meist bei Klassifikationskriterien zum Einsatz, bei denen die Ausprägung durch einen numerischen Wert gekennzeichnet ist. Dies tritt bspw. bei dem bereits zuvor genannten Kriterium des Erscheinungsdatums, aber auch bei der Hitzebeständigkeit einer spezifischen Technologie auf.

Die tatsächliche Klassifikation einer konkreten Technologie eines Herstellers von Interaktionstechnologien kann auf Basis der eingeführten Klassifikationskriterien anhand des *Vererbungsprinzips* durchgeführt werden. Nach diesem Prinzip werden auf eine konkrete Technologie alle Kriterien der zugehörigen Technologiekategorie sowie der jeweils übergeordneten Klassen angewendet. Am Beispiel eines spezifischen Drehreglers soll dieses Prinzip erläutert werden (vgl. Abbildung 5-15). Allgemein sind Drehregler der Kategorie „Befehlsgeber“ zuzuordnen, weswegen in einem ersten Schritt eine Bewertung anhand der Kriterien dieser Kategorie erfolgt. Weiterhin werden auf den spezifischen Drehregler aber ebenfalls die Kriterien der jeweils übergeordneten Technologiekategorien von den haptischen Eingabegeräten bis zu den Interaktionstechnologien angewendet. Erst durch die Anwendung aller Kriterien auf den spezifischen Drehregler ist dieser eindeutig klassifiziert und kann hinsichtlich seiner Eignung für einen konkreten Anwendungsfall bewertet werden.

Abschließend sei erwähnt, dass das entwickelte Schema zur Klassifikation von Interaktionstechnologien aktuell auf am Markt verfügbare Lösungen ausgerichtet ist. Allerdings wurde bewusst auf eine Erweiterbarkeit dieses Schemas geachtet, da für zukünftige Interaktionstechnologien ohne Aufwände neue Technologiekategorien mit jeweils eigenen Kriterien in die bestehende Struktur integriert werden können.

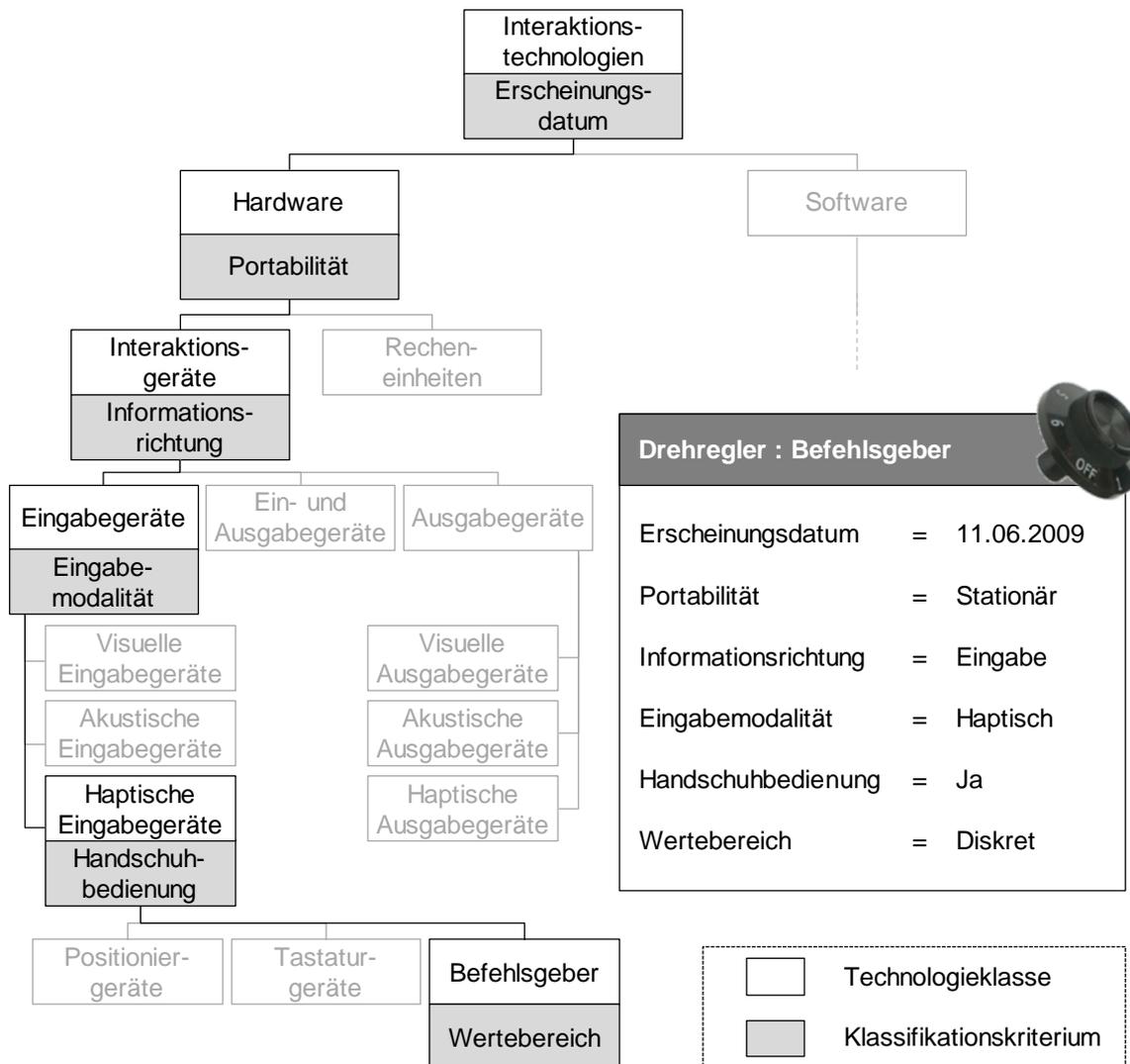


Abbildung 5-15: Exemplarische Klassifikation eines Drehreglers

Gegenüberstellung in einer Einflussmatrix

Wie bereits zu Beginn von Abschnitt 5.3.1 dargestellt, ermöglichen die Auswahl-faktoren Rückschlüsse auf die einzusetzende Interaktionstechnologie. Im Rahmen dieser Arbeit wurde der Ansatz gewählt, anhand der Ausprägungen der Auswahl-faktoren geeignete Ausprägungen auf Seite der Klassifikationskriterien zu identifizieren. Als Beispiel ist der Bewegungsradius eines Benutzers zu nennen, der u. a. großen Einfluss darauf hat, ob eine mobile oder stationäre Interaktionstechnologie einzusetzen ist. Um diese und alle weiteren Abhängigkeiten zwischen Auswahl-faktoren und Klassifikationskriterien zu formalisieren und somit eine automatisierte Bestimmung geeigneter Klassifikationskriterien für einen konkreten Anwendungs-fall zu ermöglichen, wurden diese in einer Einflussmatrix gegenübergestellt, deren wesentliche Elemente nachfolgend vorgestellt werden.

Generell werden in Einflussmatrizen, wie bspw. bei AUMAYR (2016, S. 232 ff.), die Einflussfaktoren in den Zeilen und die beeinflussten Größen in den Spalten geführt. Im vorliegenden Anwendungsfall bedeutet dies, dass die Ausprägungen der Auswahlfaktoren in den Zeilen und die Ausprägungen der Klassifikationskriterien in den Spalten aufgenommen werden. Abbildung 5-16 visualisiert den grundsätzlichen Aufbau der erarbeiteten Einflussmatrix, in der zur besseren Einordnung auch die Auswahlfaktoren mit ihren Gewichtungen und zugehörigen Stakeholdern sowie die Klassifikationskriterien mit ihren zugehörigen Technologieklassen dargestellt sind. In der resultierenden Einflussmatrix sind dabei nur Auswahlfaktoren berücksichtigt, deren Ausprägungen mindestens ein Klassifikationskriterium beeinflussen. Analog werden nur Klassifikationskriterien betrachtet, deren Ausprägungen von mindestens einem Auswahlfaktor beeinflusst werden.

				Technologieklassen							
								Klassifikationskriterien			
								Ausprägungen			
Stakeholder	Gewichtungen	Auswahlfaktoren	Ausprägungen								
				Einflussbewertungen							

Abbildung 5-16: Aufbau der Einflussmatrix

Um weiterhin eine quantifizierbare Einflussbewertung der Ausprägungen eines Auswahlfaktors auf die Ausprägungen eines Klassifikationskriteriums vornehmen zu können, wird ein Schema genutzt, das generell zwischen Eignungsbedingungen und Eignungsgraden differenziert (vgl. Abbildung 5-17). *Eignungsbedingungen* kommen zum Einsatz, wenn die Ausprägung eines Auswahlfaktors den Rückschluss ermöglicht, dass eine einzusetzende Interaktionstechnologie eine bestimmte Ausprägung eines Klassifikationskriteriums zwingend aufweisen muss (JA) oder dass eine Technologie diese Ausprägung keinesfalls aufweisen darf (NEIN). Im Beispiel der Lautstärke des Produktionsumfelds aus Abbildung 5-17 tritt dies bei der

Ausprägung „laute Hintergrundgeräusche“ auf, wodurch eine schallempfindliche Technologie ausgeschlossen wird. Darüber hinaus treten Eignungsbedingungen bei Auswahlfaktoren auf, deren Ausprägungen über numerische Eingaben bestimmt werden, wie bspw. bei der Umgebungstemperatur. Die Eignungsbedingung gibt hierbei an, ob die Ausprägung eines Klassifikationskriteriums, wie bspw. die Hitzebeständigkeit, größer-gleich (\geq) bzw. kleiner-gleich (\leq) der Ausprägung des Auswahlfaktors sein soll. Schließlich werden Eignungsbedingungen auch eingesetzt, wenn die Ausprägungen eines Auswahlfaktors und eines Klassifikationskriteriums gleich sein sollen ($=$). Dieser Fall tritt in der erarbeiteten Einflussmatrix nur beim Abgleich des Herstellers mit einem vorgegebenen Lieferanten auf.

Demgegenüber stellen *Eignungsgrade* Abhängigkeiten zwischen Ausprägungen von Auswahlfaktoren und Klassifikationskriterien dar, die zwar ebenfalls Rückschlüsse hinsichtlich der Eignung einer Interaktionstechnologie ermöglichen, aber nicht die direkte Notwendigkeit oder den zwingenden Ausschluss einer Ausprägung eines Klassifikationskriteriums zur Folge haben. Dazu kommen insgesamt fünf unterschiedliche Grade der Beeinflussung zur Anwendung, die von einem besonders negativen Einfluss (--) über entsprechende Abstufungen bis zu einem besonders positiven Einfluss (++) reichen. In Abbildung 5-17 kommen Eignungsgrade bspw. zum Einsatz, um den negativen Einfluss normaler Hintergrundgeräusche auf eine hohe Schallempfindlichkeit eines Interaktionsgeräts zum Ausdruck zu bringen.

Bewertungsschema										Interaktionsgeräte		
Eignungsbedingungen					Eignungsgrade					Schallempfindlichkeit		
JA	NEIN	\geq	\leq	=	--	-	0	+	++	Hoch	Bedingt	Keine
Umfeld/Umwelt	Gewichtung	Lautstärke im Produktionsumfeld			Geringe Hintergrundgeräusche					0	0	0
					Normale Hintergrundgeräusche					-	0	+
					Laute Hintergrundgeräusche					NEIN	--	++

Abbildung 5-17: Bewertungsschema der Einflussmatrix mit Beispiel

Um die erarbeitete Einflussmatrix bei der Auswahl einer Interaktionstechnologie für einen konkreten Anwendungsfall einsetzen zu können, bedarf es schließlich eines Formalismus zur Auswertung der Einzelbewertungen. Grundsätzlich werden dafür die Spaltensummen aller Ausprägungen der Klassifikationskriterien gebildet, wobei

allerdings nur diejenigen Zeilen und somit Ausprägungen der Auswahlfaktoren berücksichtigt werden, die für den vorliegenden Anwendungsfall zutreffen. Da weder Eignungsbedingungen noch Eignungsgrade über numerische Werte definiert sind, bedarf es zur Bildung der Spaltensummen folgender Berechnungsvorschrift: Sobald mindestens eine relevante Zeile eine Eignungsbedingung enthält, bildet diese die Spaltensumme. Andernfalls werden die einzelnen Eignungsgrade in die Zahlenwerte von -2 bis +2 überführt und aufsummiert. Bei dieser Summenbildung sind allerdings noch die Gewichtungen der einzelnen Auswahlfaktoren (von 0 bis 3) zu berücksichtigen, die mit den jeweiligen Eignungsgraden vor der Summation multipliziert werden.¹¹ Im nachfolgenden Abschnitt 5.3.2 wird diese Vorschrift sowie das generelle Vorgehen zur Verwendung der Einflussmatrix und zur Interpretation der Ergebnisse anhand eines Beispiels noch detailliert erläutert.

5.3.2 Prozessablauf für den Einsatz der Einflussmatrix bei der Technologieauswahl

Um mit der erarbeiteten Einflussmatrix die Technologieauswahl für einen konkreten Anwendungsfall unterstützen zu können, wird die nachfolgend beschriebene, dreistufige Vorgehensweise vorgeschlagen. Diese sieht zunächst eine *Gewichtung der Auswahlfaktoren* vor. Darauf aufbauend erfolgt die *Einordnung des Anwendungsfalls*, indem die jeweils zutreffenden Ausprägungen der Auswahlfaktoren bestimmt werden. Über die Einflussmatrix erhält man als Ergebnis geeignete Ausprägungen für die einzelnen Klassifikationskriterien. Diese werden in der *Auswertung der Ergebnisse* verwendet, um eine passende Interaktionstechnologie für den jeweils vorliegenden Anwendungsfall zu identifizieren. In den folgenden Abschnitten werden diese Schritte im Detail beschrieben.

Gewichtung der Auswahlfaktoren

Um Anwendern der erarbeiteten Einflussmatrix eine Möglichkeit zu geben, die spezifischen Rahmenbedingungen eines konkreten Anwendungsfalls bestmöglich zu berücksichtigen, können die einzelnen Auswahlfaktoren in einem ersten Schritt anhand der Abstufungen von 0 bis 3 gewichtet werden. Dies kann z. B. erforderlich sein, wenn für einen konkreten Anwendungsfall monetäre Zielgrößen des Einkaufs nur eine untergeordnete Rolle einnehmen, da sich ein Unternehmen primär als

¹¹ Hinsichtlich der Eignungsbedingungen werden die Gewichtungen der Auswahlfaktoren insofern berücksichtigt, dass irrelevante Auswahlfaktoren (= Gewichtung von Null) nicht in die Berechnungsvorschrift eingehen. Zwischen den unterschiedlichen Gewichtungswerten (1-3) wird aufgrund der generellen Kritikalität einer Eignungsbedingung nicht weiter differenziert.

Innovationsführer etablieren möchte. Auch etwaigen Unsicherheiten bei der Einordnung des Anwendungsfalls hinsichtlich einzelner Auswahlfaktoren, wie z. B. im Bereich „Umfeld/Umwelt“, kann mit einer niedrigen Gewichtung dieser Faktoren begegnet werden. Wichtig ist bei der Vergabe der Gewichtungen allerdings eine ganzheitliche Betrachtung aller Stakeholder, um nicht Auswahlfaktoren von Stakeholdern zu vernachlässigen, die an der Durchführung der Gewichtung nicht beteiligt sind. Vor diesem Hintergrund wird an dieser Stelle die Einbeziehung von Experten mit Kenntnis über die erarbeitete Einflussmatrix vorgeschlagen, was sich insbesondere bei der Evaluation des Ansatzes (vgl. Kapitel 6) als zielführend herausgestellt hat. Neben der Berücksichtigung aller Stakeholder können diese Experten auch bestehende Abhängigkeiten zwischen Auswahlfaktoren in die Gewichtung einbeziehen.¹² Auch bei der Bestimmung der zutreffenden Ausprägungen von einzelnen Auswahlfaktoren kann auf diese Weise Fehlinterpretationen vorgebeugt werden.

Einordnung des Anwendungsfalls

Auf Basis der durchgeführten Gewichtung erfolgt in einem zweiten Schritt die Einordnung des Anwendungsfalls. Je nach Art des Auswahlfaktors werden dabei entweder die zutreffenden Ausprägungen ausgewählt (Einfach- bzw. Mehrfachauswahl) oder es wird der konkrete Wert des Auswahlfaktors eingetragen (Freitexteingabe). Abbildung 5-18 verdeutlicht dieses Vorgehen anhand eines fiktiven Anwendungsfalls, der anhand eines exemplarischen Ausschnitts der Einflussmatrix eingeordnet wird. Durch das Klassifikationskriterium „Portabilität“ wird das in Abschnitt 5.3.1 vorgestellte Berechnungsschema für Eignungsgrade verdeutlicht, für das in dem Beispiel nur die Ausprägung „Mobil“ einen positiven Wert besitzt und somit die einzig geeignete Ausprägung darstellt. Anhand des Kriteriums „Latenzzeit“ wird demgegenüber deutlich, dass auch mehrere Ausprägungen einen positiven Wert besitzen und somit für einen Anwendungsfall geeignet sein können, wobei die Zahlenwerte weiterhin einen Vergleich zwischen den jeweiligen Ausprägungen ermöglichen. Ferner wird anhand der Kriterien „Informationsrichtung“ und „Hitzebeständigkeit“ die Funktion der Eignungsbedingungen deutlich, durch die einerseits die Verwendung von Ausgabegeräten ausgeschlossen und andererseits ein Grenzwert für die Hitzebeständigkeit definiert wird. In Summe ergibt sich eine Menge an geeigneten und notwendigen Ausprägungen, die eine Interaktionstechnologie aufweisen sollte, um eine zielführende Bedienung einer Maschine bzw. Anlage zu ermöglichen.

¹² Ein Beispiel hierfür sind die Auswahlfaktoren „Bewegungsradius“ und „Interaktionshäufigkeit“, da der Bewegungsradius allgemein einen deutlich größeren Einfluss auf bspw. die Portabilität besitzt.

		zutreffende Ausprägung des Anwendungsfalls	Informations- richtung			Portabilität		Latenzzeit			Hitze- bestand.
			Eingabe	Ausgabe	Ein- und Ausgabe	Mobil	Stationär	Keine Echtzeit	Weiche Echtzeit	Harte Echtzeit	„Freitext- eingabe“
2	Interaktionsart	Beobachten	NEIN	++	--						
		Bedienen	++	NEIN	--						
		Beob. & Bed.	+	+	++						
3	Bewegungs- radius der Bediener	< 1m				-	+				
		1 - 10m				0	0				
		10 - 100m				+	-				
		> 100m				++	--				
1	Interaktions- häufigkeit	< 10 p. Tag				-	+	0	0	0	
		10 - 50 p. Tag				0	0	-	0	+	
		> 50 p. Tag				+	-	--	+	++	
1	Geforderte Datensicherheit	Gering				0	0				
		Mittel				-	+				
		Hoch				--	++				
2	Temperatur im Fabrikumfeld	„60 °C“									≥
Auswertung			4	NEIN	-4	3	-3	-2	1	2	≥ 60 °C

Abbildung 5-18: Beispielhafte Einordnung eines fiktiven Anwendungsfalls

Abbildung 5-18 macht deutlich, dass mit der erarbeiteten Einflussmatrix lediglich ein allgemeingültiger Vorschlag für geeignete bzw. notwendige Ausprägungen von Klassifikationskriterien für die Bedienung einer gesamten Maschine bzw. Anlage gegeben wird. Das Konzept berücksichtigt somit insbesondere nicht den gerade bei größeren Anlagen praxisrelevanten Anwendungsfall, dass mehrere, teilweise auch unterschiedliche Interaktionstechnologien für spezifische Aufgaben zum Einsatz kommen können. Um das erarbeitete Konzept auch hier nutzen zu können, bedarf es einer initialen Unterteilung des Anwendungsfalls in einzelne Bereiche, für die ein Anwender den Einsatz unterschiedlicher Interaktionstechnologien vorschlägt. Für jeden dieser Bereiche können anschließend die geeigneten Ausprägungen der Auswahlfaktoren identifiziert und entsprechend fundiertere Ergebnisse für die Ausprägungen der Klassifikationskriterien generiert werden. Signifikante Unterschiede zwischen einzelnen Bereichen können sich dabei bspw. bei Kriterien wie der Interaktionsart oder der Interaktionshäufigkeit ergeben.

Auswertung der Ergebnisse

Mit der für einen konkreten Anwendungsfall erhaltenen Menge an notwendigen und geeigneten Ausprägungen der einzelnen Klassifikationskriterien kann in einem letzten Schritt ein Abgleich mit am Markt verfügbaren Interaktionstechnologien durchgeführt werden. Ziel ist die Identifikation einer technologischen Lösung, die möglichst viele der notwendigen und geeigneten Ausprägungen aufweist. Dabei sollten zunächst die Eignungsbedingungen betrachtet werden, um den Lösungsraum potenzieller Technologien einzugrenzen. Daraufhin kann in einem zweiten Schritt analysiert werden, welche der verbliebenen Technologien möglichst viele Ausprägungen der Klassifikationskriterien mit einem positiven Eignungsgrad aufweisen. Weiterhin bietet sich bei der Analyse der einzelnen Klassifikationskriterien ein Top-Down-Vorgehen hinsichtlich der einzelnen Technologieklassen (vgl. Abbildung 5-13) an. Auf diese Weise kann bspw. sichergestellt werden, dass nicht unterschiedliche Befehlsgeber hinsichtlich geeigneter Wertebereiche analysiert werden, wenn bereits auf der Ebene der Interaktionsgeräte der Einsatz von haptischen Interaktionstechnologien generell ausgeschlossen wird.

Analog zu den beiden vorangegangenen Schritten wird auch bei der Auswertung der Ergebnisse eine Durchführung mit HMI-Experten vorgeschlagen. Dies ist insbesondere vor dem Hintergrund notwendig, dass Maschinen- und Anlagenbauer oft nur eine begrenzte Anzahl an möglichen Interaktionstechnologien kennen und nur ein geringes Wissen hinsichtlich der tatsächlichen Fähigkeiten dieser Technologien haben. Hierdurch könnten letztlich Fehlinterpretationen bezüglich der Eignung einer technischen Lösung für einen konkreten Anwendungsfall entstehen. Zukünftig besteht allerdings die Möglichkeit, die Auswertung der Ergebnisse automatisiert durchzuführen, sofern ein breites Spektrum an Technologien unterschiedlicher Hersteller unter Verwendung der zuvor definierten Klassifikationskriterien in einer Datenbank bereitgestellt werden kann.

Schließlich ist zu erwähnen, dass der im Rahmen dieser Arbeit entwickelte Ansatz durch die unterschiedlichen Stakeholder (vgl. Abbildung 5-11) zwar bereits vielfältige Auswahlfaktoren berücksichtigt, aber dennoch nicht alleinstehend für die Technologieauswahl eines konkreten Anwendungsfalls herangezogen werden sollte. Vielmehr wird eine kombinierte Verwendung dieses Ansatzes mit weiterführenden Methoden, wie bspw. einer Wirtschaftlichkeitsbetrachtung oder einer Bewertung der Technologiereife (vgl. OEHMEN 2016, S. 89 ff.) vorgeschlagen, die aufgrund existierender Vorarbeiten selbst nicht im Fokus der vorliegenden Arbeit stehen.

5.4 Aktivität III: HMI-Implementierung

Mit dem Modell einer Mensch-Maschine-Schnittstelle sowie einer ausgewählten Technologie erfolgt im Rahmen der Verfeinerung die HMI-Implementierung.¹³ Der ursprüngliche IMoMeSA-Ansatz sieht für diese Phase zunächst eine Transformation der einzelnen Modellinhalte vor, die dann in den jeweiligen Disziplinen konkretisiert werden können. Das spezifische Konzept einer solchen Modelltransformation für HMIs wird nachfolgend in Abschnitt 5.4.1 vorgestellt.

Auf Basis der Modelltransformation kann die Entwicklung in den einzelnen Disziplinen weitestgehend losgelöst voneinander und mit spezifischen Herangehensweisen und Werkzeugen durchgeführt werden. Da sich im Hinblick auf die Mensch-Maschine-Schnittstelle bereits zahlreiche Ansätze etabliert haben (vgl. Abschnitt 3.4) und die Entwicklung speziell auf die ausgewählte technologische Lösung ausgerichtet ist, wurde im Rahmen dieser Arbeit keine eigene Herangehensweise für die HMI-Implementierung erarbeitet. Da allerdings auch in der Verfeinerung eine regelmäßige Synchronisation der einzelnen Disziplinen anhand des Systemmodells notwendig ist, wird in Abschnitt 5.4.2 ein Prozessablauf für die HMI-Implementierung beschrieben, der insbesondere die Verwendung des Systemmodells und der Modelltransformation während der Verfeinerung festlegt.

5.4.1 Transformation der Modelle von Mensch-Maschine-Schnittstellen

Das grundsätzliche Ziel der Transformation eines HMI-Modells in die jeweiligen Entwicklungsumgebungen einer Mensch-Maschine-Schnittstelle liegt darin, bereits auf Modellebene spezifizierte Inhalte zu übernehmen und dadurch den Aufwand in der Implementierung zu reduzieren. Im Folgenden wird zuerst ein Gesamtüberblick gegeben, welche Inhalte aus dem mechatronischen Konzept für eine solche Transformation relevant sind und wie diese Inhalte in der jeweiligen Zielumgebung abgebildet werden. Anschließend werden die notwendigen Transformationsschritte im Detail vorgestellt, wobei vor allem methodische Aspekte der HMI-Modelltransformation im Fokus stehen. Die konkreten Vorschriften für die Transformation hängen hingegen von der gewählten Technologie sowie der Entwicklungsumgebung ab und müssen somit spezifisch für diese implementiert werden. Eine spezielle Transformationsvorschrift wird im Rahmen dieser Arbeit bei der Anwendung des Ansatzes an einem Fallbeispiel in Abschnitt 6.3 vorgestellt.

¹³ Diese Aktivität fokussiert somit Interaktionselemente, die mit einer Software realisiert werden (GUIs, etc.). Einfache Interaktionselemente, wie Tasten oder Lampen, sind bereits im HMI-Modell hinreichend spezifiziert und werden beim physikalischen Aufbau gemäß dem mechatronischen Konzept direkt umgesetzt.

Gesamtüberblick über die Modelltransformation

Wie bereits bei der HMI-Modellierung verdeutlicht, können zur Umsetzung einer Mensch-Maschine-Schnittstelle mehrere Interaktionselemente eingesetzt werden. Diese werden in der Regel mit unterschiedlichen Softwarelösungen und somit in verschiedenen Entwicklungsumgebungen implementiert. Dieser Aspekt ist bei der Modelltransformation zu berücksichtigen, weswegen zunächst entschieden wird, in welche Entwicklungsumgebungen die einzelnen Interaktionselemente transformiert werden sollen (vgl. Abbildung 5-19). Um in einer Entwicklungsumgebung ein initiales HMI-Programm zu generieren, werden die einem Interaktionselement zugeordneten *Benutzerrollen*, *HMI-Ports*, *HMI-Komponenten*, *Verhalten* und die *Screen-Repräsentationen* des Interaktionselements selbst benötigt. Ein letztes Element bilden die *Datenports* von HMI-Komponenten, da diese die Schnittstelle zur Maschinen- bzw. Anlagensteuerung abbilden. Die Zuordenbarkeit dieser Elemente zu einem Interaktionselement ist durch die Verknüpfung der Screen-Repräsentationen mit den HMI-Ports gegeben, über die sich weiterhin der Bezug zu den Benutzerrollen, den HMI-Komponenten und den enthaltenen Verhalten ergibt.

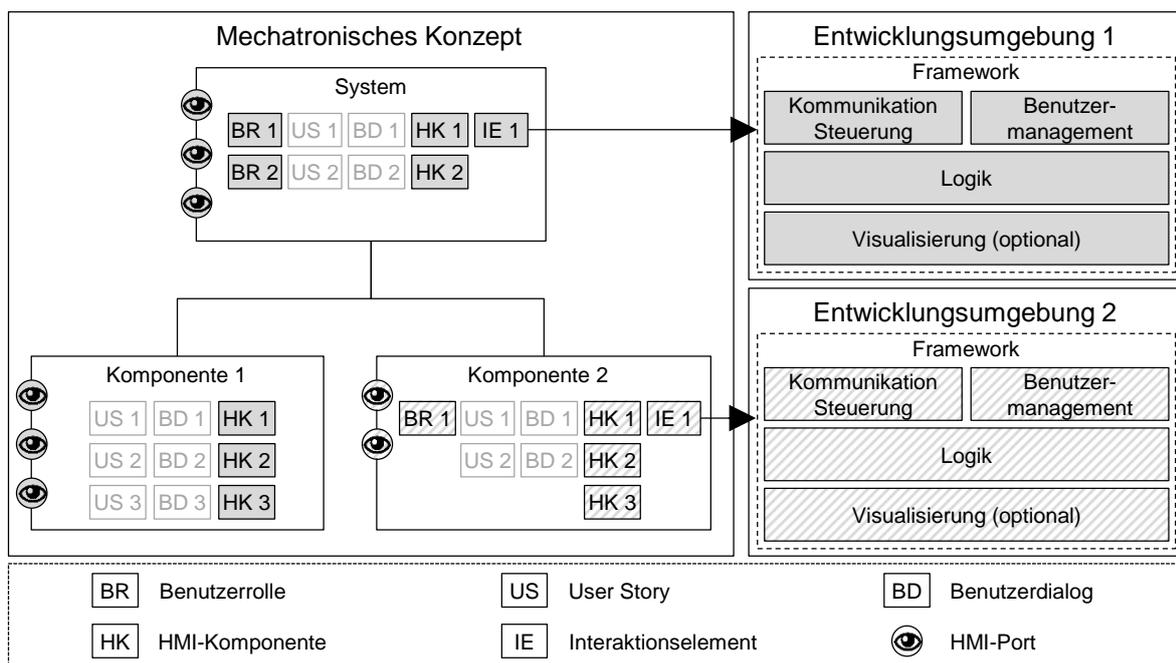


Abbildung 5-19: Gesamtüberblick über die Modelltransformation

Auf Seite der jeweiligen Zielumgebung werden diese Elemente verwendet, um für die in Abbildung 5-19 rechts dargestellten Bausteine ein initiales Programm zu generieren. Diese umfassen zunächst einen Baustein zur *Kommunikation mit der Steuerung* der Maschine bzw. Anlage über die auf Modellebene mit Datenports

spezifizierte Schnittstelle. Demgegenüber werden die modellierten Benutzerrollen sowie deren Verknüpfungen mit HMI-Ports verwendet, um ein initiales *Benutzermanagement* in der jeweiligen Zielumgebung zu realisieren. Des Weiteren können die HMI-Komponenten mit ihrem integrierten Verhalten genutzt werden, um ein Programm für die *Logik* einer Mensch-Maschine-Schnittstelle bereitzustellen. Einen weiteren Baustein bildet die *Visualisierung*, die allerdings nur generiert wird, wenn als Interaktionstechnologie ein visuelles bzw. graphisches Ausgabegerät ausgewählt wurde.¹⁴ Dabei dienen die Screen-Repräsentationen eines Interaktionselements als Ausgangspunkt, um die einzelnen Ansichten der Mensch-Maschine-Schnittstelle zu generieren. Einen letzten Baustein bildet ein umgebendes *Framework*, welches für die Kommunikation zwischen den eben skizzierten Bausteinen sowie die generelle Ablauffähigkeit der Mensch-Maschine-Schnittstelle zuständig ist. Dadurch soll insbesondere eine frühzeitige Qualitätssicherung des HMIs während der Verfeinerung ermöglicht werden (vgl. Abschnitt 5.5).

Bevor nachfolgend eine Detailbetrachtung der einzelnen Transformationsschritte erfolgt, ist zu erwähnen, dass die eben eingeführten Bausteine und ihre modulare Strukturierung mit der Trennung von Logik und Visualisierung nur eine Grundkonfiguration für die HMI-Implementierung bilden. Während der Verfeinerung ist diese Struktur um zusätzliche Bausteine, wie bspw. für ein Logging, zu erweitern, für die im mechatronischen Konzept keine relevanten Informationen vorhanden sind und die deswegen nicht Gegenstand der Modelltransformation sind.

Detailbetrachtung der einzelnen Transformationsschritte

Die zentrale Aufgabe des im letzten Abschnitt eingeführten Bausteins zur *Kommunikation mit der Steuerung* liegt in der Verwaltung und dem Austausch der Schnittstellenvariablen, über welche die Kommunikation zwischen Interaktionsgerät und Maschinen- bzw. Anlagensteuerung stattfindet. Diese Variablen sind bereits auf der Ebene des mechatronischen Konzepts als Datenports der HMI-Komponenten (vgl. Abbildung 5-5) spezifiziert. Da auf Modellebene auch der Typ der Datenports festgelegt wird (Boolean, Integer, etc.), kann eine vollständige Variablenliste in der Zielumgebung generiert werden, wobei die Gliederung dieser Liste anhand der Komponentenstruktur des mechatronischen Konzepts erfolgt. Neben der Verwaltung der Variablenliste liegt eine zweite Aufgabe in der Realisierung der eigentlichen Kommunikation mit der Steuerung, um einerseits den momentanen

¹⁴ Diese Einschränkung resultiert aus der Tatsache, dass auf Modellebene ausschließlich Möglichkeiten zur Abbildung graphischer Mensch-Maschine-Schnittstellen angeboten werden.

Zustand der Maschine auf dem Interaktionsgerät anzuzeigen und andererseits die Nutzereingaben an die Maschine zu übermitteln. Da der hierzu benötigte Softwarebaustein vom eingesetzten Kommunikationsprotokoll abhängt und meist in Form von Bibliotheken in der jeweiligen Zielumgebung bereitgestellt wird, kann an dieser Stelle kein allgemeingültiger Ansatz für eine automatische Generierung aus dem mechatronischen Konzept angegeben werden. Stattdessen wird eine spezielle Ausprägung des Transformationsschritts bei der Anwendung der Entwicklungsmethodik in Abschnitt 6.3 beschrieben. Weiterhin ist für die eben beschriebene Kommunikation auch auf Steuerungsseite ein entsprechender Baustein notwendig. Dafür ist der vorhandene Modelltransformator zur Generierung von Steuerungscode (vgl. HACKENBERG ET AL. 2015, S. 51) zu erweitern, wobei auch hier die Datenports der HMI-Komponenten für die Modelltransformation verwendet werden.

Ein zweiter Baustein, der auf Basis des mechatronischen Konzepts initial generiert werden kann, ist das *Benutzermanagement*. Dieses kommt bei Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau fast immer zum Einsatz und ist einerseits für die An- und Abmeldung der einzelnen Benutzer sowie andererseits für die Verwaltung der Rechte einzelner Benutzer im Umgang mit der Maschine bzw. Anlage zuständig. Für die Generierung dieses Bausteins können die im mechatronischen Konzept abgebildeten Benutzerrollen sowie deren Zuordnung zu den einzelnen HMI-Ports verwendet werden. Die Benutzerrollen können dabei direkt anhand ihrer Namen in die jeweilige Zielumgebung übernommen und dort als Liste verwaltet werden. Für die An- und Abmeldung der Nutzer wird in Abhängigkeit der Entwicklungsumgebung ein Programmcode generiert, um eine Benutzerrolle an der Mensch-Maschine-Schnittstelle auszuwählen und deren An- bzw. Abmeldung zu bestätigen. Dieser Code kann während der Verfeinerung entsprechend erweitert werden, wie etwa um eine passwortgeschützte Anmeldung. Für die Verwaltung der Rechte kann die auf Modellebene vorgesehene Verknüpfung von HMI-Ports mit Benutzerrollen genutzt werden. Diese Verknüpfung kann in den Baustein für das Benutzermanagement dahingehend überführt werden, ob einem Benutzer eine Information zur Verfügung steht bzw. ob er eine Aktion durchführen darf.¹⁵ Bei visuellen Ausgabegeräten wird dies bspw. durch die Sichtbarkeit der Bedien- und Anzeigeelemente bestimmt, die in den meisten Entwicklungsumgebungen als Standard-Eigenschaft vorgesehen ist und je nach angemeldetem Nutzer auf den Wert „sichtbar“ oder „unsichtbar“ festgelegt wird.

¹⁵ Dabei ist insbesondere das auf Modellebene eingeführte Vererbungsprinzip von Benutzerrollen zu berücksichtigen, durch das einer abgeleiteten Nutzerrolle auch die Rechte der übergeordneten Rollen zuteil werden.

Den dritten Softwarebaustein, der sich auf Basis der Inhalte des mechatronischen Konzepts generieren lässt, bildet die *Logik* einer Mensch-Maschine-Schnittstelle. Die grundsätzlichen Aufgaben dieses Bausteins liegen in der Verarbeitung der von Benutzern getätigten Eingaben, der Bereitstellung der Ausgaben an dem Ausgabegerät sowie das Auslesen und Beschreiben der Schnittstellenvariablen des zuvor eingeführten Bausteins zur Kommunikation mit der Steuerung. Für diese Aufgaben stellen auf Modellebene HMI-Komponenten mit ihrem integrierten Verhalten bereits eine vereinfachte Lösung bereit und können deswegen als Basis für die Modelltransformation verwendet werden. Bei der Transformation selbst werden zuerst die für eine Zielumgebung relevanten HMI-Komponenten bestimmt. Für diese werden jeweils eigene, voneinander gekapselte Teilbausteine¹⁶ innerhalb des Logik-Bausteins generiert und anhand der mechatronischen Komponentenstruktur gegliedert. Weiterhin werden diesen Teilbausteinen Ein- und Ausgangsvariablen zugeordnet, die auf Modellebene den HMI- und Datenports einer HMI-Komponente entsprechen. Für letztere können in der Zielumgebung die im Baustein zur Kommunikation mit der Steuerung definierten Schnittstellenvariablen genutzt werden. In einem finalen Schritt wird die auf Modellebene über diskrete Zustandsautomaten dargestellte Logik in die generierten Teilbausteine integriert. In Abhängigkeit von der Entwicklungsumgebung ist der Zustandsautomat entweder in der jeweiligen Programmiersprache nachzubilden oder wird direkt in der Entwicklungsumgebung abgebildet. Ein Beispiel für die erste Variante wird bei der Anwendung des Ansatzes in Abschnitt 6.3 noch im Detail vorgestellt.

Für den Fall, dass ein Interaktionselement über ein visuelles Ausgabegerät realisiert wird, kann weiterhin der Softwarebaustein *Visualisierung* aus dem mechatronischen Konzept generiert werden. In diesem werden die einzelnen Ansichten der Mensch-Maschine-Schnittstelle mit den jeweiligen Bedien- und Anzeigeelementen sowie das zugehörige Navigationskonzept zum Wechsel zwischen diesen Ansichten verwaltet. Für beide Aspekte bieten Screen-Repräsentationen eines Interaktionselements eine vereinfachte Lösung und können daher als Basis für die Modelltransformation verwendet werden. Bei der Transformation wird für jede Screen-Repräsentation eine einzelne Ansicht der Mensch-Maschine-Schnittstelle generiert, wobei auch die jeweiligen Bedien- und Anzeigeelemente in der Zielumgebung generiert werden. Des Weiteren wird die im mechatronischen Konzept vorhandene Verknüpfung eines Bedien- oder Anzeigeelements mit einem HMI-Port genutzt, um die zugehörige Ein- und Ausgangsvariable des zuvor eingeführten Logik-Bausteins

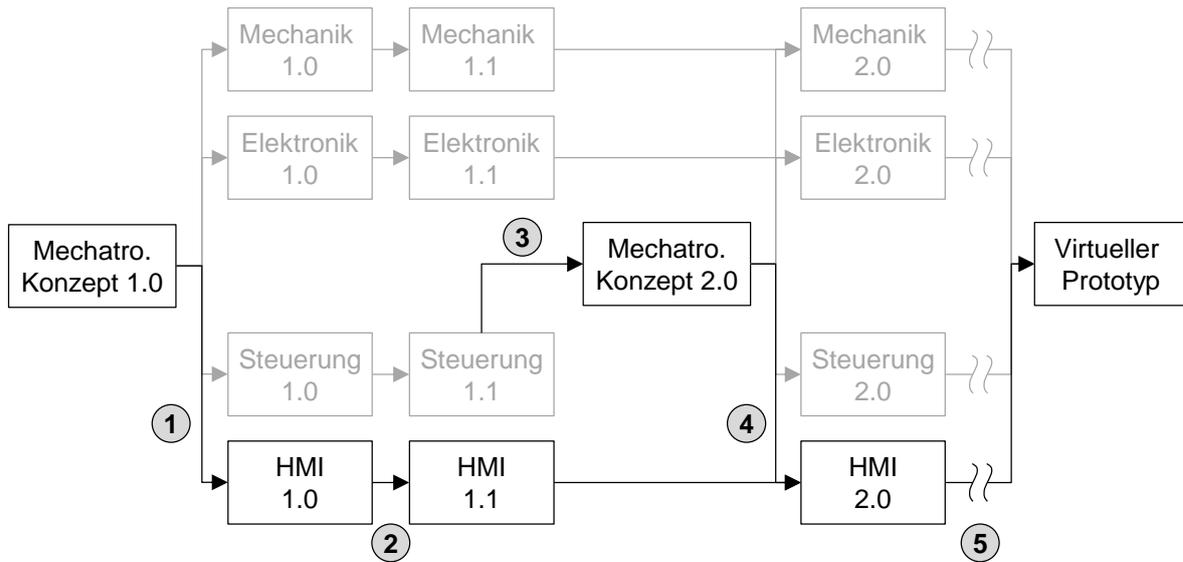
¹⁶ z. B. Methoden in der objektorientierten Programmierung

zu beschreiben bzw. auszulesen. Durch dieses Konzept werden die Abhängigkeiten zwischen Logik und Darstellung minimiert, um die bereits auf Modellebene vorgesehene Trennung dieser Bereiche auch in der Zielumgebung zu gewährleisten. Im letzten Schritt wird das Navigationskonzept in die Zielumgebung überführt. Dazu werden die auf Modellebene definierten Navigationselemente als reguläre Bedienelemente auf den einzelnen Ansichten der Mensch-Maschine-Schnittstelle angeordnet. Zudem wird eine entsprechende Logik generiert, die beschreibt, welche Ansicht beim Betätigen eines Navigationselements aufgerufen werden soll.

In einem letzten Schritt der Modelltransformation werden die eben skizzierten Bausteine durch ein umgebendes *Framework* miteinander verbunden. Für dieses stehen zwar auf Modellebene keine Informationen zur Verfügung, trotzdem ist die automatische Generierung eines Frameworks von großer Bedeutung, um bereits zu Beginn der Verfeinerung eine ausführbare Mensch-Maschine-Schnittstelle in der Qualitätssicherung nutzen zu können (vgl. Abschnitt 5.5). Dazu ist das Framework u. a. für die Instanziierung und Verknüpfung der einzelnen Bausteine sowie die Einbindung von Bibliotheken verantwortlich. Allerdings sind der Aufbau und der Inhalt des Frameworks abhängig von der gewählten Entwicklungsumgebung, weswegen an dieser Stelle kein generisches Konzept für die Modelltransformation dargestellt werden kann. Stattdessen wird im weiteren Verlauf der Arbeit eine spezifische Implementierung eines Frameworks bei der Anwendung des Ansatzes an einem Fallbeispiel (Abschnitt 6.3) vorgestellt.

5.4.2 Prozessablauf für den Einsatz der Modelltransformation während der Implementierung von Mensch-Maschine-Schnittstellen

Die eben eingeführte Modelltransformation bildet bei der HMI-Implementierung den initialen Schritt, um relevante Informationen des mechatronischen Konzepts wiederzuverwenden und in ein entsprechendes Programm zu überführen. Aufgrund der beschränkten Möglichkeiten zur Abbildung von HMIs auf Modellebene kann dieses Programm allerdings in der Regel lediglich als Basis verwendet werden und ist während der Verfeinerung weiter auszuarbeiten. Die diesbezüglich vorgesehene Herangehensweise bettet sich in den ursprünglichen IMoMeSA-Ansatz (vgl. HACKENBERG ET AL. 2015, S. 53 f.) sowie den Ansatz von GAUSEMEIER ET AL. (2014, S. 186 f.) ein und erweitert diese Konzepte um die Betrachtungsebene von HMIs. Abbildung 5-20 visualisiert den resultierenden Ablauf für die Implementierung einer Mensch-Maschine-Schnittstelle während der Verfeinerung von mechatronischen Systemen, der nachfolgend im Detail vorgestellt wird.



- ① Initiale HMI-Modelltransformation
- ② Disziplinspezifische HMI-Implementierung
- ③ Anpassung des Systemmodells aufgrund einer interdisziplinär relevanten Änderung
- ④ Erneute HMI-Modelltransformation bei Beachtung der bisherigen Weiterentwicklungen
- ⑤ Weiterführende HMI-Implementierung (iterativer Ablauf der Schritte 2-4)

Abbildung 5-20: Ablauf der HMI-Implementierung während der Verfeinerung von mechatronischen Systemen (in Anlehnung an GAUSEMEIER ET AL. 2014, S. 187)

Initiale HMI-Transformation

Der erste Schritt des erarbeiteten Prozessablaufs adressiert die initiale Generierung der HMI-Programme in den Entwicklungsumgebungen der einzelnen Interaktionselemente gemäß der in Abschnitt 5.4.1 eingeführten Systematik. Dabei werden auf Basis des Systemmodells, das zum Ende der Konzeption vorliegt (*Mechatronisches Konzept 1.0*), die einzelnen HMI-Programme generiert (*HMI 1.0*), welche dann in den Zielumgebungen weiterführend ausgearbeitet werden können.

Disziplinspezifische HMI-Implementierung

Durch den interdisziplinären Aufbau eines mechatronischen Konzepts in der Phase der Konzeption und das hierdurch geschaffene gemeinsame Systemverständnis kann die Umsetzung in den einzelnen Entwicklungsdisziplinen weitestgehend losgelöst voneinander und mit disziplinspezifischen Vorgehensweisen durchgeführt werden. Aus HMI-Sicht können die in Abschnitt 3.4 beschriebenen Ansätze genutzt werden, um die Mensch-Maschine-Schnittstelle weiterführend zu implementieren (*HMI 1.1*). Allerdings wird an dieser Stelle kein konkretes Vorgehensmodell

vorgeschlagen, da dessen Auswahl stark von den Rahmenbedingungen des Anwendungsfalls, wie bspw. der Qualifikation und Größe des Entwicklungsteams, abhängt. Generell sei aber auf agile Entwicklungsansätze wie Scrum verwiesen, die sich in der Softwareentwicklung in den letzten Jahren etabliert haben. Deren Einsatz ist bei der HMI-Implementierung in der Regel zielführend, um in einem iterativen Vorgehen die einzelnen Funktionen der Mensch-Maschine-Schnittstelle zu realisieren und somit frühzeitige Nutzerevaluationen zu ermöglichen. Zur Strukturierung der einzelnen Entwicklungszyklen können dabei die im Systemmodell modellierten User Stories genutzt werden.

Anpassung des Systemmodells aufgrund einer interdisziplinär relevanten Änderung

Trotz eines gemeinsamen Systemverständnisses kann es während der Verfeinerung vorkommen, dass in einer Entwicklungsdisziplin Änderungen mit interdisziplinären Auswirkungen vorgenommen werden müssen. Aus HMI-Sicht können Änderungen anderer Entwicklungsdisziplinen bspw. zu einer Adaption bestehender oder einer Erweiterung um neue Benutzeraufgaben führen.¹⁷ Andererseits können sich aus der HMI-Entwicklung selbst interdisziplinär relevante Änderungen ergeben, wenn bspw. festgestellt wird, dass die bisher vorgesehene Schnittstelle zur Steuerung nicht genügt, um eine Benutzeraufgabe geeignet durchführen zu können.

Gemäß dem in Abbildung 5-20 skizzierten Vorgehen führen solche Änderungen einer Entwicklungsdisziplin zu einer Anpassung des Systemmodells (*Mechatronisches Konzept 2.0*). Für das HMI ist dabei zunächst zu überprüfen, ob die Änderung einen Einfluss auf das bisherige Spektrum der Benutzeraufgaben besitzt. In diesem Fall sind die User Stories, die HMI-Ports sowie die Benutzerdialoge entsprechend anzupassen bzw. zu erweitern, um auf Ebene des Systemmodells über den gesamten Entwicklungsablauf eine vollständige Spezifikation der Benutzeraufgaben vorliegen zu haben. Auf dieser Basis kann weiterhin entschieden werden, ob neue HMI-Komponenten benötigt werden oder bestehende HMI-Komponenten anzupassen sind, um das Spektrum der Benutzeraufgaben umsetzen zu können. Im Fall einer neuen Komponente wird diese mit ihren Ports im Systemmodell abgebildet und über Kanäle eingebettet. Im Fall einer bestehenden Komponente wird deren Schnittstelle auf notwendige Änderungen analysiert und ggf. im Systemmodell angepasst. Das eigentliche Verhalten einer neuen oder bestehenden HMI-

¹⁷ Ein typisches Beispiel wäre das Einbringen eines zusätzlichen Ventils in der Mechanik, dessen Zustand an der Mensch-Maschine-Schnittstelle angezeigt werden soll und das über das HMI manuell zu bedienen ist.

Komponente wird hingegen nicht im Systemmodell abgebildet bzw. angepasst, da die hierfür verwendeten Zustandsautomaten nur eine vereinfachte Logik abbilden, die bei der erneuten disziplinspezifischen Implementierung der HMI-Programme (Iteration von Schritt 2) ohnehin anzupassen wäre. Ähnlich verhält es sich mit den Interaktionselementen im Systemmodell. Nur wenn auf Basis der Benutzeraufgaben entschieden wird, ein neues Interaktionselement im Systemmodell anzulegen, wird dieses mit Objekt- bzw. Screen-Repräsentationen abgebildet, um eine initiale Transformation des zugehörigen HMI-Programms zu ermöglichen. Bestehende Interaktionselemente werden hingegen nicht mehr angepasst, da notwendige Änderungen oder Erweiterungen an den einzelnen Ansichten direkt in den jeweiligen Entwicklungsumgebungen umgesetzt werden können.

Erneute HMI-Modelltransformation bei Beachtung der bisherigen Weiterentwicklungen

Sobald eine systemrelevante Änderung im mechatronischen Konzept hinreichend abgebildet ist, kann die disziplinspezifische Entwicklung der Mensch-Maschine-Schnittstelle mit einer erneuten HMI-Modelltransformation fortgesetzt werden. Dabei sind allerdings die bisherigen disziplinspezifischen Erweiterungen (*HMI 1.1*) zu berücksichtigen und mit den Änderungen bzw. Erweiterungen am Systemmodell (*Mechatronisches Konzept 2.0*) zusammenzuführen (*HMI 2.0*). Daher werden bei der Modelltransformation nur diejenigen Teile der HMI-Programme überschrieben, die im Systemmodell in jedem Fall den aktuellen Stand der Entwicklung repräsentieren. Dies betrifft zunächst die Schnittstellenvariablen innerhalb des Bausteins zur Kommunikation mit der Steuerung, die bei einer Änderung stets im Systemmodell aktualisiert werden müssen und daher im entsprechenden Baustein eines HMI-Programms überschrieben werden können. Ähnlich kann auch bei der erneuten Generierung der HMI-Komponenten vorgegangen werden. Für neue HMI-Komponenten werden entsprechende Teilbausteine mit Ein- und Ausgangsvariablen innerhalb des Logik-Bausteins generiert, für bestehende HMI-Komponenten werden nur die Ein- und Ausgangsvariablen überschrieben. Da das Verhalten im Systemmodell hingegen nicht mehr aktualisiert wird, wird die eigentliche Logik in den generierten Teilbausteinen nicht mehr überschrieben, wodurch bisherige disziplinspezifische Weiterentwicklungen erhalten bleiben. Der Visualisierungs-Baustein ist von der erneuten Generierung nicht betroffen, da die Screen-Repräsentationen von Interaktionselementen im Systemmodell nicht angepasst werden. Eine Ausnahme bilden hierbei neue Interaktionselemente, für die bei der Modelltransformation initiale HMI-Programme mit ihren einzelnen Bausteinen generiert werden.

Weiterführende HMI-Implementierung (Iterativer Ablauf der Schritte 2-4)

Der in den letzten Absätzen skizzierte Ablauf, bestehend aus disziplinspezifischer Umsetzung, Änderung des Systemmodells und erneuter Modelltransformation, kann sich während der Verfeinerung beliebig oft wiederholen, bis aus übergeordneter Sicht ein vollständiger virtueller Prototyp eines mechatronischen Systems vorliegt. Aus Sicht der Mensch-Maschine-Schnittstelle ist dies der Fall, wenn die einzelnen HMI-Programme eine Durchführung aller spezifizierten Benutzeraufgaben mit einer entsprechenden Gebrauchstauglichkeit ermöglichen und auf den ausgewählten Interaktionstechnologien ohne weiterführende Anpassungen ablauffähig sind.

Während des Verfeinerungsprozesses entfernen sich die einzelnen HMI-Programme dabei typischerweise von der ursprünglichen Spezifikation der Mensch-Maschine-Schnittstelle im mechatronischen Konzept. Durch den entwickelten Mechanismus der Änderung des Systemmodells und der erneuten Modelltransformation wird aber sichergestellt, dass das mechatronische Konzept und die einzelnen HMI-Programme kontinuierlich konsistent sind. Dies wird durch die einheitliche Trennung der auf Modellebene (Aufgaben, Schnittstellen und Komponenten) und Implementierungsebene (Logik und Visualisierung) durchzuführenden Änderungen ermöglicht. Dadurch wird auch gewährleistet, dass das Systemmodell bis zum Abschluss der Entwicklung gültig ist, wodurch einzelne Bestandteile des HMIs auch bei der Realisierung weiterer Systeme wiederverwendet werden können. Dabei ist allerdings zu beachten, dass die ursprünglich auf Systemebene modellierten Verhalten und 2D-Ansichten in der Regel disziplinspezifisch verfeinert werden, weswegen im Falle einer Wiederverwendung direkt auf die diesbezüglichen Umsetzungen in den jeweiligen HMI-Programmen zurückgegriffen werden sollte.

5.5 Aktivität IV: HMI-Qualitätssicherung

Die bislang vorgestellten Aktivitäten ermöglichen bereits eine interdisziplinäre Entwicklung von Mensch-Maschine-Schnittstellen im Kontext mechatronischer Systeme. Um allerdings die Funktionalität und Gebrauchstauglichkeit eines HMIs im Entwicklungsverlauf bewerten und anhand softwareergonomischer Kriterien (vgl. Abschnitt 2.3) optimieren zu können, ist eine zusätzliche Aktivität notwendig, die sich der HMI-Qualitätssicherung widmet. Wie in den Abschnitten 2.4 und 3.4 bereits dargestellt wurde, existieren hierfür vielfältige Möglichkeiten, die von Experten-Reviews über Papierprototypen bis zu Nutzerevaluationen an der realen Maschine reichen. Im Rahmen dieser Arbeit wird vor dem Hintergrund der

eingeschränkten Entwicklungskapazitäten im Maschinen- und Anlagenbau der Einsatz der Simulation für die HMI-Qualitätssicherung fokussiert. Hierfür bietet der ursprüngliche IMoMeSA-Ansatz mit seinem formalen Aufbau und den eingesetzten Simulationstechniken eine geeignete Basis (vgl. HACKENBERG ET AL. 2015, S. 33 ff.). In Abschnitt 5.5.1 wird deshalb dargelegt, inwiefern diese Techniken zu erweitern sind, um eine Qualitätssicherung von Mensch-Maschine-Schnittstellen zu ermöglichen. Neben dieser technischen Betrachtung wird in Abschnitt 5.5.2 ein Prozessablauf vorgestellt, der den zielführenden Einsatz der Qualitätssicherung während des gesamten Entwicklungsverlaufs ermöglicht. Dabei wird insbesondere auf die Unterschiede zwischen den Phasen der Konzeption und der Verfeinerung eingegangen, da die in Abschnitt 5.5.1 beschriebenen Techniken primär die Qualitätssicherung in der Konzeption adressieren.

5.5.1 Eingesetzte Simulationstechniken

Wie im letzten Abschnitt erwähnt, adressiert die Aktivität der Qualitätssicherung einerseits die Funktionalität einer Mensch-Maschine-Schnittstelle, um die generelle Durchführbarkeit von spezifizierten Benutzeraufgaben nachzuweisen. Andererseits soll explizit die Gebrauchstauglichkeit eines HMIs bewertet werden, um Benutzern eine intuitive Durchführung ihrer jeweiligen Aufgaben zu ermöglichen. Diese grundsätzliche Zweiteilung findet sich auch bei den eingesetzten Simulationstechniken wieder. Für die Absicherung der Funktionalität können sog. *Dialogtests* eingesetzt werden, während für die Absicherung der Gebrauchstauglichkeit sog. *Nutzerevaluationen* vorgesehen sind. Beide Simulationstechniken werden im Folgenden im Detail vorgestellt.

Dialogtests

Für einen simulationsbasierten Test der generellen Durchführbarkeit von Benutzeraufgaben bietet sich die Nutzung der über Nachrichten-Reihenfolge-Diagramme modellierten Benutzerdialoge an. Durch ihre syntaktische und semantische Ähnlichkeit zu den im ursprünglichen IMoMeSA-Ansatz vorhandenen Szenarien kann der dort adressierte Simulationsansatz als Basis verwendet und hinsichtlich der Spezifika von Benutzerdialogen erweitert werden (vgl. HACKENBERG ET AL. 2015, S. 33 ff.). Dieser Ansatz sieht es vor, ein Szenario in einer Simulationsumgebung aufzurufen, dort auszuführen und zu überprüfen, ob sich das System gemäß dem spezifizierten Szenario verhält. Dieses Grundprinzip lässt sich auch auf die hier adressierten Dialogtests übertragen, welche in Abbildung 5-21 schematisch dargestellt sind.

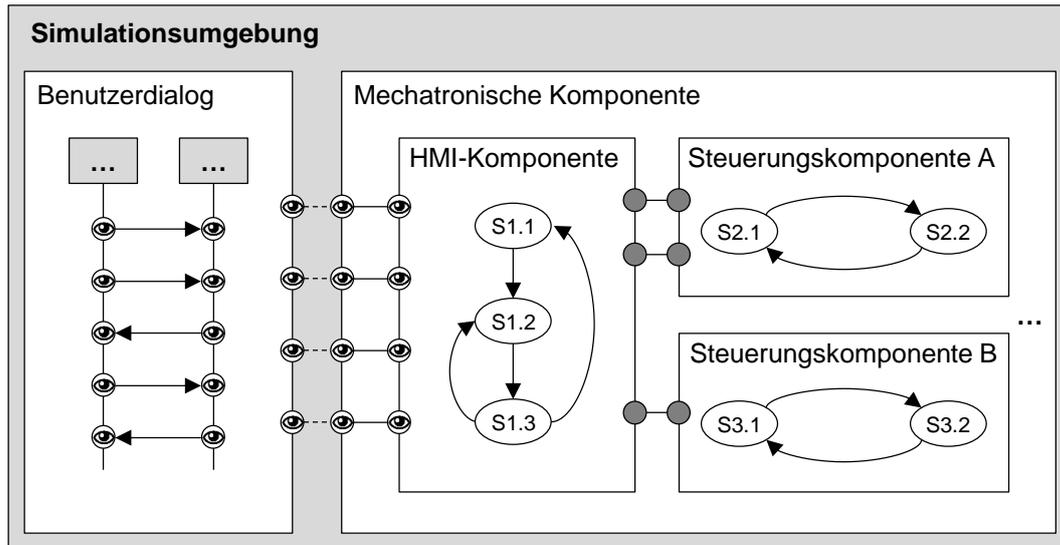


Abbildung 5-21: Schematische Übersicht über Dialogtests

Nach dem Aufruf eines Benutzerdialogs in der Simulationsumgebung werden die ersten modellierten Aktionen gemäß der Abfolge des Nachrichten-Reihenfolge-Diagramms ausgeführt, wodurch die entsprechenden HMI-Ports der mechatronischen Komponente beschrieben werden. Diese Eingaben werden innerhalb der mechatronischen Komponente gemäß dem Verhalten der Teilkomponenten, wie u. a. den HMI-Komponenten, verarbeitet und führen zu entsprechenden Zustandsänderungen und Aktionen. Sofern dieses Verhalten richtig abgebildet ist, werden durch die Ausführung der Zustandsautomaten schließlich die notwendigen Ausgaben an den HMI-Ports gesetzt, um den Benutzerdialog mit der Ausführung der nächsten Nutzeraktionen fortsetzen zu können. Dieser Vorgang wiederholt sich, bis ein spezifizierter Dialog vollständig simuliert oder durch einen Time-Out abgebrochen wurde. Das Ergebnis wird in der Simulationsumgebung dargestellt und weist darauf hin, ob die Modellierung der mechatronischen Komponente eine Durchführung der Benutzeraufgabe aus einer funktionalen Sicht ermöglicht. Wie bereits in der schematischen Übersicht in Abbildung 5-21 deutlich wird, beschränkt sich die Erprobung der mechatronischen Komponente nicht alleine auf die in den HMI-Komponenten modellierte Logik der Mensch-Maschine-Schnittstelle. Vielmehr wird geprüft, ob alle Bestandteile einer mechatronischen Komponente, wie auch Steuerungs- oder Elektronikkomponenten, bereits soweit umgesetzt sind, dass ein Benutzer seine konkrete Aufgabe im Umgang mit der Maschine bzw. Anlage grundsätzlich durchführen kann. Dabei fokussieren Dialogtests eine funktionale Absicherung eines mechatronischen Systems, während für Aspekte der Gebrauchstauglichkeit die im nächsten Abschnitt betrachteten Nutzerevaluationen zum Einsatz kommen können.

Zur einfachen Verständlichkeit beschränken sich die Ausführungen im letzten Absatz auf den Fall, dass mit einem Benutzerdialog genau diejenige mechatronische Komponente getestet wird, innerhalb derer der Dialog modelliert wurde. Wie allerdings in Abschnitt 5.2.1 deutlich wurde, können Benutzerdialoge auch auf die HMI-Ports von abgeleiteten mechatronischen Komponenten zugreifen, wodurch auch deren Verhalten in der Simulation abgesichert werden kann. Eine weitere Vereinfachung im letzten Absatz betrifft die Fokussierung auf einen einzelnen Nutzer. Benutzerdialoge berücksichtigen jedoch auch den Fall, dass mehrere Benutzer zu der Umsetzung einer Benutzeraufgabe beitragen können und somit als eigenständige Entitäten im Dialog auftreten. Für die Simulation ist dieser Gesichtspunkt jedoch unproblematisch, da die Aktionen im Nachrichten-Reihenfolge-Diagramm nur anhand ihrer spezifizierten Reihenfolge bearbeitet werden, ohne dass die auslösende Entität bzw. die durchführende Benutzerrolle darauf einen Einfluss hat.

Weiterhin wird durch die Verteilung von Benutzerdialogen über die mechatronische Komponentenstruktur (vgl. Abschnitt 5.2.2) die Möglichkeit zur Erprobung einzelner Systembestandteile losgelöst vom Gesamtsystem geschaffen. Benutzerdialoge haben zwar einen Durchgriff auf die HMI-Ports abgeleiteter mechatronischer Komponenten, jedoch sind sie von übergeordneten Komponenten oder weiteren Komponenten der gleichen Hierarchieebene unabhängig. Dadurch können einzelne mechatronische Komponenten und somit spezifische Bestandteile der Mensch-Maschine-Schnittstelle bereits zu einem Zeitpunkt erprobt werden, zu dem nur die betrachtete Komponente sowie ihre etwaigen Teilkomponenten modelliert sind. Dies ist insbesondere für die Anwendung der Dialogtests in der Konzeptionsphase relevant, auf die in Abschnitt 5.5.2 noch im Detail eingegangen wird.

Wie eingangs erwähnt, fokussieren Dialogtests ausschließlich die Erprobung der Funktionalität einer Mensch-Maschine-Schnittstelle, wofür in der Simulation die HMI-Ports, die HMI-Komponenten sowie deren integriertes Verhalten benötigt werden. Die Interaktionselemente kommen bewusst nicht zum Einsatz, um zunächst die Qualität der Logik einer Mensch-Maschine-Schnittstelle losgelöst von der Darstellung sicherzustellen. Um allerdings auf Modellebene auch die Darstellung und somit die HMI-Gebrauchstauglichkeit bewerten zu können, wurde im Rahmen dieser Arbeit eine zweite Simulationstechnik entwickelt, die im nachfolgenden Abschnitt beschrieben wird. Durch die Einbindung realer Nutzer bei dieser Technik kann neben der Gebrauchstauglichkeit auch die Funktionalität des HMIs weiterführend abgesichert werden. Dies ist damit zu begründen, dass in einem Benutzerdialog die Funktionalität nur anhand eines Ausschnitts des Nutzerverhaltens abgesichert wird, das nicht zwingend mit dem tatsächlichen Verhalten übereinstimmt.

Nutzerevaluationen

Im Unterschied zur Funktionalität hängt die Gebrauchstauglichkeit einer Mensch-Maschine-Schnittstelle im Wesentlichen davon ab, ob die jeweilige Darstellung Benutzern eine einfache und intuitive Durchführung ihrer Arbeitsaufgaben ermöglicht. Vor diesem Hintergrund ist es zielführend, die Gestaltung eines HMIs an softwareergonomischen Kriterien (vgl. Abschnitt 2.3) auszurichten. Die Einhaltung dieser Kriterien lässt sich dabei durch eine frühzeitige und kontinuierliche Einbeziehung von späteren Benutzern bewerten, die in der Regel ein integraler Bestandteil von Ansätzen zur HMI-Entwicklung ist (BENGLER 2015, S. 618). Daher wurde in dieser Arbeit die Simulationstechnik der Nutzerevaluationen entwickelt. An die Stelle eines modellierten Nutzerverhaltens in den Benutzerdialogen treten dabei reale Benutzer, die über die Screen-Repräsentationen eines Interaktionselements mit dem mechatronischen System bzw. einer mechatronischen Komponente interagieren. Wie in Abbildung 5-22 deutlich wird, ist die schematische Übersicht dieser Simulationstechnik ansonsten weitestgehend analog zu den Benutzerdialogen.

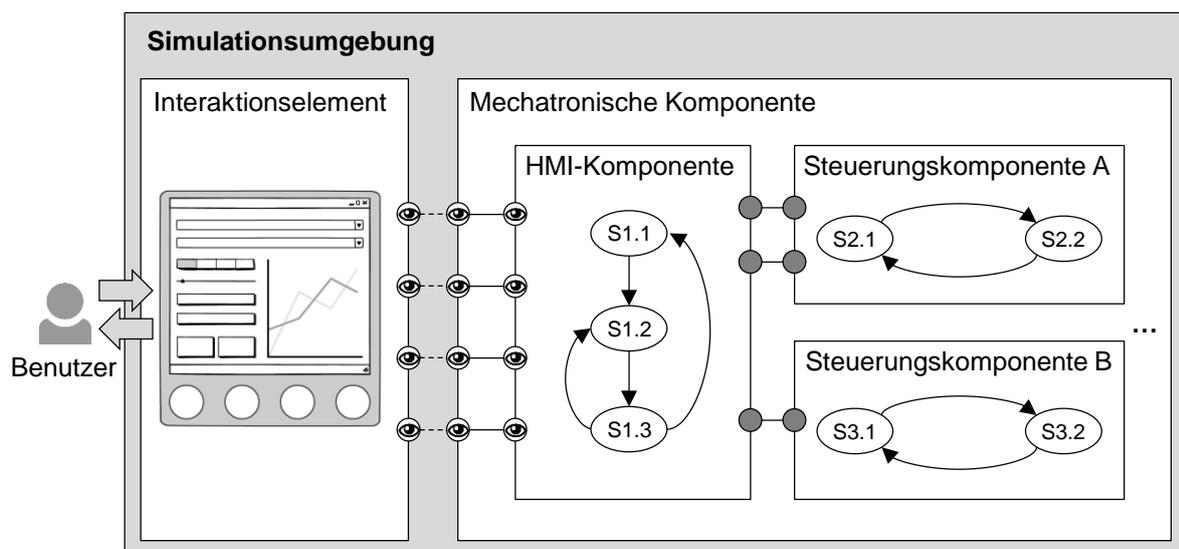


Abbildung 5-22: Schematische Übersicht über Nutzerevaluationen

Zunächst wird ein Interaktionselement mit der mechatronischen Komponente, in der es modelliert wurde, in die Simulationsumgebung geladen. Dort wird dem Benutzer die erste Screen-Repräsentation des Interaktionselements gemäß dem spezifizierten Navigationskonzept dargestellt. Der Benutzer kann nun mit dem modellierten HMI über dessen Bedien- und Anzeigeelemente (Buttons, etc.) interagieren und dadurch Eingaben an den HMI-Ports der betrachteten mechatronischen Komponente oder deren Teilkomponenten betätigen. Die simulierte mechatronische Komponente reagiert wie bei Dialogtests auf diese Eingaben durch entsprechende Zustandswechsel

und Aktionen, wodurch Ausgaben an den HMI-Ports der mechatronischen Komponente erfolgen, die auf den Screen-Repräsentationen des Interaktionselements dargestellt werden. Grundsätzlich ist es dabei dem Benutzer überlassen, welche Eingaben er in welcher Reihenfolge durchführt und wie er auf entsprechende Rückmeldungen reagiert. Allerdings bietet es sich für eine zielgerichtete Erprobung der Gebrauchstauglichkeit an, den Benutzer mit konkreten Aufgaben zu konfrontieren, die ihm anhand der User Stories veranschaulicht werden.¹⁸ Während der Durchführung dieser Aufgaben ist zu evaluieren, inwiefern das modellierte HMI den Benutzer unterstützt oder ihn evtl. sogar behindert. Für die Beantwortung dieser Fragestellung können etablierte Methoden des Usability Engineerings wie z. B. eine Nutzerbeobachtung oder ein strukturiertes Experten-Interview genutzt werden. Die Ergebnisse dieser Auswertung ermöglichen bspw. Rückschlüsse hinsichtlich der Anordnung und Benennung von Bedien- und Anzeigeelementen oder des eingesetzten Navigationskonzepts und können daher für Optimierungen der Logik und Darstellung der Mensch-Maschine-Schnittstelle verwendet werden.

Um bei der Anwendung der Nutzerevaluationen möglichst realitätsnahe Ergebnisse zu erzielen, ist die mit dem HMI-Modell interagierende Person ein wichtiger Faktor. Ideal ist dabei die Einbeziehung der tatsächlichen Benutzer, was allerdings aus Aufwandsgründen im Maschinen- und Anlagenbau in der Regel nicht möglich ist. Dies macht es notwendig, zumindest die relevanten Eigenschaften der einzelnen Benutzerrollen, die mit der Maschine interagieren, zu kennen (z. B. Bildungsstand) und bei der Durchführung der Simulation zu berücksichtigen. Insbesondere sollte beachtet werden, dass auch zwischen einzelnen Benutzerrollen, wie einem Werker oder Administrator, unterschiedliche Ausprägungen relevanter Eigenschaften vorliegen können, was auf das Empfinden der Gebrauchstauglichkeit einen wichtigen Einfluss hat. Dieser Aspekt ist auch bei der Absicherung von Aufgaben zu berücksichtigen, an deren Durchführung mehrere Personen beteiligt sind. Bei der Nutzerevaluation sollte in diesem Fall je eine Person pro Nutzerrolle entsprechend ihrer jeweiligen Teilaufgabe mit dem Simulationsmodell interagieren.

An dieser Stelle ist nochmals darauf hinzuweisen, dass mit Nutzerevaluationen nicht nur die Gebrauchstauglichkeit, sondern auch die Funktionalität einer Mensch-Maschine-Schnittstelle weiterführend abgesichert wird. Das beliebige Verhalten der mit dem HMI-Modell interagierenden Personen, das in Benutzerdialogen nur ausschnittsweise abgebildet wird, ermöglicht die Identifikation von Fehlern in der

¹⁸ Benutzerdialoge sollten dem Benutzer demgegenüber nicht zur Verfügung gestellt werden, um ihm nicht vorab die Reihenfolge der einzelnen Tätigkeiten vorzuschreiben.

HMI-Logik, wie bspw. Deadlocks, welche durch die Benutzerdialoge zuvor nicht erkannt wurden. Dies gilt allerdings nicht nur für die Logik der Mensch-Maschine-Schnittstelle, sondern lässt sich wie bei Dialogtests auf die Funktionalität des gesamten mechatronischen Systems übertragen, welche für die Durchführung der Benutzeraufgaben benötigt wird. Dabei wird in Analogie zu den Dialogtests immer diejenige mechatronische Komponente mit ihren jeweiligen Teilkomponenten getestet, innerhalb derer das Interaktionselement modelliert wurde. Dadurch kann auch bei Nutzerevaluationen der Gedanke des modularen Testens beibehalten werden, sofern einzelne Interaktionselemente nur für einen Ausschnitt eines mechatronischen Systems relevant sind.

5.5.2 Prozessablauf für den Einsatz der Simulationstechniken

Wie in den vorangegangenen Ausführungen in Abschnitt 5.5.1 bereits deutlich wurde, adressieren die entwickelten Simulationstechniken vor allem eine Absicherung von Mensch-Maschine-Schnittstellen während der *Konzeption*. Durch die Umsetzung der HMI-Programme in spezifischen Zielumgebungen können diese Simulationstechniken in der Phase der *Verfeinerung* zwar nicht mehr genutzt werden, allerdings kann dafür auf etablierte disziplinspezifische Methoden zurückgegriffen werden.

Vor diesem Hintergrund sind die nachfolgenden Ausführungen zur Anwendung der Qualitätssicherung im Entwicklungsverlauf in zwei Abschnitte gegliedert. Zunächst wird der Einsatz der im Rahmen dieser Arbeit entwickelten Simulationstechniken in der Phase der Konzeption erläutert, bevor die Möglichkeiten zur Qualitätssicherung mit etablierten disziplinspezifischen Methoden sowie deren Anwendung während der Verfeinerung beschrieben werden. Zuvor ist in Abbildung 5-23 noch ein allgemeiner Überblick über den Prozessablauf für die Qualitätssicherung dargestellt.

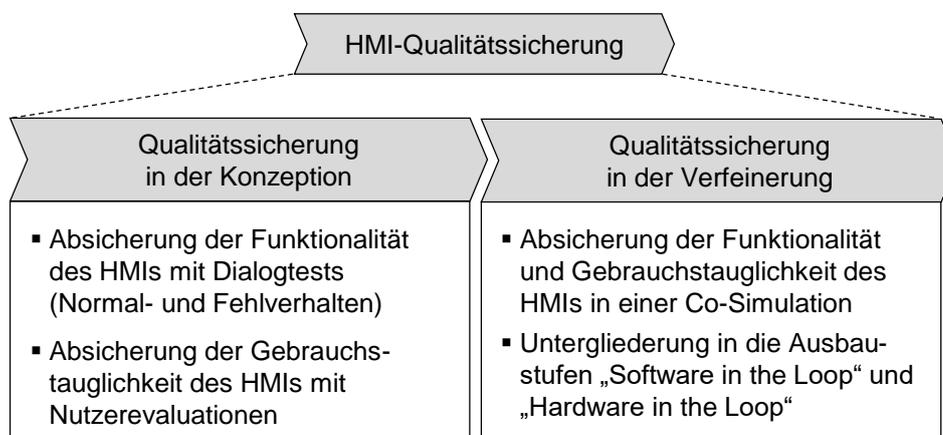


Abbildung 5-23: Prozessablauf zur HMI-Qualitätssicherung

Qualitätssicherung in der Konzeption

Da beim Aufbau der mechatronischen Komponentenstruktur noch die Verhaltensmodelle für die Erprobung der mechatronischen Komponenten fehlen, beginnt die Qualitätssicherung in der Konzeption mit der Ausgestaltung der modellierten Komponenten (vgl. Abbildung 5-7). Dabei kommen zunächst die *Dialogtests* zum Einsatz, wobei gemäß dem Bottom-up-Vorgehen aus der HMI-Modellierung zunächst die mechatronischen Komponenten der untersten Hierarchieebene im Sinne des Unit Testings (vgl. DESIKAN & RAMESH 2008, S. 56 ff.) erprobt werden. Erst wenn die Funktionalität dieser mechatronischen Komponenten mit ihren Bestandteilen (Mechanik, Elektrik, Steuerungstechnik und HMI) abgesichert wurde, wird mit deren Integration auf der nächsthöheren Ebene begonnen. Im Sinne des Integration Testings (vgl. DESIKAN & RAMESH 2008, S. 107 ff.) werden die dort modellierten Benutzerdialoge verwendet, um das Zusammenspiel der zuvor einzeln getesteten Komponenten sowie die Funktionalität der mechatronischen Komponenten dieser Hierarchieebene abzusichern. In Anlehnung an das übergeordnete Vorgehen in der Konzeption wiederholt sich diese Herangehensweise zur Qualitätssicherung mit Dialogtests, bis die Systemebene erreicht ist und die hier modellierten Benutzerdialoge erfolgreich abgesichert werden können.

Hierdurch ist eine korrekte Funktionsweise des mechatronischen Konzepts und seines HMIs hinsichtlich des Normalverhaltens sichergestellt und es kann mit der Erweiterung des Konzepts um mögliches Fehlverhalten fortgefahren werden. Auch hier werden zunächst die Dialogtests eingesetzt, um zu ermitteln, ob Benutzer ihre modellierten Aufgaben trotz auftretender Fehler des Systems, wie einem Sensorausfall, oder eigener Fehler, wie einer falschen Werteingabe, durchführen können oder das System zumindest geeignet auf einen Fehler reagiert. Die Einbindung der Dialogtests folgt dabei einem festen Schema: Nach dem Einbringen eines konkreten Fehlverhaltens in eine Komponente des mechatronischen Konzepts wird anhand der Dialogtests analysiert, ob durch den Fehler einzelne Benutzerdialoge nicht mehr korrekt durchgeführt werden können. Ist dies der Fall, wird eine entsprechende Fehlerbehandlung in das System eingebracht und ggf. der Benutzerdialog angepasst (z. B. durch eine Anzeige des aufgetretenen Fehlers). Daraufhin können die Dialogtests erneut durchgeführt und hinsichtlich eines korrekten Ablaufs evaluiert werden.

Wenn auf diese Weise alle relevanten Fehler eines Anwendungsfalls behandelt wurden, können die Interaktionselemente gestaltet und mit den *Nutzerevaluationen* abgesichert werden. Zur strukturierten Anwendung dieser Simulationstechnik bietet es sich an, die einzelnen Interaktionselemente isoliert voneinander zu betrachten

und für die jeweilige Evaluation die modellierten User Stories einzusetzen. Diese werden von den Benutzern während der Simulation ausgeführt, wobei mit den User Stories begonnen werden sollte, welche für die mechatronischen Komponenten der untersten Hierarchieebene spezifiziert wurden und die eine geringere Komplexität als die umfassenden User Stories auf Systemebene besitzen. Dies ermöglicht eine aufwandsarme Evaluation der Gebrauchstauglichkeit eines Interaktionselements hinsichtlich dieser Grundaufgaben und gestattet ferner die isolierte Erprobung der hierfür notwendigen Screen-Repräsentationen, ohne dass zu diesem Zeitpunkt ein gesamtes Interaktionselement modelliert sein muss. Sobald in einem alternierenden Vorgehen aus Nutzerevaluationen und der Anpassung der Screen-Repräsentationen die Gebrauchstauglichkeit auf dieser Ebene sichergestellt ist, kann analog zu den Dialogtests mit Benutzeraufgaben der nächsthöheren Hierarchieebene fortgefahren werden. Dieses Vorgehen wiederholt sich, bis die umfassenden Benutzeraufgaben auf Systemebene erreicht werden. Durch deren Evaluation und die gewonnenen Erkenntnisse zur Anpassung einzelner Screen-Repräsentationen und des eingesetzten Navigationskonzepts kann eine hohe Gebrauchstauglichkeit eines modellierten Interaktionselements sichergestellt werden.

Dialogtests und Nutzerevaluationen kommen während der Konzeption von mechatronischen Systemen zu unterschiedlichen Zeitpunkten und losgelöst voneinander zum Einsatz. Dies wird durch die verschiedenen Zielgrößen beider Simulationstechniken (Absicherung der Funktionalität bzw. der Gebrauchstauglichkeit) sowie die bewusste Trennung der Logik vom Design ermöglicht, wodurch sich der Aufwand zur Qualitätssicherung erheblich reduzieren lässt. So kann ein Großteil der HMI-Funktionalität bereits mit der Simulationstechnik der Dialogtests abgesichert werden, was in den Nutzerevaluationen die spezielle Fokussierung der Gebrauchstauglichkeit von Interaktionselementen erlaubt. Damit lassen sich die aufwändigen Schritte zur Einbindung von Nutzern, deren Beobachtung durch Usability-Experten sowie die Auswertung der Evaluationsergebnisse auf ein Minimum reduzieren, ohne dabei qualitative Einbußen im Hinblick auf das HMI hinnehmen zu müssen.

Qualitätssicherung in der Verfeinerung

Wie bereits zu Beginn dieses Abschnitts angesprochen, können die entwickelten Simulationstechniken in der Verfeinerung nicht mehr verwendet werden. Zwar wird das Systemmodell auch in dieser Phase noch zur Konsistenzsicherung zwischen den Disziplinen eingesetzt, allerdings wird aus HMI-Sicht sowohl die Funktionalität als auch die Darstellung innerhalb der jeweiligen Entwicklungsumgebungen adaptiert bzw. konkretisiert. Daher müssen sich die Maßnahmen zur Qualitätssicherung auf

die dort entwickelten Programme beziehen, wofür allerdings spezifische Ansätze in Abhängigkeit von der Interaktionstechnologie und Entwicklungsumgebung benötigt werden. Diese Konstellation führt dazu, dass im Rahmen dieser Arbeit kein eigenes Konzept zur Absicherung von HMIs während der Verfeinerung entwickelt wurde. Generell wird allerdings in Anlehnung an den ursprünglichen IMoMeSA-Ansatz (vgl. HACKENBERG ET AL. 2015, S. 51 f.) der Einsatz der Simulation als generelles Mittel zur Qualitätssicherung während der gesamten Entwicklung vorgeschlagen.

Um diese zu ermöglichen, kann eine Co-Simulation (vgl. FRIEDRICH 2011) genutzt werden, um zur Laufzeit Daten zwischen verschiedenen Simulationsumgebungen auszutauschen. Aus Sicht des HMIs bietet dieses Konzept die Möglichkeit, HMI-Programme mit einem Verhaltens- und Geometriemodell des mechatronischen Systems zu verknüpfen und dadurch eine vergleichbare Konstellation wie bei den Nutzerevaluationen zu schaffen. Sowohl die Mensch-Maschine-Schnittstelle als auch das Verhaltens- und Geometriemodell werden dabei in entsprechenden Laufzeitumgebungen ausgeführt und tauschen bei der Simulation die notwendigen Daten aus, welche für die Verarbeitung von Nutzereingaben oder die Anzeige von Informationen an der Mensch-Maschine-Schnittstelle benötigt werden. Als allgemeingültiger Standard für diesen Datenaustausch wurde im IMoMeSA-Ansatz das Functional Mock-up Interface (vgl. BLOCHWITZ ET AL. 2012) diskutiert, allerdings besteht in Abhängigkeit von den eingesetzten Laufzeitumgebungen auch die Möglichkeit, proprietäre Formate für den Datenaustausch zu verwenden.

Unabhängig von den eingesetzten Simulationstechniken und Laufzeitumgebungen lässt sich die Co-Simulation aus methodischer Sicht als Fortsetzung der Nutzerevaluationen aus der Phase der Konzeption betrachten. Auch hier können Benutzer mit dem HMI in der Laufzeitumgebung interagieren und ihre einzelnen Aufgaben durchführen. Dabei kann analog zu den Nutzerevaluationen anhand von Nutzerbeobachtungen oder Experteninterviews die Gebrauchstauglichkeit des verfeinerten HMIs ermittelt und verbessert werden. Darüber hinaus kann auch erkannt werden, ob durch die Verfeinerung des HMIs oder des Verhaltens- und Geometriemodells funktionale Fehler entstanden sind, die dann behoben werden können. Abschließend ist zu erwähnen, dass die Co-Simulation während der Verfeinerung in der Regel zwei Ausbaustufen umfasst. Dabei werden zunächst die HMI-Programme gemeinsam mit dem Verhaltens- und Geometriemodell auf einem Simulationsrechner ausgeführt und getestet (*Software in the Loop*). Im weiteren Verlauf der Verfeinerung bietet es sich allerdings an, auch die realen Endgeräte in die Erprobung einzubeziehen und diese mit dem Simulationsrechner zu koppeln, um dadurch den Gebrauch des HMIs bestmöglich nachbilden zu können (*Hardware in the Loop*).

5.6 Zusammenfassung

Im zurückliegenden Kapitel wurde die erarbeitete Methodik zur modellbasierten Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau im Detail vorgestellt. Ausgehend von einem Gesamtüberblick der Methodik sowie deren Einbettung in die gewählte Forschungsgrundlage, wurden die vier zentralen Aktivitäten zur HMI-Entwicklung anhand ihrer technischen Lösungsbausteine und der jeweiligen Prozessabläufe beschrieben.

Dabei widmet sich die erste Aktivität der Modellierung einer Mensch-Maschine-Schnittstelle im Kontext eines mechatronischen Systemmodells. Hierfür wurde die IMoMeSA-Modellierungstechnik als Basis verwendet und um neue Modellierungselemente zur Abbildung von HMIs, wie bspw. Interaktionselemente, erweitert. Des Weiteren wurde deren methodischer Einsatz während der übergeordneten Phase der Konzeption beschrieben. Da es während der Modellierung erforderlich ist, eine Auswahl hinsichtlich einzusetzender Interaktionstechnologien zu treffen, wird diese Aufgabe als zweite Aktivität der Entwicklungsmethodik adressiert. Den zentralen Lösungsbaustein bildet dabei eine Einflussmatrix zur Gegenüberstellung von Einflussfaktoren und Merkmalen einzelner Interaktionstechnologien. Diese ermöglicht die Identifikation wichtiger Merkmale, die ein HMI für einen konkreten Anwendungsfall aufweisen sollte. Ein zugehöriger Prozessablauf definiert, wie die identifizierten Merkmale bei der Technologieauswahl genutzt werden können.

Auf der Basis eines HMI-Modells und einer ausgewählten Interaktionstechnologie erfolgt die Implementierung einer Mensch-Maschine-Schnittstelle. Hierfür wurde ein Ansatz zur Modelltransformation erarbeitet, um bereits auf Modellebene abgebildete Inhalte auch bei der HMI-Implementierung verwenden zu können und damit Mehraufwände zu vermeiden. Weiterhin wurde ein Vorgehen definiert, um die Modelltransformation durchgängig während der Implementierung zu nutzen, wodurch eine kontinuierliche Konsistenz zwischen den Entwicklungsdisziplinen sichergestellt wird. Die letzte Aktivität fokussiert schließlich die Qualitätssicherung während des gesamten Entwicklungsverlaufs. Sowohl bei der Modellierung als auch bei der Implementierung wird hier der Einsatz von Simulation vorgeschlagen, für die einerseits eigene Techniken (Dialogtests und Nutzerevaluationen) entwickelt wurden und andererseits ein Prozessablauf erarbeitet wurde.

Die einzelnen Lösungsbausteine sowie das jeweils vorgeschlagene Vorgehen für deren Einsatz bilden die Basis für die Anwendung und Bewertung der modellbasierten Entwicklungsmethodik, die im nächsten Kapitel betrachtet werden.

6 Anwendung und Bewertung der modellbasierten Entwicklungsmethodik

Die im letzten Kapitel vorgestellte Methodik zur modellbasierten Entwicklung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau wird in diesem Kapitel angewendet und anschließend bewertet. Dabei wird in Abschnitt 6.1 zunächst die Vorgehensweise dargelegt, die für die Auswahl eines industriell relevanten Fallbeispiels und für eine möglichst ganzheitliche Bewertung der Methodik gewählt wurde. Daraufhin wird in Abschnitt 6.2 die von einem Anwendungsfall unabhängige Umsetzung der technischen Lösungsbausteine in entsprechenden Softwarewerkzeugen beschrieben. Mit diesen Werkzeugen erfolgt in Abschnitt 6.3 die Anwendung der Entwicklungsmethodik an einem industrienahen Fallbeispiel. Die nachfolgenden Abschnitte 6.4 und 6.5 widmen sich schließlich der technischen sowie wirtschaftlichen Bewertung des erarbeiteten Ansatzes, bevor das Kapitel in Abschnitt 6.6 kurz zusammengefasst wird.

6.1 Vorgehensweise bei der Anwendung und Bewertung

Bei der Anwendung und Bewertung der erarbeiteten Entwicklungsmethodik konnte auf ein interdisziplinäres Team mit Experten aus den Bereichen des Maschinen- und Anlagenbaus, des Software Engineerings sowie des Usability Engineerings zurückgegriffen werden. Dieses Team begleitete im Rahmen des Forschungsprojekts „Phasenübergreifende Entwicklung von Benutzerschnittstellen im Maschinen- und Anlagenbau – PEBEMA“ (RICHTER 2016) kontinuierlich die Konzeption, Anwendung und Bewertung des Ansatzes in regelmäßigen Expertenworkshops.

Neben kontinuierlichen Diskussionen des Konzepts, deren Erkenntnisse direkt in die Ausarbeitung der Entwicklungsmethodik eingeflossen sind, stand insbesondere die Auswahl eines geeigneten Fallbeispiels im Fokus der Workshops. Dazu wurde das Maschinenportfolio der Anwendungsunternehmen aus dem Werkzeug- und Verpackungsmaschinenbau hinsichtlich typischer Benutzeraufgaben analysiert, die in einem Fallbeispiel abzubilden sind. Neben diesem Aufgabenspektrum wurde auch die konträre Anforderung im Hinblick auf die Komplexität des Fallbeispiels berücksichtigt, die so gewählt werden sollte, dass ein schneller Erkenntnisgewinn und eine leicht verständliche Demonstration der Entwicklungsmethodik ermöglicht wird. Das auf Basis der einzelnen Anforderungen ausgewählte Fallbeispiel wird in Abschnitt 6.3.1 eingeführt.

Basierend auf der Auswahl des Fallbeispiels wurden die einzelnen Aktivitäten der Entwicklungsmethodik angewendet, um eine Mensch-Maschine-Schnittstelle für das Beispiel zu konzipieren und umzusetzen. Die dabei generierten Ergebnisse, im Einzelnen das HMI-Modell, die ausgewählte Interaktionstechnologie und das HMI-Programm, wurden erneut in Expertenworkshops evaluiert. Dabei stand speziell die Gebrauchstauglichkeit des resultierenden HMIs im Vordergrund, die von den beteiligten Usability Experten analysiert wurde. Die Ergebnisse dieser Evaluation flossen in die technische Bewertung der Entwicklungsmethodik (vgl. Abschnitt 6.4) ein, die abschließend mit dem Expertenteam diskutiert und abgestimmt wurde.

6.2 Softwareumsetzung der technischen Lösungsbausteine

Um die Entwicklungsmethodik an industriellen Fallbeispielen anwenden zu können, war eine softwareseitige Umsetzung der technischen Lösungsbausteine der vier Entwicklungsaktivitäten notwendig. Analog zur Ausarbeitung des wissenschaftlichen Konzepts konnte auch bei der Umsetzung auf den IMoMeSA-Ansatz zurückgegriffen werden, der in dem Softwarewerkzeug MACON prototypisch implementiert wurde (HACKENBERG ET AL. 2016). Dieses Werkzeug wurde im Rahmen dieser Arbeit entsprechend adaptiert, um die erweiterte Modellierungstechnik, die Modelltransformation zur HMI-Generierung sowie die konzipierten Simulationstechniken zu implementieren (vgl. Abschnitt 6.2.1). Die erarbeitete Einflussmatrix zur Unterstützung der Technologieauswahl wurde dagegen in einem Excel-basierten Prototyp umgesetzt, der in Abschnitt 6.2.2 beschrieben wird.

6.2.1 Umsetzung der Modellierungstechnik, der Modelltransformation und der Simulationstechniken

Das Softwarewerkzeug MACON (MANufacturing CONception) wurde von HACKENBERG ET AL. (2016) entwickelt, um den IMoMeSA-Ansatz an unterschiedlichen Fallbeispielen erproben zu können. Technologisch basiert MACON auf einer Java-Applikation, die in der integrierten Entwicklungsumgebung Eclipse unter Verwendung des JavaFX-Frameworks implementiert wurde. Intern ist das Werkzeug in einer „Relaxed Layered System“-Architektur (BUSCHMANN ET AL. 1998, S. 47-48) mit verschiedenen Modulen aufgebaut. Die unterste Schicht bildet dabei das Modul *Core*, welches Basisfunktionalitäten wie bspw. ein Event Handling zur Verfügung stellt. Das Modul *Model* repräsentiert demgegenüber die Modellierungstechnik und ist zudem für die Verwaltung von Modellen als JSON-Dokumente

verantwortlich. Auf diese JSON-Dokumente greift auch das Modul *Transformation* zu, welches für die initiale Generierung der Entwicklungsartefakte der einzelnen Disziplinen (CAD-Modell, Steuerungscode, etc.) verantwortlich ist. Des Weiteren ermöglichen die Module *Checker* und *Simulator* einen Test von Modellen hinsichtlich ihrer syntaktischen und semantischen Korrektheit. Auf der obersten Schicht ist das Modul *Workbench* angeordnet, welches das HMI des Werkzeugs zur Modellierung von mechatronischen Systemen beinhaltet (vgl. Abbildung 6-1).

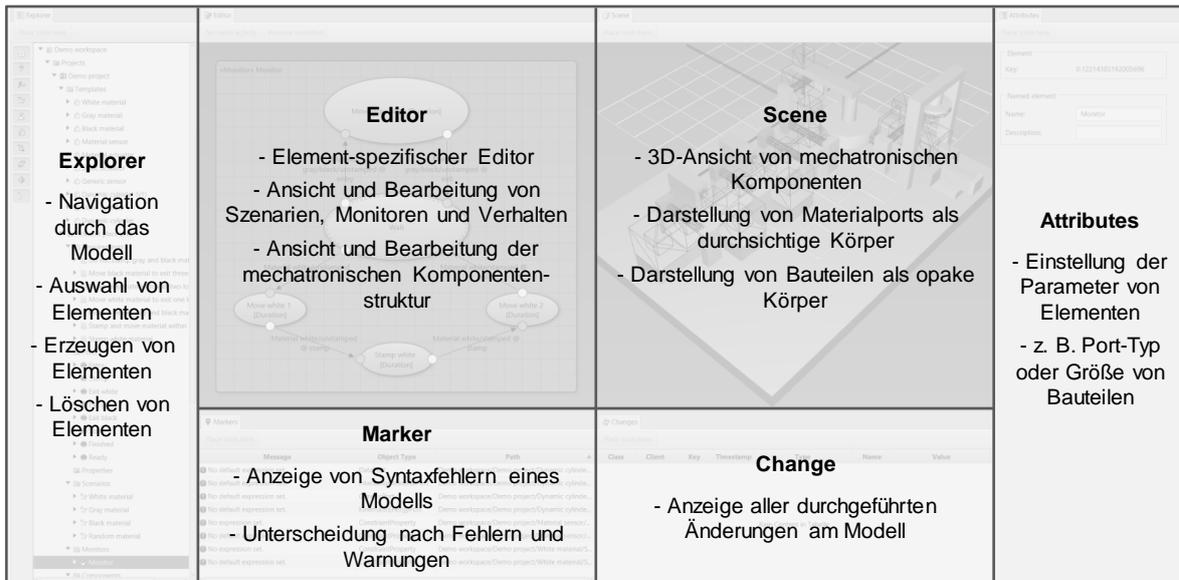


Abbildung 6-1: Mensch-Maschine-Schnittstelle von MACON
(in Anlehnung an HACKENBERG ET. AL. 2016, S. 1177)

Wie in der Abbildung deutlich wird, ist das HMI von MACON in voneinander gekapselte Bereiche unterteilt. Das zentrale Element bildet zunächst der am linken Bildrand angeordnete *Explorer*, in dem beliebige Elemente, wie Komponenten oder Szenarien, erzeugt, ausgewählt oder gelöscht werden können. Die Inhalte eines ausgewählten Elements, wie bspw. ein Zustandsautomat, werden im *Editor*-Bereich angezeigt und können dort bearbeitet werden, wie bspw. beim Modellieren von Zuständen und Transitionen. Für die geometrischen Elemente einer mechatronischen Komponentenstruktur (Bauteile und Materialports) ist der Bereich *Scene* vorgesehen. Dieser Bereich dient allerdings nur zur Visualisierung der geometrischen Elemente, während die Eigenschaften eines Bauteils oder Materialports, wie die Farbe, Größe oder Orientierung, im *Attribut*-Bereich festgelegt werden. Generell können dort in Abhängigkeit des im *Explorer* ausgewählten Elements dessen spezifische Eigenschaften definiert werden. Die beiden weiteren Bereiche *Marker* und *Change* dienen ferner zur Anzeige von Syntaxfehlern eines Modells, wie bspw. einer Verbindung inkompatibler Ports, sowie zur Darstellung aller durchgeführten

Änderungen am Modell. Für die Simulation von modellierten Systemen ist weiterhin eine separate Ansicht vorgesehen, die über einen Button „Start Simulation“ erreicht werden kann. Dort werden nach Ablauf der Simulation für ein ausgewähltes Szenario die Ergebnisse, wie z. B. der Werteverlauf von einzelnen Ports, dargestellt und es besteht die Möglichkeit, den Materialfluss von Werkstücken sowie die Bewegungen der Bauteile in einer Animation zu visualisieren.

Der grundsätzliche Programmaufbau sowie die Struktur des HMIs von MACON konnten auch bei der im Rahmen dieser Arbeit durchgeführten Erweiterung des Werkzeugs beibehalten werden. Für die Umsetzung der erweiterten Modellierungstechnik wurde zunächst das Modul *Model* gemäß dem spezifizierten Meta-Modell (vgl. Anhang A2) angepasst. Durch den modularen Aufbau des Programms konnten die weiteren Module dabei weitestgehend unverändert bleiben, einzig das HMI als Teil der *Workbench* musste im Bereich des Editors zur Darstellung und Bearbeitung der Screen-Repräsentationen von Interaktionselementen erweitert werden (vgl. Abbildung 6-2). Für die Darstellung der weiteren Modellierungselemente, wie HMI-Komponenten oder Benutzerdialoge, konnte auf bereits vorhandene Mechanismen zur Darstellung und Bearbeitung im Editor zurückgegriffen werden.

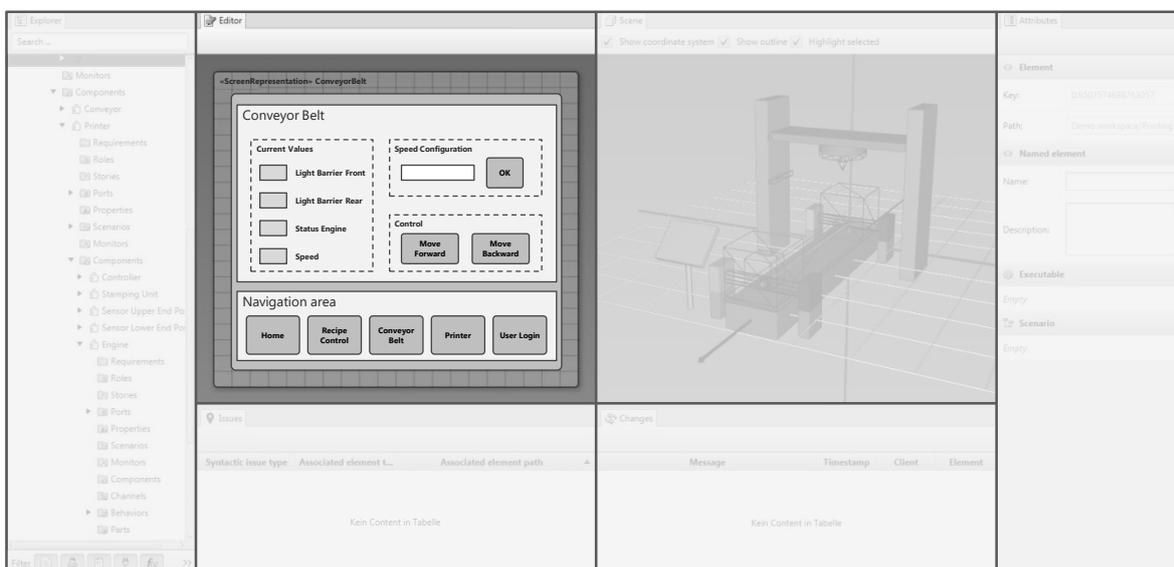


Abbildung 6-2: Umsetzung der Interaktionselemente in MACON

Neben der Modellierungstechnik wurde auch die Modelltransformation zur Generierung von HMI-Programmen in MACON prototypisch umgesetzt. Die dafür implementierten Klassen sind in dem Modul *Transformation* integriert und können auf JSON-Dokumente zugreifen, in denen die einzelnen mechatronischen Konzepte verwaltet werden. Über die in den Klassen integrierten Transformationsvorschriften

können die einzelnen Bausteine von HMI-Programmen gemäß dem in Abschnitt 5.4.1 beschriebenen Konzept generiert werden. Wie bereits dort erklärt, hängen die konkreten Transformationsvorschriften von der jeweiligen Entwicklungsumgebung ab. Daher beschränkt sich die in dieser Arbeit umgesetzte Modelltransformation auf die Generierung von HMI-Programmen für das .NET-Framework mit der darin enthaltenen Klassenbibliothek „Windows Presentation Foundation“ (WPF), die zur Umsetzung der Mensch-Maschine-Schnittstelle für das Fallbeispiel gewählt wurde (vgl. Abschnitt 6.3). Die für diese Zielumgebung implementierten Transformationsvorschriften werden bei der Anwendung der Entwicklungsmethodik an dem gewählten Fallbeispiel noch im Detail erklärt.

Aufgrund des engen Zusammenhangs zwischen der Modellierung und Qualitätssicherung von mechatronischen Systemen wurde weiterhin das Ziel verfolgt, auch die konzipierten Simulationstechniken in MACON umzusetzen. Zur Implementierung der Dialogtests konnten dabei die bereits für den Test von Szenarien vorhandenen Algorithmen innerhalb des Moduls *Simulator* als Basis verwendet und entsprechend erweitert werden, um alle Aspekte der konzeptionierten Dialogtests (vgl. Abschnitt 5.5.1) in der Simulation berücksichtigen zu können. Gleichzeitig wurde auch die spezifische Simulationsansicht entsprechend erweitert, um u. a. die bei einer Simulation jeweils aktiven Entitäten hervorzuheben. Demgegenüber konnten die konzeptionierten Nutzerevaluationen nicht in MACON umgesetzt werden, was auf die grundsätzliche Funktionsweise des Moduls *Simulator* sowie der darin eingesetzten Frameworks (z. B. JUnit) zurückzuführen ist. So werden im *Simulator* alle notwendigen Berechnungen für einen gesamten Simulationslauf a priori und nicht in Realzeit durchgeführt, erst anschließend werden die Ergebnisse dem Benutzer visualisiert. Dieses Prinzip verhindert den für Nutzerevaluationen notwendigen Eingriff eines Nutzers in das System zur Simulationslaufzeit. Vor diesem Hintergrund wäre für eine Implementierung der Nutzerevaluationen ein eigenes Softwarewerkzeug erforderlich, was allerdings die Durchgängigkeit in der Nutzung und somit die Gebrauchstauglichkeit des Konzepts einschränken würde. Daher wurde bewusst entschieden, im Rahmen dieser Arbeit auf eine Softwareumsetzung der Nutzerevaluationen zu verzichten. Um dennoch das Konzept der Nutzerevaluationen an dem gewählten Fallbeispiel anwenden zu können, wurde auf das in Abschnitt 2.4.3 eingeführte Konzept des Paper Prototyping zurückgegriffen. Wie die Nutzerevaluationen anhand dieses Konzepts umgesetzt wurden, wird bei der Anwendung der Entwicklungsmethodik an dem gewählten Fallbeispiel in Abschnitt 6.3 entsprechend erläutert.

6.2.2 Umsetzung der Einflussmatrix

Den einzigen Lösungsbaustein, der außerhalb des Werkzeugs MACON umgesetzt wurde, bildet die Einflussmatrix zur Gegenüberstellung von Anwendungsfällen und Interaktionstechnologien. Wie bereits bei der Vorstellung des Konzepts in Abschnitt 5.3 deutlich wurde, ist die Auswahl einer geeigneten Interaktionstechnologie weitestgehend losgelöst von den übrigen Entwicklungsaktivitäten, weswegen auch die Umsetzung in einem eigenständigen Softwareprototyp realisiert werden konnte. Durch den Aufbau der Einflussmatrix sowie dem zugehörigen Bewertungsschema bietet sich als Grundlage für die Umsetzung ein Tabellenkalkulationsprogramm an, wobei im konkreten Fall Microsoft® Excel gewählt wurde.

Innerhalb dieses Werkzeugs wurde die erarbeitete Einflussmatrix analog zu deren Aufbau aus der Konzeption in einem Tabellenblatt abgebildet. Abbildung 6-3 zeigt einen Ausschnitt der Umsetzung der Einflussmatrix, in dem neben den statischen Abhängigkeiten zwischen ausgewählten Auswahlfaktoren und Klassifikationskriterien auch die Felder für das Eintragen eines konkreten Anwendungsfalls sowie zur Darstellung der Bewertungsergebnisse visualisiert sind. Für die Bestimmung der für einen konkreten Anwendungsfall notwendigen bzw. geeigneten Ausprägungen der Klassifikationskriterien sind vom Anwender des Softwarewerkzeugs lediglich die Gewichtungen der Auswahlfaktoren festzulegen sowie deren zutreffende Ausprägungen anzukreuzen. Im Bereich der Auswertung wird dabei kontinuierlich der Eignungsgrad bzw. die Eignungsbedingung pro Klassifikationskriterium berechnet.

Gruppe	Technologieklassen				Hardware			Interaktionsgeräte						
	Stakeholder	Gewichtung	Auswahlfaktoren	Zutreffende Ausprägung(en)	Klassifikationskriterien		Ausprägungen			Informationsrichtung				
					Portabilität	Latenzzeit	Mobil	Stationär	Keine Echtzeit	Weiche Echtzeit	Harte Echtzeit	Eingabe	Ausgabe	Ein- & Ausgabe
Kundenseitige Stakeholder	Bediener	2	Interaktionsart	Beobachten								NEIN	++	-
				x Bedienen								++	NEIN	-
				Beobachten und Bedienen								+	+	++
	Bediener	3	Bewegungsradius	< 1m	-	+								
				1 - 10m	0	0								
				x 10 - 100m	+	-								
				> 100m	++	--								
	Bediener	1	Interaktionshäufigkeit	<10 p. Tag	-	+	0	0	0					
				10 - 50 p. Tag	0	0	-	0	+					
				x >50 p. Tag	+	-	--	+	++					
Prod.-Leiter	1	Geforderte Datensicherheit	gering	0	0									
			x mittel	-	+									
			hoch	--	++									
Auswertung					3	-3	-2	1	2	4	NEIN	-4		

Abbildung 6-3: Ausschnitt der umgesetzten Einflussmatrix in MS Excel

6.3 Anwendung an einem industrienahen Fallbeispiel

Auf Basis der Umsetzung der technischen Lösungsbausteine in den beschriebenen Softwarewerkzeugen konnte die Entwicklungsmethodik an einem repräsentativen Fallbeispiel angewendet werden. Dieses Beispiel wird zunächst in Abschnitt 6.3.1 eingeführt, wobei bereits ein Überblick der einzelnen Benutzeraufgaben sowie der eingesetzten Interaktionstechnologien dargestellt wird. Darauf aufbauend erfolgt in den Abschnitten 6.3.2 bis 6.3.5 die eigentliche Anwendung der Methodik gemäß den definierten Entwicklungsaktivitäten. Wichtig ist zu erwähnen, dass aufgrund der Interdisziplinarität des gesamten Ansatzes auch Aspekte der gesamten Systementwicklung betrachtet werden, um die Wechselwirkungen zwischen den einzelnen Disziplinen herauszustellen. Der Fokus liegt allerdings speziell auf der Konzeption und Umsetzung der Mensch-Maschine-Schnittstelle des Fallbeispiels.

6.3.1 Einführung des Fallbeispiels

Wie bereits in Abschnitt 6.1 beschrieben, wurde bei der Identifikation eines Fallbeispiels auf einen Kompromiss zwischen einem angemessenen Komplexitätsgrad zur schnellen und einfachen Ergebnisdemonstration und einem aus Industriesicht möglichst repräsentativen Szenario für den Transfer der Erkenntnisse in die Praxis geachtet. Als Grundlage wurde daher ein Modell einer Einrichtung zum Stempeln von Werkstücken gewählt, deren mechanischer Grundaufbau sowie die Aktorik und Sensorik mit Fischertechnik realisiert sind. Die Steuerung erfolgt dabei über eine SPS, in diesem Fall eine CX9020 der Firma Beckhoff. Wie in Abbildung 6-4 deutlich wird, setzt sich das Modell aus einem Förderband mit zwei Lichtschranken sowie dem eigentlichen Stempel mit zwei eigenen Endlagenschaltern zusammen. Wird ein Werkstück von einem Werker an der Eingabeposition auf das Förderband gelegt, wird es automatisch zur Stempelposition transportiert, dort gestempelt und anschließend an die Eingabeposition zurückgebracht, von der es schließlich entnommen werden kann. Da die Möglichkeiten zur Nutzerinteraktion bei diesem Vorgang grundsätzlich sehr eingeschränkt sind, wurde die Einrichtung im virtuellen Bereich entsprechend erweitert, um eine Reihe typischer Aufgaben für die Benutzerinteraktion mit Maschinen und Anlagen berücksichtigen zu können. Diese Erweiterungen werden nachfolgend kurz zusammengefasst, in Abbildung 6-4 wird aber bereits deutlich, welche unterschiedlichen Interaktionstechnologien zur Umsetzung des gesamten Aufgabenspektrums zum Einsatz kommen.

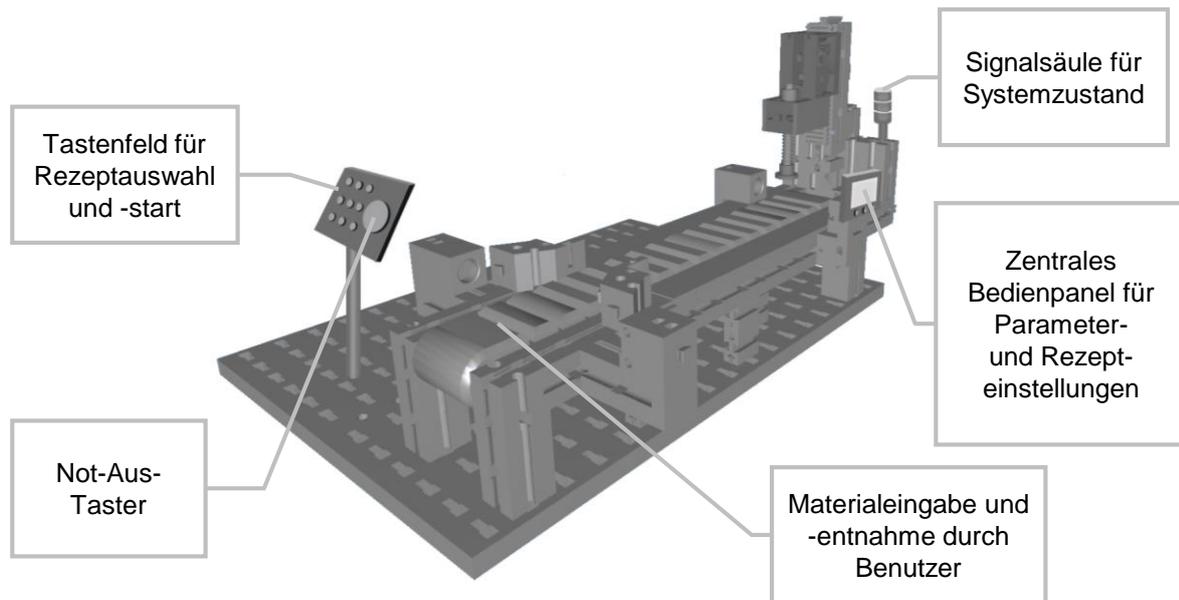


Abbildung 6-4: Stempelinrichtung mit eingesetzten Interaktionstechnologien

Eine erste Erweiterung bezieht sich auf die Erstellung und Verwaltung von Rezepten an einem zentralen Bedienpanel, um für bestimmte Werkstücke die Geschwindigkeit des Förderbands, den Stempelhub sowie die Stempelkraft in Rezepten zu hinterlegen. Wenn ein Werker ein konkretes Werkstück auf das Förderband legt, kann er am Tastenfeld das entsprechende Rezept auswählen und den Bearbeitungsvorgang mit den für das Rezept hinterlegten Parametern starten. Das zentrale Bedienpanel bietet zudem die Möglichkeit, sich als Benutzer anzumelden, aktuelle Sensor- und Aktorwerte zu visualisieren, Fehler- und Warnmeldungen anzuzeigen und zu quittieren, zwischen unterschiedlichen Betriebsarten zu wechseln sowie in einem Einrichtbetrieb manuell Aktoren zu betätigen. Darüber hinaus existieren ein Not-Aus-Taster, um das System zu jedem Zeitpunkt anhalten zu können, sowie eine Signalsäule, die den aktuellen Systemzustand bzw. die Betriebsart visualisiert. Die Entwicklung dieser Mensch-Maschine-Schnittstelle mit den eben eingeführten Interaktionstechnologien unter Verwendung der erarbeiteten Entwicklungsmethodik wird in den folgenden Abschnitten beschrieben.

6.3.2 HMI-Modellierung

Gemäß dem spezifizierten Entwicklungsvorgehen des IMoMeSA-Ansatzes (vgl. Abschnitt 5.2.2) wurde bei der Konzeption des Fallbeispiels zuerst eine mechatronische Komponentenstruktur für die zu entwickelnde Stempelinrichtung aufgebaut. Aus HMI-Sicht wurden während dieser Phase die einzelnen Aufgaben der Benutzer anhand der Modellierungselemente Benutzerrollen, User Stories, HMI-Ports und

Benutzerdialoge spezifiziert. Dabei wurden zunächst auf Systemebene die mit der Stempelinrichtung interagierenden *Benutzerrollen* festgelegt. Für das Fallbeispiel wurden ausgehend von der generischen Rolle „Systembenutzer“ die drei zentralen Rollen „Werker“, „Schichtleiter“ und „Instandhalter“ definiert, die jeweils ein eigenes Aufgabenspektrum im Umgang mit der Stempelinrichtung besitzen. Um zudem innerhalb des Fallbeispiels die Möglichkeit abzubilden, dass eine Benutzerrolle nur für einen speziellen Bereich eines Systems zuständig ist, wurde weiterhin die Rolle „Instandhalter Stempel“ eingeführt, der ausschließlich für den Austausch eines defekten Stempels verantwortlich ist und dementsprechend nicht der Systemebene, sondern der mechatronischen Komponente „Stempel“ zugeordnet ist.

Für jede der definierten Benutzerrollen konnten im Verlauf der Entwicklung die jeweiligen Benutzeraufgaben spezifiziert werden, wobei zunächst die informellen *User Stories* zum Einsatz kamen. Exemplarische Aufgaben des Werkers liegen bspw. in der Bereitstellung eines Materials auf dem Förderband, der Auswahl eines passenden Rezepts sowie dem Start des Bearbeitungsvorgangs. Dem Schichtleiter kommen dagegen erweiterte Aufgaben wie bspw. zur Einstellung von Systemparametern oder dem manuellen Verfahren von Aktoren zu. Der Instandhalter ist schließlich für das Beheben von Fehlern und den Austausch von mechanischen Komponenten verantwortlich und benötigt dazu einen entsprechenden Überblick über aktuelle und vergangene Sensorwerte. Neben diesen speziellen Aufgaben wurden auch User Stories definiert, die für alle Benutzer relevant sind und daher der generischen Rolle „Systembenutzer“ zugeordnet wurden. Dies betrifft bspw. die Aufgaben, das System im Falle eines kritischen Fehlers zum Stillstand zu bringen oder sich generell über den aktuellen Betriebszustand des Systems zu informieren. Je nach Aufgabe wurden die modellierten User Stories entlang der mechatronischen Komponentenstruktur zugeordnet. Während übergeordnete Aufgaben, wie z. B. die Rezeptauswahl oder der Start des Bearbeitungsvorgangs der Systemebene zugeordnet wurden, konnten spezifische Aufgaben, wie bspw. das Einstellen der Geschwindigkeit des Förderbands auch innerhalb einzelner mechatronischer Komponenten modelliert werden. Zudem wurden voneinander abhängige User Stories miteinander verknüpft, wie dies bspw. bei den User Stories zur Visualisierung und Einstellung des Betriebszustands der Fall ist. Abbildung 6-5 zeigt schließlich einen Ausschnitt der im Editor-Bereich von MACON auf Systemebene modellierten User Stories sowie die Detailansicht der angewählten User Story im Attribut-Bereich. Man erkennt dabei im Attribut-Bereich die Möglichkeit, eine Benutzerrolle für die User Story auszuwählen, wohingegen im Bereich des Ziels und des Nutzens entsprechende Freitexteingaben möglich sind.

Name	Role	Goal	Benefit
Adjust system ...	Shift Supervisor	Adjust all rel...	Optimize the pr...
Error Handling	Shift Supervisor	Quick overvi...	Determination ...
Insert Material	Operator	Insert raw m...	Get the material...
Maintenance	Service Engineer	Overview of ...	Determination ...
Move Actors	Shift Supervisor	Manual mov...	Check the right ...
Operation Mode	User	See the curre...	know what the ...
Remove Material	Operator	Remove print...	A new printing ...
Select recipe	Operator	Select the rig...	The workpiece i...
Start Process	Operator	Start the ma...	Get the material...

Story

Role:

Goal:

Benefit:

Abbildung 6-5: Ausgewählte User Stories des Fallbeispiels auf Systemebene

Die abgebildeten User Stories konnten daraufhin als Basis für die Formalisierung in *HMI-Ports* und *Benutzerdialogen* verwendet werden. Dabei wurden zunächst die unterschiedlichen Ein- und Ausgabemöglichkeiten in Form von Material-, Energie- und Datenports modelliert und den jeweiligen mechatronischen Komponenten zugeordnet. Über die Materialports wurden dabei die Eingabe- und Entnahmeposition von Bauteilen definiert, während über die Energieports bspw. ein Not-Aus-Signal oder der Hauptschalter des Förderbandantriebs abgebildet wurden. Den Großteil der modellierten Interaktionsmöglichkeiten des Fallbeispiels bildeten aber Datenports, die u. a. für den Systemstart, die Rezeptauswahl, die Anzeige und Einstellung des Betriebszustands oder Parametereinstellungen benötigt werden. Neben der Spezifikation der HMI-Ports anhand einer ID, eines Namens und eines Datentyps wurden den HMI-Ports auch die jeweils relevanten Benutzerrollen zugeordnet, um bereits auf Modellebene die grundsätzlichen Rechte einzelner Benutzer im Umgang mit der Stempelmaschine festzulegen. Darüber hinaus bildeten die HMI-Ports die Basis für die formelle Abbildung der einzelnen Benutzeraufgaben mit Benutzerdialogen. Eine zentrale Aufgabe besteht bspw. in der Durchführung eines Stempelvorgangs durch die Rolle „Werker“, der ein Bauteil an der definierten Eingabeposition in das System einlegt, anschließend ein definiertes Rezept auswählt, den Vorgang startet und am Ende das Bauteil entnimmt. Alle diese Schritte konnten in einem entsprechenden Dialog auf Systemebene unter Verwendung der zuvor modellierten HMI-Ports abgebildet werden. Weitere Dialoge adressieren bspw. das manuelle Verfahren des Förderbands oder des Stempels, das Einstellen der Betriebsart oder das Auslösen des Not-Aus-Tasters. Analog zu den User Stories wurden auch die Benutzerdialoge entlang der mechatronischen Komponentenstruktur verteilt und mit den relevanten Benutzerrollen verknüpft. Abbildung 6-6 zeigt den modellierten Benutzerdialog für die Durchführung des Stempelvorgangs, wobei zum besseren Verständnis die einzelnen Transitionen kurz beschrieben sind.

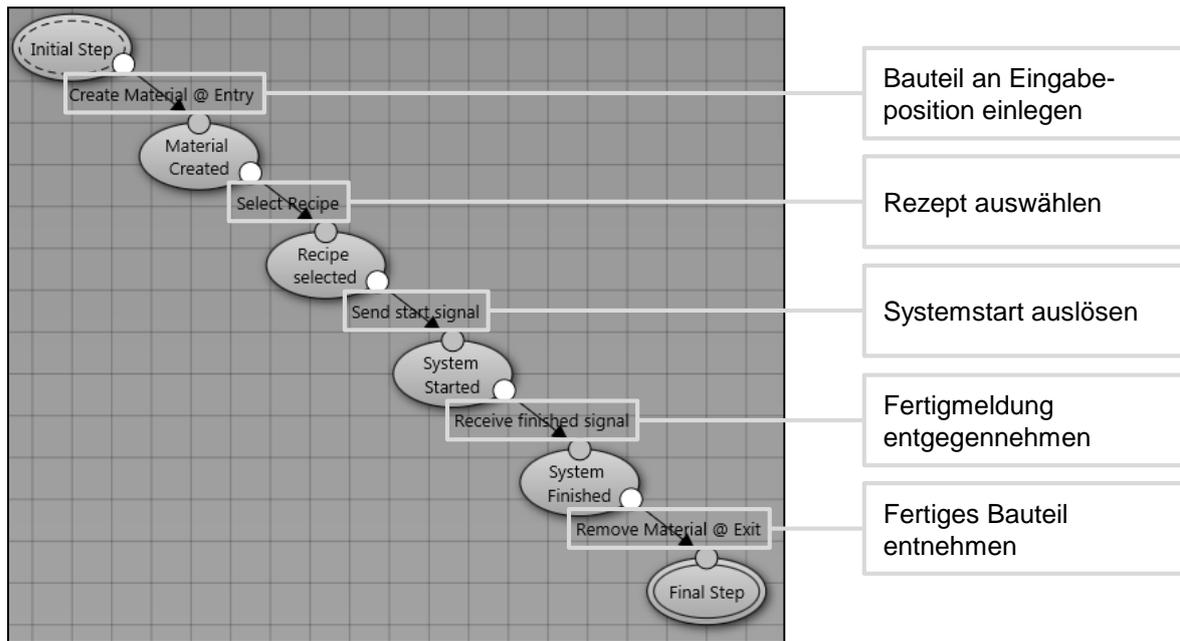


Abbildung 6-6: Benutzerdialog für den Stempelvorgang

Gemäß dem übergeordneten Entwicklungsvorgehen schließt sich an den Aufbau der mechatronischen Komponentenstruktur die Ausgestaltung der modellierten Komponenten an. Aus Sicht des HMIs wird hierbei mittels *HMI-Komponenten* und den integrierten *Verhalten* die Logik abgebildet, um die modellierten Benutzerdialoge zu realisieren. Für den in Abbildung 6-6 dargestellten Benutzerdialog des Stempelvorgangs wurde bspw. eine HMI-Komponente eingeführt, welche das Startsignal und das ausgewählte Rezept im Falle einer gültigen Rezeptnummer und der Betriebsart „Automatik“ an die Maschinensteuerung weiterleitet. Weitere HMI-Komponenten wurden im Hinblick auf das Fallbeispiel u. a. für den Signalaustausch mit der Maschinensteuerung (einzustellende Parameter, anzuzeigende Sensorwerte, etc.), für die Realisierung der Not-Aus-Funktionalität oder die Verwaltung der Betriebszustände eingeführt. Wichtig ist an dieser Stelle zu erwähnen, dass während der Modellierung der HMI-Komponenten auch neue Benutzeraufgaben identifiziert werden konnten, die anschließend mit den zuvor eingeführten Modellierungselementen abgebildet und mit eigenen HMI-Komponenten umgesetzt werden konnten. Im vorliegenden Beispiel betraf dies vor allem die Verwaltung von Rezepten, die sich erst während der Modellierung des Förderbands und des Stempels als zusätzliche Aufgabe für den Schichtleiter herausstellte.

Durch die Integration der HMI-Komponenten konnten alle bis hierhin spezifizierten Benutzeraufgaben erfolgreich umgesetzt werden, was durch entsprechende Dialogtests nachgewiesen werden konnte (vgl. Abschnitt 6.3.5). Allerdings beschränkte

sich die Funktionalität des HMIs sowie des gesamten mechatronischen Systems zu diesem Entwicklungszeitpunkt auf den Sollablauf ohne die Berücksichtigung etwaiger Störungen. Gemäß dem definierten Prozessablauf wurde daher in einem nächsten Schritt das mechatronische Konzept um potenzielles Fehlverhalten erweitert. Aus Sicht des HMIs kamen dadurch neue Benutzeraufgaben hinzu, die sich der Behandlung der Verhaltensfehler widmen, wie bspw. dem Austausch eines defekten Stempels mit anschließender Quittierung. Diese Aufgaben wurden mit User Stories abgebildet, über HMI-Ports und Benutzerdialoge formalisiert sowie mit entsprechenden HMI-Komponenten und Verhalten umgesetzt. In einem zweiten Schritt wurde auch ein mögliches Fehlverhalten einzelner Benutzerrollen durch die Adaption bestehender Benutzerdialoge analysiert. Dabei wurden u. a. die Fehler einer ungültigen Rezepteingabe oder eines Startbefehls während einer laufenden Bearbeitung in bestehende Benutzerdialoge integriert. Mittels Dialogtests wurde daraufhin evaluiert, wie das System auf die eingebrachten Fehler reagiert. Die generierten Erkenntnisse konnten schließlich genutzt werden, um bestehende HMI-Komponenten in ihrer Funktionalität zu erweitern, um dadurch einen robusten Systemablauf zu ermöglichen.

Nach der Erweiterung des mechatronischen Konzepts für die Stempelinrichtung um potenzielles Fehlverhalten lag aus HMI-Sicht ein vollständiges Modell der Benutzeraufgaben sowie der Logik der Mensch-Maschine-Schnittstelle vor. Auf dieser Basis wurden in einem letzten Schritt die Interaktionselemente modelliert, die zur Umsetzung der einzelnen Aufgaben benötigt werden. Welche Interaktionselemente dabei grundsätzlich zum Einsatz kommen, wurde durch Anwendung der erarbeiteten Einflussmatrix entschieden, worauf im nachfolgenden Abschnitt 6.3.3 noch im Detail eingegangen wird. Generell wurde aber festgelegt, die Umsetzung der Benutzeraufgaben auf mehrere Interaktionstechnologien aufzuteilen, wie bereits in Abbildung 6-4 deutlich wurde. Für jede der dort abgebildeten Technologien wurde in MACON ein eigenes Interaktionselement mit einer Objekt-Repräsentation und teilweise entsprechenden Screen-Repräsentationen modelliert. Abbildung 6-7 zeigt beispielhaft die zwei Repräsentationen des Interaktionselements „Tastenfeld“. Auf der Screen-Repräsentation erkennt man dabei die einzelnen Knöpfe anhand derer ein Benutzer ein Rezept auswählen und einen Bearbeitungsvorgang starten kann. Diese Bedienelemente wurden über den Editorbereich weiterhin mit den jeweiligen HMI-Ports (Systemstart, etc.) verknüpft, um hierüber den Bezug zu den HMI-Komponenten herzustellen. Dieser Schritt wurde auch für die weiteren modellierten Interaktionselemente durchgeführt, wodurch die Konzeption des Fallbeispiels aus HMI-Sicht abgeschlossen werden konnte.

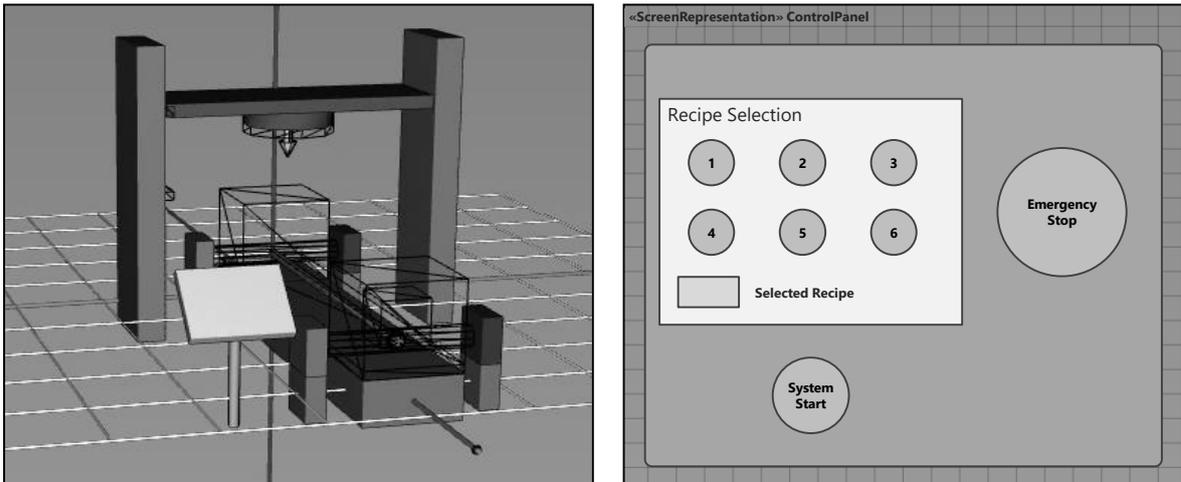


Abbildung 6-7: Objekt- und Screen-Repräsentation des Interaktionselements „Tastenfeld“

6.3.3 Auswahl von Interaktionstechnologien

Wie bereits im letzten Abschnitt dargelegt, wurde noch vor der Modellierung der Interaktionselemente eine Auswahl von Interaktionstechnologien anhand der erarbeiteten Einflussmatrix durchgeführt. Gemäß dem spezifizierten Vorgehen wurden dabei zunächst die relevanten Stakeholder für das Fallbeispiel identifiziert und die jeweiligen Auswahlfaktoren entsprechend priorisiert. Obwohl für das gewählte Fallbeispiel kein realer Kunde und Hersteller existieren, wurden für alle Kategorien fiktive Stakeholder eingeführt, um eine möglichst ganzheitliche Betrachtung bei der Technologieauswahl zu ermöglichen. Allerdings wurde bereits auf dieser Ebene eine grundsätzliche Priorisierung vorgenommen, bei der u. a. entschieden wurde, dass auf Kundenseite die Auswahlfaktoren der Benutzer ein höheres Gewicht als diejenigen des Managements oder des Einkaufs besitzen sollen. Auch hinsichtlich der Rahmenbedingungen (vgl. Abbildung 5-11) wurden für die Kategorien Normung sowie Umfeld bzw. Umwelt fiktive, aber für den Maschinen- und Anlagenbau typische Konstellationen gewählt, weswegen auch diese Kategorien mit einer generell hohen Priorisierung berücksichtigt wurden. Auf Basis dieser Vorüberlegungen konnten die Auswahlfaktoren von allen Stakeholdern und Rahmenbedingungen entsprechend gewichtet werden. In Abbildung 6-8 sind die Ergebnisse der Priorisierung für exemplarische Auswahlfaktoren dargestellt.

In einem zweiten Schritt konnte die eigentliche Klassifikation des Anwendungsfalls durch Bestimmung der geeigneten Ausprägungen innerhalb der Auswahlfaktoren durchgeführt werden. Bei einigen Auswahlfaktoren zeigte sich allerdings, dass die geeignete Ausprägung vom Spektrum der betrachteten Benutzeraufgaben abhängt,

wie z. B. beim Interaktionsradius der Benutzer oder bei der Aufgabenkomplexität. Daher wurde entschieden, das Spektrum der in MACON modellierten Benutzeraufgaben in vier Teilmengen zu untergliedern und die Technologieauswahl jeweils separat durchzuführen. Neben der gekapselten Betrachtung der übergeordneten Aufgaben zur Visualisierung des Systemzustands und zur Not-Aus-Funktionalität wurden die Aufgaben des Werkers von denen des Schichtleiters und des Instandhalters separiert betrachtet. Abbildung 6-8 zeigt für die Aufgaben der zuletzt genannten Benutzerrollen die zutreffenden Ausprägungen für beispielhafte Auswahl-faktoren. Ferner sind auf der rechten Seite der Abbildung die aus der Einflussmatrix erhaltenen Ausprägungen ausgewählter Klassifikationskriterien dargestellt, die bei der Auswahl einer Interaktionstechnologie für das betrachtete Aufgabenspektrum des Anwendungsbeispiels zu berücksichtigen sind.

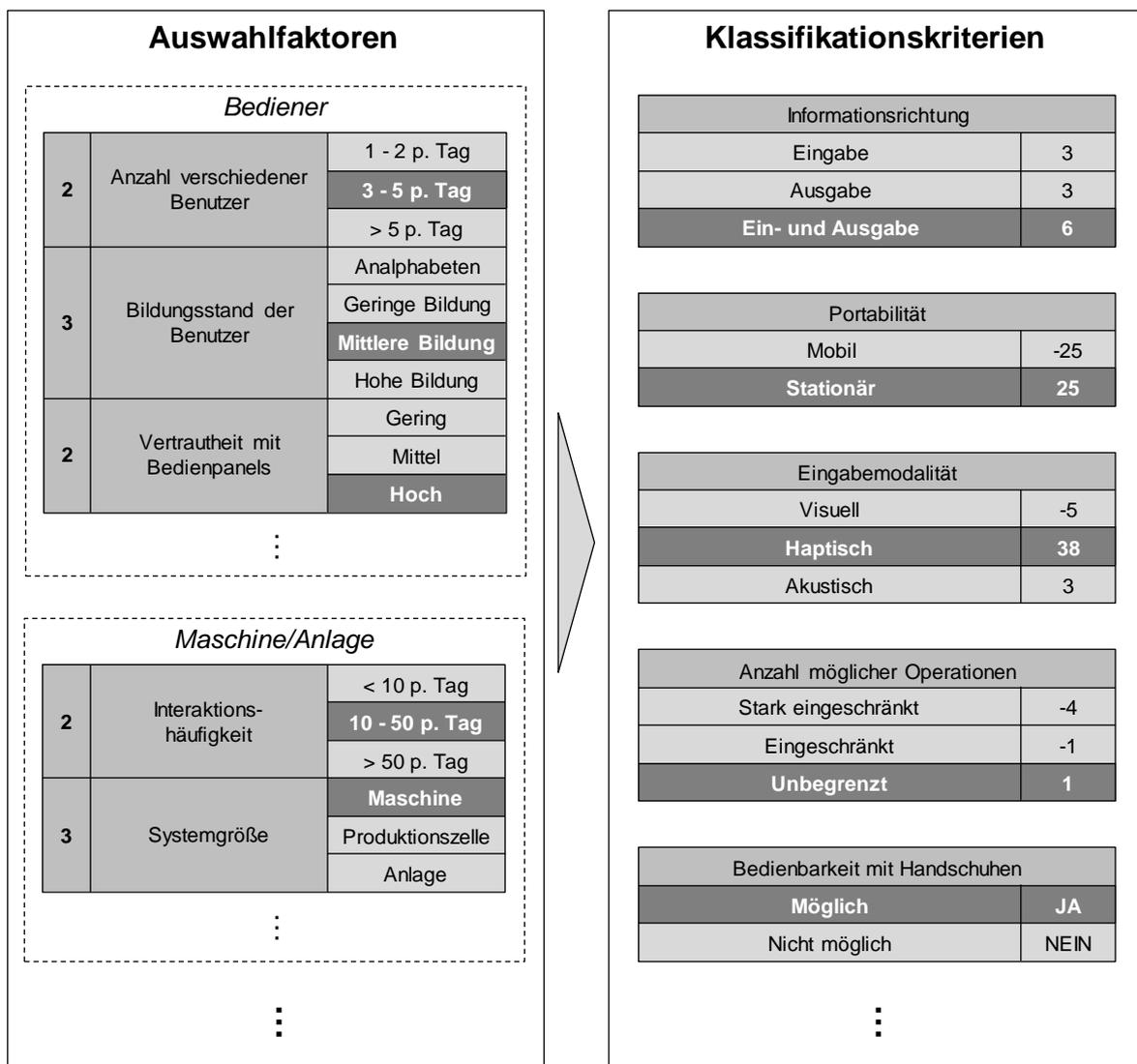


Abbildung 6-8: Exemplarische Auswahlfaktoren und Klassifikationskriterien mit den jeweils zutreffenden Ausprägungen für das Fallbeispiel

So wird bspw. der Einsatz einer stationären Interaktionstechnologie für die Aufgaben des Schichtleiters und Instandhalters empfohlen, was u. a. aus dem Zugriff verschiedener Personen auf die Interaktionstechnologie und der geforderten Nähe des Benutzers zum System resultiert. Ferner wird deutlich, dass für den konkreten Anwendungsfall eine haptische Eingabe sowie eine visuelle Ausgabe zielführend sind, was durch zahlreiche Anzeigen und regelmäßige Eingaben numerischer Parameter oder die hohe Vertrautheit der Benutzer mit dieser Ein- und Ausgabeform zu begründen ist. Mit der Summe der generierten Ausprägungen, von denen auch einige weitere in Abbildung 6-8 dargestellt sind, konnte die Technologieauswahl durchgeführt werden. Dabei wurden unterschiedliche Lösungen mehrerer Hersteller recherchiert und hinsichtlich ihrer Abdeckung mit den geforderten Ausprägungen abgeglichen. Als Ergebnis wurde ein konkretes Touch-Panel eines Herstellers von Automatisierungslösungen ausgewählt, welches die größte Überschneidung hinsichtlich der geforderten Ausprägungen aufweist.

Auf analoge Weise konnte auch die Technologieauswahl für die weiteren Teilmengen der Benutzeraufgaben durchgeführt werden. Auf Basis der jeweils mit der Einflussmatrix generierten Ausprägungen wurden ein konkreter Not-Aus-Taster, eine Signalleuchte zur Visualisierung des Maschinenzustands sowie ein Tastenfeld für die Benutzeraufgaben des Werkers ausgewählt. Darüber hinaus konnte die erarbeitete Einflussmatrix auch verwendet werden, um die Auswahl einer Softwarelösung für das Bedienpanel zu unterstützen. So wurde bspw. anhand der Auswahl-faktoren „Gestaltungsfreiheit“ und „Know-how der Entwickler“ ermittelt, eine individualisierbare und hochsprachenbasierte Software einzusetzen. Aus der Kombination der generierten Ausprägungen und dem bereits ausgewählten Bedienpanel wurde entschieden, eine Applikation unter Verwendung des WPF-Frameworks von Microsoft (vgl. NATHAN 2014) sowie der Entwicklungsumgebung Visual Studio umzusetzen. Das Ergebnis dieser Implementierung wird im folgenden Abschnitt im Detail betrachtet.

6.3.4 HMI-Implementierung

Basierend auf der Modellierung einer Mensch-Maschine-Schnittstelle innerhalb des mechatronischen Konzepts sowie der Auswahl geeigneter Interaktionstechnologien sieht die erarbeitete Entwicklungsmethodik die HMI-Implementierung vor. Dies bezieht sich allerdings lediglich auf Interaktionstechnologien, für deren Umsetzung eine Software benötigt wird. Für das Fallbeispiel betraf dies ausschließlich das graphische Bedienpanel mit der zugehörigen WPF-Applikation.

Für diese Applikation sind in einem ersten Schritt die auf Modellebene abgebildeten Inhalte in die Entwicklungsumgebung Visual Studio zu übertragen. Hierfür wurde basierend auf dem allgemeingültigen Konzept der Modelltransformation (vgl. Abschnitt 5.4.1) ein spezifischer Transformator für WPF-Applikationen umgesetzt. Dessen Funktionalität wird nachfolgend anhand des Fallbeispiels erläutert, wobei zum besseren Verständnis zuvor die Softwarearchitektur in der Zielumgebung beschrieben wird (vgl. Abbildung 6-9). Den Kern dieser Architektur bildet das sog. MVVM-Pattern (GAROFALO 2011), welches speziell die Trennung von Logik und Darstellung unterstützt und dazu die Schichten *Model*, *View* und *ViewModel* definiert. Jede dieser Schichten ist in der Zielumgebung in einer eigenen Projektdatei gekapselt, wobei eine gemeinsame Projektmappe den Rahmen bildet. Die Kapselung erfolgt dabei vor dem Hintergrund, dass in jeder Schicht typischerweise spezielle Frameworks eingebunden sind, die nur für die jeweilige Schicht relevant sind. In der *View* sind zunächst die graphischen Informationen der HMI-Ansichten in sog. XAML-Dateien abgebildet, wobei jede Ansicht in einer XAML-Datei gekapselt ist. Über Datenbindungen ist die *View* mit dem *ViewModel* verknüpft, in der die Logik der Mensch-Maschine-Schnittstelle abgebildet ist, wobei üblicherweise für jede XAML-Datei eine eigene Klasse definiert ist. Die dritte Schicht bildet schließlich das *Model*, welches in einzelnen Klassen übergreifende Methoden und Dienste für das *ViewModel*, wie bspw. für eine Datenbankanbindung oder für komplexe Berechnungen, bereitstellt. Zu jeder Schicht gehören zudem weitere Bausteine, wie Konfigurationsdateien oder globale Ressourcen, die in Abbildung 6-9 als „weitere Bausteine“ zusammengefasst sind. Eine Detailbetrachtung dieser Bausteine ist allerdings nicht notwendig, da sie für die spätere Erläuterung der Modelltransformation nicht relevant sind.

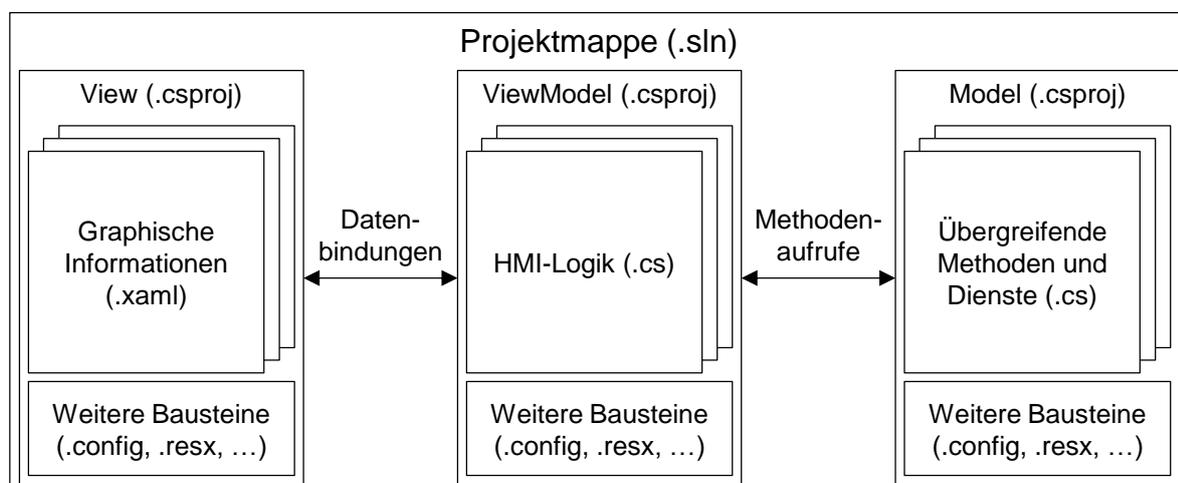


Abbildung 6-9: Architektur einer WPF-Applikation mit dem MVVM-Pattern

Zum Aufbau von WPF-Applikationen mit dem MVVM-Pattern werden meist spezielle Frameworks, wie bspw. das im Rahmen dieser Arbeit genutzte *MVVM light*-Toolkit (VERMEIR 2013, S. 32 ff.), eingesetzt. Diese Frameworks definieren spezielle Mechanismen, wie zur Datenbindung zwischen *View* und *ViewModel*, und unterstützen den Grundaufbau der Architektur von WPF-Applikationen mit dem MVVM-Pattern (vgl. Abbildung 6-9). Dieser Grundaufbau kann unabhängig vom Anwendungsfall genutzt und muss somit nicht automatisch generiert werden. Daher konzentriert sich der nachfolgend vorgestellte Modelltransformator auf die Generierung der einzelnen Bausteine gemäß dem in Abschnitt 5.4.1 eingeführten Konzept sowie deren Einbettung in den Grundaufbau einer WPF-Applikation.

In einem ersten Schritt der Modelltransformation werden die graphischen Elemente der Mensch-Maschine-Schnittstelle innerhalb der *View*-Schicht generiert. Hierbei wird für jede Screen-Repräsentation eines Interaktionselements eine eigene XAML-Datei generiert, die in eine generische Hauptansicht über einen sog. *Frame* eingebettet sind. In jede XAML-Datei werden anschließend die einzelnen Bedien- und Anzeigeelemente aus einer Screen-Repräsentation integriert und positioniert. Ferner werden in den XAML-Code dieser Elemente bereits die Datenbindungen zum *ViewModel* über das sog. *Binding*-Konzept (vgl. SELLS & GRIFFITHS 2007, S. 22 ff.) integriert. Zur Vervollständigung der *View* werden für jede XAML-Datei entsprechende Teilklassen als CS-Dateien generiert, die ausschließlich für die Initialisierung der XAML-Dateien verantwortlich sind.

Darauf aufbauend kann die *ViewModel*-Schicht erzeugt werden, in der die Logik der Mensch-Maschine-Schnittstelle abgebildet ist. Hierbei wird zuerst für jede XAML-Datei eine eigene *ViewModel*-Klasse generiert. In diese Klassen werden entsprechende Methoden integriert, welche die Logik der Zustandsautomaten aus dem mechatronischen Konzept abbilden. In diesen Methoden werden die in den XAML-Dateien spezifizierten Datenbindungen aufgegriffen, um eine Verknüpfung zur *View*-Schicht herzustellen. Auf der anderen Seite werden in den Methoden Dienste des *Model*-Bausteins zur Kommunikation mit der Steuerung genutzt, um ggf. Werte an die Steuerung zu übermitteln oder von dieser auszulesen. Zur Abbildung der Logik ist weiterhin eine Transformation der Zustandsautomaten aus den HMI-Komponenten notwendig, wofür in der Zielumgebung Enumerationen und *Switch-Case*-Anweisungen zum Einsatz kommen. Eine Besonderheit bildet schließlich die Logik zur Abbildung des Navigationskonzepts. Hierfür wird die im .NET-Framework enthaltene Klasse *NavigationService* genutzt, um bei einer Betätigung eines Navigationselements mittels der *Navigate()*-Methode zwischen den einzelnen XAML-Dateien zu wechseln.

Als drittes zentrales Element wird die *Model*-Schicht generiert, die sich aus den Bausteinen zum Benutzermanagement und zur Kommunikation mit der Steuerung zusammensetzt, für die jeweils eigene Klassen erzeugt werden. Innerhalb des zuletzt genannten Bausteins wird die Liste der Schnittstellenvariablen als *Member-Variable* mit einer verschachtelten Datenstruktur erzeugt. Ferner werden Methoden integriert, über die die einzelnen *ViewModel*-Klassen auf die Datenstruktur zugreifen und somit SPS-Variablen auslesen und beschreiben können. Für den eigentlichen Datenaustausch mit der SPS sind spezielle Methoden notwendig, die vom eingesetzten Kommunikationsprotokoll abhängen. Im vorliegenden Fallbeispiel wird eine SPS der Fa. Beckhoff genutzt, welche zur Kommunikation das proprietäre ADS-Protokoll unterstützt. Zu dessen Nutzung kann in Visual Studio eine spezielle Bibliothek eingebunden werden, deren relevante Methoden bei der Modelltransformation in den Kommunikationsbaustein integriert und mit der Datenstruktur der Schnittstellenvariablen verknüpft werden.

Den zweiten Teil dieser Schicht bildet die *Model*-Klasse „Benutzermanagement“, welche die An- und Abmeldung von Benutzern koordiniert und die Rechte der jeweils angemeldeten Benutzerrolle verwaltet. Neben der eigentlichen Klasse wird bei der Modelltransformation eine XML-Datei erzeugt, in der für jede Benutzerrolle die Sichtbarkeit der Bedien- und Anzeigeelemente spezifiziert ist. Diese Datei wird bei Bedarf (Benutzeranmeldung bzw. Benutzerwechsel) von der *Model*-Klasse aufgerufen und die für den neu angemeldeten Benutzer zutreffende Konfiguration wird geladen. Über Methoden wird die Konfiguration allen *ViewModel*-Klassen bereitgestellt, die dementsprechend die Sichtbarkeit ihrer zugehörigen Bedien- und Anzeigeelemente festlegen. Wichtig ist an dieser Stelle zu erwähnen, dass zur Umsetzung des Benutzermanagements neben der *Model*-Klasse auch eine XAML-Datei und eine zugehörige *ViewModel*-Klasse für die Benutzeranmeldung generiert werden. In der XAML-Datei werden dabei zwei Bedienelemente zur Auswahl und Bestätigung einer Benutzerrolle definiert. Die zugehörige *Viewmodel*-Klasse leitet eine ausgewählte Benutzerrolle an die *Model*-Klasse weiter, wo diese als *Member-Variable* während der HMI-Laufzeit abgespeichert wird.

Um die generierten Dateien in den bereits durch das *MVVM light*-Toolkit vorhandenen Grundaufbau zu integrieren, müssen in einem letzten Schritt der Modelltransformation noch die bestehenden Projektdateien der drei Schichten überschrieben werden. Dies bezieht sich allerdings ausschließlich auf das Referenzieren der generierten Dateien, während die bestehenden Konfigurationen der Projekte unverändert beibehalten werden können. Hierdurch entsteht eine lauffähige Applikation, die nachfolgend weiter detailliert werden kann.

Im Hinblick auf das Stempelmodul konnte mit dem eben eingeführten Modelltransformator ein initiales Programm für das Bedienpanel generiert werden, welches bereits zur Umsetzung der Benutzeraufgaben des Schichtleiters und Instandhalters genutzt werden kann. Aufgrund des vereinfachten Aufbaus dieses Programms war allerdings eine entsprechende Verfeinerung erforderlich. In Bezug auf den graphischen Aufbau der Mensch-Maschine-Schnittstelle wurden bspw. Header- und Footer-Bereiche eingefügt, in denen von der jeweiligen Ansicht unabhängige Informationen (z. B. eine Fehlerübersicht) sowie die Navigationselemente integriert wurden. Auf inhaltlicher Ebene wurde hingegen u. a. eine graphische Darstellung des Stempelkraftverlaufs eingebettet, die mit den einfachen Mechanismen auf Modellebene nicht realisiert werden konnte. Weiterhin wurde die Benutzeranmeldung um eine Passworteingabe ergänzt und es wurde eine Funktionalität zum Abspeichern und Einlesen von Rezepten als XML-Dateien integriert.

Neben der disziplinspezifischen Verfeinerung umfasst die Phase der Verfeinerung die Aktualisierung des Systemmodells bei interdisziplinär relevanten Änderungen und die erneute Generierung der betroffenen Bestandteile eines HMI-Programms. Um auch diesen Aspekt bei der Umsetzung des Bedienpanels zu erproben, wurde eine exemplarische Änderung in die Mechanik eingebracht, die sowohl für die Steuerung als auch die Benutzerinteraktion mit dem Stempelmodul Anpassungen erforderlich macht. Im Konkreten wurde entschieden, für kleine Werkstücke eine Positioniereinheit in das Förderband zu integrieren, die eine exakte Ausrichtung der Werkstücke an der Stempelposition ermöglicht. Da der Einsatz der Positioniereinheit vom jeweiligen Werkstück abhängt, wurden aus HMI-Sicht eine Anpassung der Rezeptverwaltung, eine Möglichkeit zum manuellen Ein- und Ausfahren und eine Anzeige des aktuellen Zustands der Positioniereinheit erforderlich. Neben der Modellierung der damit einhergehenden Benutzeraufgaben über User Stories, HMI-Ports und Benutzerdialoge wurden auch die Schnittstellen bestehender HMI-Komponenten erweitert sowie neue HMI-Komponenten in das Systemmodell integriert. Durch diese Änderungen wurde eine erneute Modelltransformation erforderlich, bei der zunächst die *Model*-Klasse zur Kommunikation mit der Steuerung erneut generiert und in der Zielumgebung überschrieben wurde. Ferner wurden diejenigen *ViewModel*-Klassen erneut generiert, die von der Anpassung der HMI-Komponenten betroffen waren. Allerdings wurden die generierten Klassen nicht unmittelbar in die Zielumgebung integriert, da der Code der vorhandenen *ViewModel*-Klassen bereits manuell angepasst wurde. Um diese Änderungen nicht zu überschreiben, wurden stattdessen die neuen bzw. angepassten Methoden direkt in die vorhandenen *ViewModel*-Klassen integriert.

Mit dem aktualisierten Programm konnte die WPF-Applikation für das Bedienpanel vollständig umgesetzt werden. In diesem Zuge wurde auch die Gebrauchstauglichkeit der Mensch-Maschine-Schnittstelle erneut getestet, worauf im folgenden Abschnitt im Detail eingegangen wird. Abschließend zeigt Abbildung 6-10 anhand einer exemplarischen Ansicht die einzelnen Ausbaustufen des Bedienpanels von der Screen-Repräsentation des mechatronischen Konzepts über das initiale Programm bis zum finalen Programm.

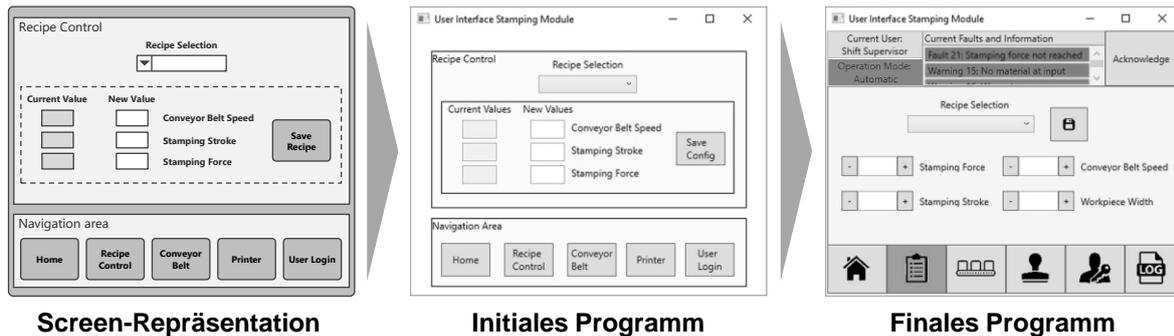


Abbildung 6-10: Ausbaustufen bei der Entwicklung des Bedienpanels

6.3.5 HMI-Qualitätssicherung

Zur Absicherung der Funktionalität und Gebrauchstauglichkeit der Mensch-Maschine-Schnittstelle für das Stempelmodul konnten während der Entwicklung die spezifizierten Maßnahmen zur Qualitätssicherung genutzt werden. Während der *Konzeption* kamen zunächst Dialogtests zum Einsatz, um zu überprüfen, ob die implementierten HMI-Komponenten eine Realisierung der spezifizierten Benutzerdialoge ermöglichen. In Anlehnung an das definierte Vorgehen wurden dabei zuerst die Komponenten der untersten Hierarchieebene anhand ihrer zugehörigen Dialoge im Sinne des Unit Testings erprobt. So konnten u. a. die Funktionen zum manuellen Verfahren des Förderbands oder zum Anzeigen einzelner Sensorwerte losgelöst vom Rest des Systems abgesichert werden. Durch diese Tests konnten erste Modellierungsfehler, wie eine noch fehlende Verknüpfung eines HMI-Ports zur Maschinensteuerung oder ein falsch implementierter Zustandsautomat der HMI-Komponente „Stempelkraft einstellen“ erkannt und behoben werden. Nachdem auf diese Weise alle HMI-Komponenten der untersten Hierarchieebene analysiert wurden, konnten umfangreichere Dialogtests auf höheren Hierarchieebenen bis hin zur Systemebene durchgeführt werden. Neben einem übergreifenden Dialogtest zur Inbetriebnahme des Stempelmoduls wurde auch der Bearbeitungsvorgang durch den Werker in einem Simulationslauf abgesichert, aus dem ein exemplarischer Screenshot in Abbildung 6-11 dargestellt ist.

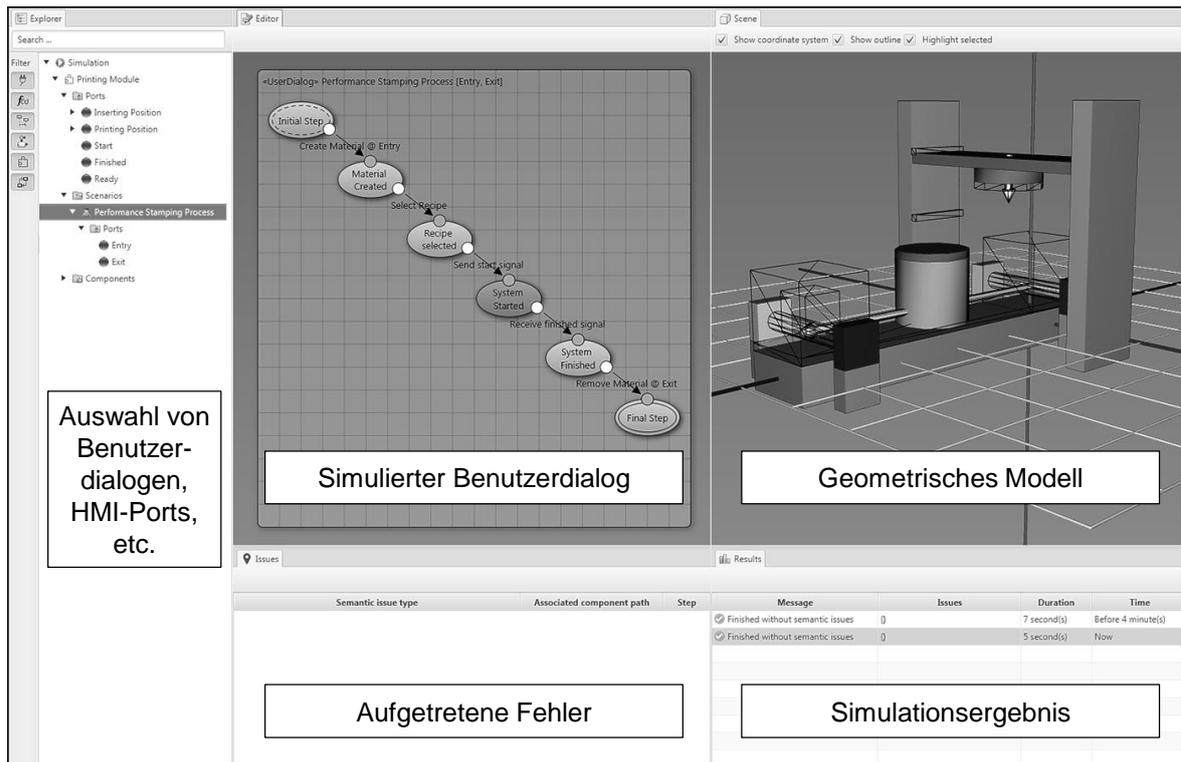


Abbildung 6-11: Exemplarischer Dialogtest des Stempelmoduls

Die abgebildete Simulationsansicht unterscheidet sich dabei nur geringfügig von der zuvor erläuterten Modellierungsansicht von MAISON. Im *Explorer* können alle für die Simulation relevanten Bestandteile eines Modells ausgewählt werden, die daraufhin im *Editor*-Bereich visualisiert werden. Dort wird bspw. der zum aktuellen Simulationszeitpunkt aktive Schritt innerhalb eines Benutzerdialogs angezeigt, es besteht aber auch die Möglichkeit, den Werteverlauf an einzelnen Ports oder den aktuellen Verhaltenszustand von Komponenten zu visualisieren. Ferner wird im *Scene*-Bereich die zum jeweiligen Simulationszeitpunkt aktuelle 3D-Ansicht der geometrischen Elemente visualisiert. Die Ergebnisse des Simulationslaufs werden schließlich in den beiden unteren Bereichen *Issues* und *Results* dargestellt, wobei evtl. aufgetretene Fehler im Detail analysiert werden können.

Im abgebildeten Fall ermöglichte das modellierte Verhalten des Stempelmoduls bereits eine korrekte Durchführung des gesamten Bearbeitungsvorgangs von der Materialeingabe über den Stempelvorgang bis zur Materialentnahme. Damit konnte über den Dialogtest nicht nur die korrekte Funktionsweise des HMIs, sondern auch aller weiterer Systemkomponenten von der Steuerung bis zur Mechanik nachgewiesen werden. Da durch diesen Test das mechatronische Konzept hinsichtlich seines Normalverhaltens vollständig abgesichert war, konnte anschließend mit der Erweiterung des Systems um mögliches Fehlverhalten fortgefahren werden.

In dieser Phase konnten erneut die Dialogtests genutzt werden, um zunächst die Auswirkungen eines eingebrachten Fehlers zu evaluieren und im weiteren Verlauf die modellierten Fehlerbehandlungsroutinen auf ihre korrekte Funktionalität hin zu überprüfen. So wurde u. a. der Fehler modelliert, dass der Stempel die erforderliche Kraft während des Bearbeitungsvorgangs nicht aufbringen kann. Zur Behandlung wurde ein sofortiger Stopp des Stempelmoduls, eine Meldung an den Werker sowie ein manuelles Ausschleusen des Werkstücks definiert. Neben der dafür benötigten Anpassung des Systemverhaltens, wobei auch die Mensch-Maschine-Schnittstelle zur Fehleranzeige und Quittierung erweitert wurde, musste auch der zugehörige Benutzerdialog entsprechend modifiziert werden. Durch einen weiteren Dialogtest konnte die korrekte Behandlung des Fehlers nachgewiesen werden. Neben Fehlern des Systems wurden anhand der Dialogtests auch fehlerhafte Eingaben einzelner Benutzerrollen analysiert. So wurde bspw. in einem Dialogtest festgestellt, ob die Mensch-Maschine-Schnittstelle beim manuellen Verfahren des Förderbands auch ein gleichzeitiges Ansteuern des Motors in beide Richtungen zulässt. Der hierbei erkannte Fehler konnte durch eine Erweiterung des Zustandsautomaten der zugehörigen HMI-Komponente behoben werden. Nachdem auf vergleichbare Weise weitere potenzielle Fehlbedienungen analysiert und behandelt wurden, konnte mit der Modellierung der Interaktionselemente und aus Sicht der Qualitätssicherung mit dem Einsatz der Nutzerevaluation fortgefahren werden.

Wie bereits in Abschnitt 6.2.1 erläutert, konnte dabei allerdings nicht auf das Tool MACON zurückgegriffen werden, da die Nutzerevaluation nicht mit den zur Simulation genutzten Frameworks umgesetzt wurde. Daher wurde an dieser Stelle auf einen „Paper Prototyping“-Ansatz (vgl. Abschnitt 2.4.3) zurückgegriffen, bei dem sich Nutzer anhand ausgedruckter Graphiken der Screen-Repräsentationen durch die Mensch-Maschine-Schnittstelle navigieren und dadurch einen Einblick in die Anordnung der Bedien- und Anzeigeelemente sowie den Aufbau des HMIs erhalten. Mit dieser Methode wurde die Gebrauchstauglichkeit des HMIs zwar nicht in einem vergleichbaren Detaillierungsgrad wie bei der vorgeschlagenen Nutzerevaluation analysiert, allerdings konnten für die Screen-Repräsentationen des Bedienpanels mehrere Verbesserungsmöglichkeiten hinsichtlich der Gebrauchstauglichkeit ermittelt werden. Dazu wurden verschiedenen Anwendern zunächst die generelle Funktionalität des Stempelmoduls sowie einzelne Benutzeraufgaben anhand der jeweiligen User Stories erläutert. Anschließend führten die Anwender diese Aufgaben anhand der ausgedruckten Screen-Repräsentationen durch und erklärten ihre Vorgehensweise und getätigten Aktionen. Zur Evaluation wurden die Anwender bei der Durchführung dieser Aufgaben von einem Usability-Experten

beobachtet und anschließend befragt. Durch diesen Ansatz konnten unterschiedliche Verbesserungspotenziale für die Implementierung des Bedienpanels erkannt und umgesetzt werden. Ein Beispiel einer umgesetzten Maßnahme bildete u. a. eine erweiterte Benutzerführung bei der Eingabe von Rezepten, bei der neben der Texteingabe von Parametern (bspw. für die Stempelkraft) auch die Möglichkeit zur Anpassung eines eingegebenen Werts über entsprechende Buttons geschaffen wurde. Nach Abschluss der vereinfachten Nutzerevaluationen mit dem gewählten „Paper Prototyping“-Ansatz stand ein vollständig abgesichertes mechatronisches Konzept des Stempelmoduls zur Verfügung, das anschließend mit den disziplinspezifischen Werkzeugen verfeinert werden konnte. Die dabei hinsichtlich der Qualitätssicherung durchgeführten Maßnahmen werden nachfolgend vorgestellt.

Grundsätzlich können diese Maßnahmen als Übertragung der Nutzerevaluationen auf die Phase der *Verfeinerung* verstanden werden. Im konkreten Anwendungsfall wurde die generierte bzw. später verfeinerte WPF-Applikation auf einem Desktop-PC ausgeführt und mit dem bereits zu diesem Zeitpunkt aufgebauten Stempelmodul über dessen SPS gekoppelt. Vor diesem Hintergrund war keine Co-Simulation notwendig, da die Funktionalität und Gebrauchstauglichkeit des Bedienpanels mit dem realen System erprobt werden konnte. Im Hinblick auf die Evaluationsmethode wurde der Ansatz aus der Konzeption übernommen. Die bereits zuvor ausgewählten Anwender führten unter Beobachtung eines entsprechenden Experten einzelne Benutzeraufgaben in der WPF-Applikation anhand der formulierten User Stories durch und wurden anschließend hinsichtlich potenzieller Verbesserungsmaßnahmen befragt. Diese Tests wurden bereits bei der ersten Generierung des HMI-Programms begonnen und wurden zu exemplarischen Zeitpunkten während der Umsetzung der WPF-Applikation wiederholt. Beispielhafte Verbesserungsmaßnahmen, die durch diese Erprobungen erkannt und anschließend umgesetzt wurden, betrafen die Optimierung des Navigationskonzepts durch die Einführung eines statischen Bereichs für alle Navigationselemente oder das automatische Einblenden eines numerischen Tastenfelds zur Eingabe von Parametern bei der Einstellung von Rezepten.

Nach der Umsetzung aller identifizierten Verbesserungspotenziale in der WPF-Applikation war die Phase der Verfeinerung für das Bedienpanel abgeschlossen. Da die weiteren Interaktionselemente (Tastenfeld, Signallampe und Not-Aus-Taster) bereits auf der Ebene des mechatronischen Konzepts vollständig spezifiziert waren und somit in der Verfeinerung nicht betrachtet wurden, war die exemplarische HMI-Entwicklung zur Erprobung der Entwicklungsmethodik abgeschlossen. Der physikalische Aufbau des Systems wurde im Rahmen dieser Arbeit nicht adressiert, da in dieser Phase keine HMI-spezifischen Aktivitäten durchzuführen sind.

6.4 Technische Bewertung

Das im bisherigen Verlauf der Arbeit vorgestellte Konzept, dessen prototypische Umsetzung in entsprechenden Softwarelösungen sowie die Anwendung an einem Fallbeispiel bildeten die Grundlage für die technische Bewertung, die Gegenstand des vorliegenden Abschnitts ist. Dieser unterteilt sich in eine Bewertung anhand der in Kapitel 4 formulierten Anforderungen an die Methodik (Abschnitt 6.4.1) sowie eine darauf aufbauende Beschreibung der Nutzenpotenziale sowie vorhandener Aufwände und Einschränkungen (Abschnitt 6.4.2).

6.4.1 Bewertung anhand der formulierten Anforderungen

Im vierten Kapitel dieser Arbeit wurden insgesamt elf Anforderungen an die zu erarbeitende Entwicklungsmethodik formuliert. Der Erfüllungsgrad der einzelnen Anforderungen wird in den nachfolgenden Absätzen jeweils qualitativ bewertet.

Die HMI-Entwicklung ist über unterschiedliche Modellierungselemente in die gesamte Phase der Konzeption mechatronischer Systeme integriert. Auch danach wird über die Modelltransformation und die disziplinspezifische Verfeinerung das Vorgehen bis zu einem virtuellen Prototyp der Mensch-Maschine-Schnittstelle als Teil des mechatronischen Systems durchgängig beschrieben. Der abschließende physikalische Aufbau des Systems wird zwar nicht mehr betrachtet, allerdings können für die HMI-Entwicklung die bereits im ursprünglichen IMoMeSA-Ansatz definierten Aktivitäten, wie z. B. zum Systemtest, genutzt werden. Daher erfüllt der erarbeitete Ansatz die Anforderung nach einer *ganzheitlichen Prozessbetrachtung*.

Ferner wird durch die Spezifikation der einzelnen Modellierungselemente und die für deren Verwendung skizzierte Vorgehensweise die Anforderung nach *konkreten Aktivitäten und Ergebnissen* im Hinblick auf die Konzeptionsphase erfüllt. In der Verfeinerung können demgegenüber nur übergeordnete Aktivitäten und Ergebnisse definiert werden, was sich allerdings durch die Vielfalt an potenziell einsetzbaren Interaktionstechnologien begründen lässt, die einen entscheidenden Einfluss auf die konkreten Entwicklungsaufgaben haben. Allgemein wurde darauf geachtet, die einzelnen Aktivitäten mit einer agilen Herangehensweise umsetzen zu können. Dafür bietet sich speziell die Strukturierung der Benutzeraufgaben anhand der User Stories an, die dann in iterativen Entwicklungszyklen implementiert werden und damit einen inkrementellen Aufbau der Funktionalität des HMIs ermöglichen.

Eine dritte Anforderung bestand in der Ausarbeitung eines *schlanken Vorgehens*, das aufgrund der in der Regel begrenzten Entwicklungskapazitäten im Maschinen-

und Anlagenbau benötigt wird. Diese Anforderung wurde insbesondere bei der Qualitätssicherung berücksichtigt, in der die Simulation anstelle von umfangreichen Nutzerstudien und Feldtests zur Absicherung der Funktionalität und Gebrauchstauglichkeit fokussiert wird. Ferner wurde auch bei der Auswahl der Modellierungselemente bewusst auf ein schlankes Vorgehen geachtet, weswegen verschiedene Konzepte des Usability Engineerings (z. B. Personas) nicht in die Modellierungstechnik integriert wurden. Durch die Generierung von HMI-Programmen aus dem mechatronischen Konzept wird darüber hinaus der Aufwand zur Modellbildung relativiert, was eine schlanke Herangehensweise weiter unterstützt. Schließlich bietet der entwickelte Ansatz an unterschiedlichen Stellen Möglichkeiten zur Skalierung, um den Umfang an die Bedürfnisse des jeweiligen Unternehmens anzupassen. Insbesondere im Bereich der Modellierung von User Stories und Benutzerdialogen sowie deren Absicherung in der Simulation kann der Umfang prinzipiell frei gewählt werden.

Die Anforderung zur *Nutzung bestehender Ansätze* konnte schon durch die Auswahl des IMoMeSA-Ansatzes als Forschungsgrundlage geeignet berücksichtigt werden. Ferner konnten auch Konzepte aus der nutzerzentrierten Entwicklung von Mensch-Maschine-Schnittstellen genutzt und in den erarbeiteten Ansatz integriert werden. Hier ist insbesondere der Ansatz von MEIXNER (2010) zu nennen, aus dem u. a. die Konzepte zur Modellierung von Benutzeraufgaben und zur Spezifikation von Dialogen auf die vorliegende Problemstellung übertragen werden konnten.

Aus Sicht der Mechatronik wurde weiterhin die Anforderung zur Stärkung der *interdisziplinären Zusammenarbeit* formuliert. Um dieser Anforderung gerecht zu werden, wurde insbesondere bei der Modellierungstechnik auf die Nutzung bzw. Erweiterung bestehender Konzepte geachtet, um eine gemeinsame Sprache und einheitliche Schnittstellen zwischen den Disziplinen zu ermöglichen. Auch der Ansatz der Qualitätssicherung, bei dem mit Dialogtests und Nutzerevaluationen nicht nur das HMI, sondern das gesamte Systemverhalten analysiert werden kann, unterstützt die interdisziplinäre Zusammenarbeit. Schließlich liegt auch in der Verfeinerung eine klare Regelung vor, wie die Zusammenarbeit zwischen Disziplinen, insbesondere bei auftretenden Wechselwirkungen, zu erfolgen hat.

Die darauf aufbauende Anforderung nach der Verwendung eines *zentralen Systemmodells* wurde bereits durch die Auswahl des modellbasierten IMoMeSA-Ansatzes als Forschungsgrundlage umgesetzt. Das dort beschriebene Systemmodell bildet auch nach der Erweiterung des Ansatzes um HMI-Aspekte das Kernstück des gesamten Entwicklungsprozesses für mechatronische Systeme.

Zur Wiederverwendung bestehender Teillösungen bildet die *mechatronische Modularisierung* einen weiteren wichtigen Aspekt, der auch im Hinblick auf die HMI-Entwicklung berücksichtigt werden sollte. Generell bot der IMoMeSA-Ansatz mit der hierarchischen Strukturierung eines Systems in mechatronische und disziplinspezifische Komponenten bereits eine geeignete Basis. Durch die Einbettung der HMI-spezifischen Elemente in diese Struktur konnte das Modularisierungsprinzip aufrechterhalten werden, welches die Wiederverwendung erweiterter mechatronischer Komponenten mit HMI-spezifischen Bestandteilen ermöglicht.

Aus Sicht der HMI-Entwicklung bestand weiterhin die Anforderung nach einer *nutzerzentrierten Herangehensweise*. Diese wurde zunächst bei der Erweiterung der Modellierungstechnik berücksichtigt, in die verschiedene Elemente des Usability Engineerings (User Stories, Benutzerdialoge, etc.) integriert wurden, welche der eigentlichen Konzeption und Umsetzung der Mensch-Maschine-Schnittstelle vorangehen und explizit auf eine hohe Gebrauchstauglichkeit des HMIs abzielen. Ferner wird über den in der Qualitätssicherung adressierten Ansatz der Nutzerevaluation sowie deren Fortsetzung in der Verfeinerung eine kontinuierliche Einbeziehung der späteren Nutzer unterstützt.

Eng verknüpft mit der zuletzt beschriebenen Anforderung ist die Notwendigkeit zum *Test von Mensch-Maschine-Schnittstellen*. Um die Relevanz hervorzuheben, wurde die Qualitätssicherung als eine der vier zentralen Aktivitäten für die HMI-Entwicklung definiert, welche sowohl die Konzeptions- als auch die Verfeinerungsphase kontinuierlich begleitet. In diesem Zusammenhang wurden auch zwei eigene Simulationstechniken auf Basis des IMoMeSA-Ansatzes entwickelt, um sowohl die Funktionalität als auch die Gebrauchstauglichkeit eines HMIs absichern zu können.

Eine weitere Anforderung bestand in der Berücksichtigung von *HMI-Spezifika im Maschinen- und Anlagenbau*. Um dieser gerecht zu werden, wurde bspw. die Objekt-Repräsentation von Interaktionselementen eingeführt, um auch die Position eines HMIs an einer Maschine bzw. Anlage zu spezifizieren. Weiterhin wurde bei Benutzerdialogen auch der insbesondere im Maschinen- und Anlagenbau auftretende Fall berücksichtigt, dass mehrere Benutzer an der Durchführung einer Aufgabe beteiligt sind. Darüber hinaus wurden bei der Auswahl der Bedien- und Anzeigeelemente innerhalb der Screen-Repräsentationen typische Aufgaben für den Maschinen- und Anlagenbau (z. B. Parametereingaben) berücksichtigt. Schließlich unterstützt der Ansatz auch die Bereitstellung sich wiederholender Muster über das Template-Konzept des ursprünglichen IMoMeSA-Ansatzes. Allerdings wurde dieses Konzept im Rahmen der Arbeit nicht explizit weiterentwickelt.

Die letzte Anforderung an die Entwicklungsmethodik adressierte die Berücksichtigung *innovativer Interaktionstechnologien*. Diese Anforderung wurde speziell bei der Technologieauswahl beachtet. Die dort erarbeiteten Klassen von Interaktionstechnologien mit jeweiligen Klassifikationskriterien erstrecken sich über das gesamte Spektrum der Interaktionstechnologien und lassen sich auch um zukünftige Entwicklungen in diesem Bereich aufwandsarm erweitern. Ferner ermöglichen fast alle Elemente der Modellierungstechnik eine technologieunabhängige Spezifikation einer Mensch-Maschine-Schnittstelle. Eine Ausnahme bilden hier die Interaktionselemente, für die im Rahmen dieser Arbeit nur eine vereinfachte Nachbildung von graphischen HMIs konzipiert und umgesetzt wurde. Durch den modularen Aufbau der Modellierungstechnik können aber zukünftig auch Interaktionselemente von bspw. Sprachsteuerungen in die Modellierungstechnik integriert werden. Auch in der Verfeinerung wurde grundsätzlich ein technologieunabhängiges Konzept für die Modelltransformation und die zugehörige Vorgehensweise gewählt, das allerdings im Bereich der Darstellung nur eine Konkretisierung für graphische HMIs anbietet.

6.4.2 Nutzenpotenziale, Aufwände und Einschränkungen

Die im letzten Abschnitt vorgenommene Bewertung der Anforderungen an die Entwicklungsmethodik gibt nur einen begrenzten Einblick hinsichtlich der Chancen und Herausforderungen für eine Nutzung im industriellen Umfeld. Daher werden nachfolgend bewusst die Nutzenpotenziale der Entwicklungsmethodik, aber auch zu berücksichtigende Aufwände und Einschränkungen herausgestellt.

Allgemein liegt ein Nutzenpotenzial der modellbasierten Entwicklung mechatronischer Systeme in der verbesserten Abstimmung aller beteiligten Disziplinen durch den frühzeitigen Aufbau eines gemeinsamen Systemmodells. Daher ist es sinnvoll, den IMoMeSA-Ansatz zunächst gesamtheitlich zu betrachten, dessen Vorteile allerdings bereits in HACKENBERG ET AL. (2015, S. 72 ff.) hinreichend beschrieben sind. Durch die vorgenommene Integration der HMI-Entwicklung in diesen Ansatz lassen sich diese Nutzenpotenziale auf die vorliegende Arbeit übertragen. Allerdings gehen auch die übergeordneten Aufwände zur Nutzung der modellbasierten Entwicklung einher, die sich von der Umstellung von Workflows bis zum Change-Management belaufen (vgl. RICHTER ET AL. 2016, S. 94-96). Um den Ansatz daher verwenden zu können, bedarf es einer Einführungsstrategie, für die auf vorhandene Konzepte, wie bspw. ALT (2012, S. 161 ff.), zurückgegriffen werden kann.

Nach entsprechender Etablierung des IMoMeSA-Ansatzes in einer industriellen Umgebung können die von HACKENBERG ET AL. (2015) skizzierten Potenziale

erschlossen werden, die aus HMI-Sicht nun um folgende Aspekte ergänzt werden können. Hier ist die frühzeitige Einbeziehung der Mensch-Maschine-Schnittstelle in die Entwicklung zu nennen, durch die eine höhere Qualität eines HMIs im Hinblick auf seine Funktionalität und Gebrauchstauglichkeit erreicht werden kann. Ferner können durch den interdisziplinären Austausch und die Möglichkeit zur frühzeitigen Abstimmung mit Kunden spätere Korrekturen auf ein Minimum reduziert werden. Auch die Aufwände zur Modellbildung werden durch Einsparungen in späteren Entwicklungsphasen gerechtfertigt, was insbesondere durch die automatische Generierung von HMI-Programmen ermöglicht wird. Allerdings gelten die skizzierten Potenziale vollumfänglich nur für graphische Mensch-Maschine-Schnittstellen, auf welche der Ansatz im Hinblick auf die Interaktionselemente beschränkt ist.

Die eben beschriebenen Potenziale ergeben sich aus einer ganzheitlichen Nutzung des modellbasierten Entwicklungsansatzes, es besteht allerdings durch den modularen Aufbau der Entwicklungsmethodik auch die Möglichkeit, einzelne Bausteine losgelöst vom Rest des Systems einzusetzen, um eine Unterstützung spezifischer Entwicklungsaufgaben zu ermöglichen. Dies trifft vor allem auf die erarbeitete Einflussmatrix zu, welche die Auswahl einer geeigneten Interaktionstechnologie für einen konkreten Anwendungsfall unterstützt. Die Aufwände zur Nutzung dieses Bausteins liegen vor allem in der Befragung aller relevanten Stakeholder sowie der Gewinnung valider Daten über einen Anwendungsfall, ansonsten kann die Einflussmatrix ohne zusätzliche Aufwände in einem Entwicklungsprozess genutzt werden. Die weiteren Bausteine sind hingegen eng miteinander verknüpft und speziell die Qualitätssicherung sowie die Modelltransformation können nur bei einer ganzheitlichen Verwendung des IMoMeSA-Ansatzes zielbringend verwendet werden. Aus der Modellierungstechnik selbst können jedoch einzelne Elemente losgelöst vom gesamten Ansatz genutzt werden, um eine frühzeitige Fokussierung der Nutzer zur Steigerung der Gebrauchstauglichkeit zu ermöglichen. So bietet sich bspw. eine Modellierung von Benutzeraufgaben über User Stories, HMI-Ports und Benutzerdialoge auch an, um diese in einer Systemspezifikation zu dokumentieren und mit dem Kunden zu diskutieren. In der Regel kann die Modellierung in diesem Fall sogar vereinfacht werden, da der für die Simulation notwendige Formalismus nicht beachtet werden muss. Weiterhin sind auch die Interaktionselemente, insbesondere die Screen-Repräsentationen, gekapselt nutzbar, um verschiedenen Stakeholdern frühzeitig einen Überblick des geplanten Aufbaus einer Mensch-Maschine-Schnittstelle zu ermöglichen. Dies gestattet aufwandsarme Anpassungen auf Modellebene und trägt somit zur Vermeidung umfangreicher Änderungsarbeiten in der späteren Implementierung bei.

Durch die kontinuierliche Evaluation des Ansatzes mit Fachexperten unterschiedlicher Disziplinen sowie die prototypische Umsetzung und Erprobung in einem industrienahen Anwendungsszenario ist die generelle Nutzbarkeit des Ansatzes in der industriellen Praxis sichergestellt. Dazu wurde bei der Wahl des Fallbeispiels auf ein repräsentatives Spektrum an Benutzeraufgaben geachtet, um eine Übertragbarkeit auf vielfältige Anwendungsfälle in der Industrie zu ermöglichen. Allerdings ist der betrachtete Anwendungsfall hinsichtlich seines Umfangs limitiert, weswegen eine weitere Evaluation des Ansatzes im Kontext einer gesamten Maschinen- bzw. Anlagenentwicklung für den Transfer in die Industrie zielführend ist. Ferner ist als Einschränkung hervorzuheben, dass das Fallbeispiel zum Zeitpunkt der Erprobung bereits fertiggestellt war, weswegen interdisziplinäre Wechselwirkungen in der Konzeption und Verfeinerung nur nachgestellt werden konnten. In einer realen Anwendung ist zu bestätigen, dass der erarbeitete Ansatz den interdisziplinären Austausch so zielführend unterstützt, wie dies die bisherigen Erkenntnisse nahelegen. Eine weitere Einschränkung betrifft die erarbeitete Einflussmatrix. Deren Bestandteile (Auswahlfaktoren, Technologieklassen, Bewertungsschema, etc.) wurden mit Hilfe von Literaturrecherchen und Expertengesprächen aufgebaut, allerdings sind für einen industriellen Einsatz weiterführende Analysen bspw. hinsichtlich der identifizierten Eignungsbedingungen und Eignungsgrade oder den gewählten Abstufungen bei den Ausprägungen der Auswahlfaktoren durchzuführen. Die für das Fallbeispiel generierten Ergebnisse aus der Einflussmatrix wurden von den Experten zwar als richtig eingestuft, jedoch ist auch hier eine Anwendung auf weitere Fallbeispiele zielführend, um den praktischen Nutzen für unterschiedliche Problemstellungen zu bestätigen.

6.5 Wirtschaftliche Bewertung

Aufgrund der Heterogenität in den Entwicklungsprozessen des Maschinen- und Anlagenbaus kann bei der wirtschaftlichen Bewertung des erarbeiteten Ansatzes nicht von einem allgemeingültigen Fall ausgegangen werden. Vor diesem Hintergrund wird nachfolgend eine beispielhafte Wirtschaftlichkeitsbewertung anhand der *Kapitalwertmethode* (vgl. BECKER 2015, S. 179 ff.) für ein fiktives Unternehmen durchgeführt, das eine Einführung der erarbeiteten Methodik zur modellbasierten HMI-Entwicklung in Erwägung zieht. Bei dieser Bewertung sind die jährlichen Ausgaben zu berücksichtigen, die durch die Einführung des Ansatzes entstehen, und den jährlichen Einnahmen gegenüberzustellen, die ein Unternehmen durch den Einsatz der Methodik generieren kann.

Im Hinblick auf die *Ausgaben* ist zunächst der Umstand abzubilden, dass sich der erarbeitete Ansatz in den gesamten IMoMeSA-Ansatz einbettet. Daher ist es nicht zielführend, alle entstehenden Ausgaben vollständig in die wirtschaftliche Bewertung einzubeziehen, da auch alle weiteren Disziplinen neben der HMI-Entwicklung von dem Ansatz profitieren. Andererseits soll auch nicht von einer bereits erfolgten Einführung des IMoMeSA-Ansatzes ausgegangen werden, da in diesem Fall nahezu keine Ausgaben für die Erweiterung um HMI-Aspekte entstehen würden. Vielmehr soll im Rahmen der nachfolgenden Betrachtung ein fester prozentualer Anteil der Ausgaben auf die HMI-Entwicklung umgelegt werden, der in die Kapitalwertmethode einfließt. Zur Ermittlung dieses Faktors wurde der Sollanteil der HMI-Entwicklung an der gesamten Entwicklungskapazität mit einem Faktor von 0,15 abgeschätzt. Dieser ist mit den Investitionskosten und den laufenden Kosten zu multiplizieren, um die auf die HMI-Entwicklung umlegbaren Kosten zu ermitteln. Zur Berechnung dieser Kosten wurden die auf Literaturrecherchen und Expertengesprächen basierenden Werte aus Tabelle 6-1 verwendet, durch welche sich die Investitionskosten zu 9.000 € und die laufenden Kosten zu jährlich 1.350 € ergeben.

Tabelle 6-1: Angenommene Ausgaben für die Wirtschaftlichkeitsbewertung

Investitionskosten (in €)	
Investitionen Grundaufbau der Software	30.000
Unternehmensspezifische Anpassung der Software	15.000
Organisatorische Aufwände zur Integration (u. a. Schulungen)	15.000
Summe Invest	60.000
Summe Invest (HMI)	9.000

Jährliche Kosten (in € / a)	
Service- und Wartungsvertrag	6.000
Konfigurations- und Anpassungsaufwände	3.000
Summe jährliche Kosten	9.000
Summe jährliche Kosten (HMI)	1.350

Den in Tabelle 6-1 dargelegten Aufwänden werden in einem zweiten Schritt die *Einnahmen* gegenübergestellt, welche das fiktive Unternehmen durch den Einsatz der Entwicklungsmethodik jährlich generieren kann. Auch hier sind nur diejenigen Einnahmen zu berücksichtigen, die allein durch die Mensch-Maschine-Schnittstelle generiert werden, obgleich durch die Verwendung des IMoMeSA-Ansatzes auch weiterführende wirtschaftliche Potenziale geschaffen werden. Dabei sind einerseits Einnahmen durch die Vermeidung von Korrekturen an einem HMI während der Inbetriebnahme oder der Betriebsphase zu nennen, was durch die frühe Absicherung der Funktionalität und Gebrauchstauglichkeit erreicht wird. Andererseits lässt sich

durch den interdisziplinären Austausch auch allgemein der Entwicklungsaufwand reduzieren. In beiden Fällen wird angenommen, dass die frei werdenden Kapazitäten in zusätzliche Entwicklungsprojekte eingebracht werden und hierdurch entsprechende Einnahmen generieren. Hier ist zu beachten, dass diese Einsparungen anfangs noch gering sind, aber im Lauf der Zeit ansteigen, was durch eine entsprechende Lernkurve der Mitarbeiter zu begründen ist. Im betrachteten Fall ist sogar in den ersten beiden Jahren mit einem Anstieg des Entwicklungsaufwands zu rechnen, der auf die bislang nicht vorhandenen Kapazitäten hinsichtlich der HMI-Gebrauchstauglichkeit zurückführen ist (vgl. negative Zeitersparnis in Tabelle 6-2). Zuletzt entstehen Einnahmen durch eine höhere Qualität des HMIs, durch die höhere Preise und größere Stückzahlen und somit ein zusätzlicher Umsatz generiert werden kann. In Tabelle 6-2 sind die innerhalb dieser Kategorien ermittelten Werte sowie die resultierenden Einnahmen für das fiktive Unternehmen für einen Abschreibungszeitraum von fünf Jahren aufgeführt. Die ermittelten Werte basieren erneut auf Literaturrecherchen und Expertengesprächen.

Tabelle 6-2: Angenommene Einnahmen für die Wirtschaftlichkeitsbewertung

Jährliche Einnahmen				
Jahr	Zeitersparnis in der HMI-Entwicklung* (in h)	Einnahmen durch Zeitersparnis** (in €)	Einnahmen durch Umsatzsteigerungen (in €)	Jährliche Einnahmen (in €)
1	-150	-10.500	0	-10.500
2	-75	-5.250	5.000	-250
3	50	3.500	7.500	11.000
4	100	7.000	10.000	17.000
5	150	10.500	12.500	23.000

*Berücksichtigt werden der generelle Entwicklungsaufwand und der Aufwand für Korrekturen.

**Für die Berechnung wurde der Stundensatz eines HMI-Entwicklers mit 70 € abgeschätzt.

Mit den Einnahmen und Ausgaben kann der Kapitalwert für eine Investition in die HMI-Entwicklungsmethodik für das Fallbeispiel bestimmt werden. Hierbei sind der Abschreibungszeitraum von fünf Jahren und der kalkulatorische Zinssatz von fünf Prozent zu berücksichtigen. Aus dem Zinssatz kann der jährliche Abzinsfaktor ermittelt werden, der mit der Differenz der Ausgaben und Einnahmen multipliziert wird, um den jährlichen Barwert der Überschüsse zu berechnen. Der Kapitalwert eines Jahres ergibt sich durch Summenbildung aller Barwerte bis zu diesem Jahr. Als gesamten Kapitalwert ergibt sich für das fiktive Unternehmen ein Wert von 16.438 €, der für eine Investition in den erarbeiteten Ansatz spricht. Tabelle 6-3 zeigt zusammenfassend die durchgeführte Berechnung des Kapitalwerts.

Tabelle 6-3: Berechnung des Kapitalwerts für das fiktive Unternehmen

Kapitalwertberechnung						
Jahr	Einnahmen (in €)	Ausgaben (in €)	Überschuss (in €)	Abzins- faktor	Barwert der Über- schüsse (in €)	Kapitalwert (in €)
0	-	9.000	-9.000	1,00000	-9.000	-9.000
1	-10.500	1.350	-11.850	0,95238	-11.286	-20.286
2	-250	1.350	-1.600	0,90703	-1.451	-21.737
3	11.000	1.350	9.650	0,86384	8.336	-13.401
4	17.000	1.350	15.650	0,82270	12.875	-526
5	23.000	1.350	21.650	0,78353	16.963	16.438
Summe	40.250	15.750	24.500			16.438

Zur Verallgemeinerung der Erkenntnisse aus der exemplarischen Anwendung der Kapitalwertmethode wird eine *Grenzwertbetrachtung* (vgl. VOGL 2009) durchgeführt, um einen generellen Wirtschaftskorridor für den Einsatz der Entwicklungsmethodik anzugeben. Dabei werden die ermittelten Ausgaben des Fallbeispiels (vgl. Tabelle 6-1) als konstant angenommen, während die jährliche Zeitersparnis in der Entwicklung sowie die Einnahmen durch Umsatzsteigerungen als Variablen definiert werden. Zur Vereinfachung werden sowohl für die Zeitersparnis als auch die Umsatzsteigerungen konstante bzw. gemittelte Werte über den Abschreibungszeitraum von fünf Jahren verwendet. Auf Basis dieser Annahme sowie unter Nutzung des Abzinsfaktors von fünf Prozent kann eine Isolinie für Wertpaarungen der Zeitersparnis und der Einnahmen ermittelt werden, bei der sich ein Kapitalwert von Null ergibt (vgl. Abbildung 6-12). Wertepaarungen über dieser Linie besitzen einen positiven Kapitalwert und stellen somit vorteilhafte Investitionen dar, wohingegen bei Wertpaarungen unter dieser Linie eine modellbasierte HMI-Entwicklung keinen finanziellen Mehrwert darstellt. Anhand der Grenzkurve lässt sich auch das zuvor betrachtete Fallbeispiel mit positiven Kapitalwert einordnen (x_0), wobei die in Tabelle 6-2 angegebenen Werte zur Zeitersparnis und den Einnahmen entsprechend gemittelt wurden. Es ist zu erkennen, dass bei einer jährlichen Zeitersparnis von 15 Stunden bereits durchschnittliche Einnahmen von ca. 2.400 € ausreichen würden, um einen neutralen Kapitalwert zu erzielen (x_1). Auf der anderen Seite wären sogar durchschnittliche Mehraufwände von ca. 50 Stunden pro Jahr für einen neutralen Kapitalwert zu tolerieren, wenn sich wie im Fallbeispiel die durchschnittlichen Einnahmen zu 7.000 € ergeben (x_2). Allgemein wird deutlich, dass ab einem Grenzwert von 3.430 € bei den jährlichen Einnahmen auch zeitliche Mehraufwände in der Entwicklung denkbar sind, um als Unternehmen insgesamt vom Einsatz einer modellbasierten HMI-Entwicklung zu profitieren.

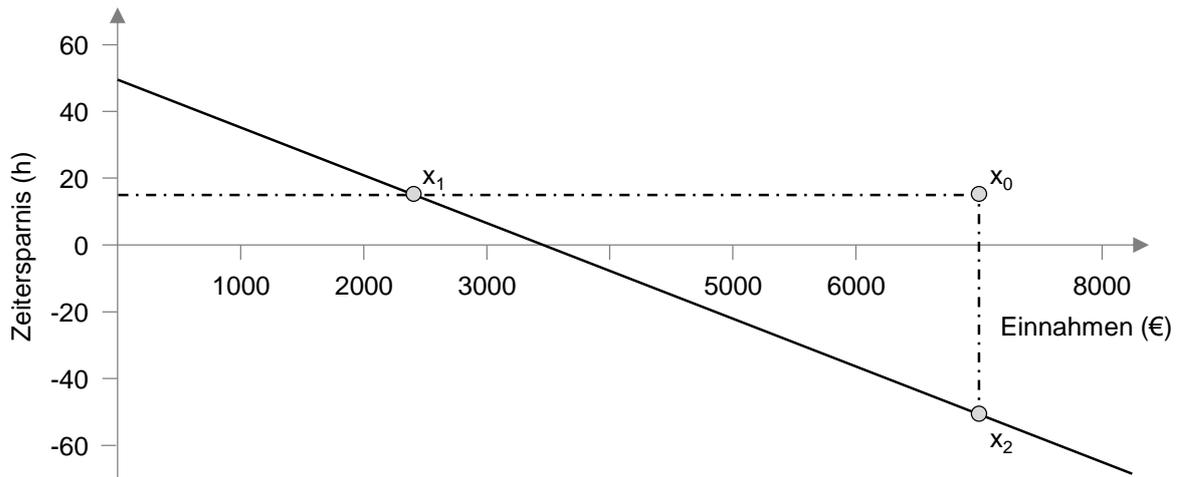


Abbildung 6-12: Wirtschaftlichkeitsgrenze der Entwicklungsmethodik

6.6 Zusammenfassung

Dieses Kapitel befasste sich mit der Anwendung und Bewertung der erarbeiteten Methodik. Nach einer Beschreibung der Vorgehensweise wurden die Softwarewerkzeuge vorgestellt, in denen die technischen Lösungsbausteine prototypisch umgesetzt wurden. Die erweiterte Modellierungstechnik, die Modelltransformation und die Simulationstechniken wurden in das Eclipse-basierte Werkzeug „MACON“ integriert, während die Einflussmatrix in Microsoft® Excel umgesetzt wurde.

Auf dieser Basis wurde das Fallbeispiel eingeführt, anhand dessen die praktische Anwendbarkeit des Ansatzes demonstriert wurde. Hierfür wurde ein Modell einer Einrichtung zum Stempeln von Werkstücken gewählt, das ein breites Spektrum typischer Benutzeraufgaben im Maschinen- und Anlagenbau abbildet. Für die Entwicklung des HMIs dieser Einrichtung wurde zunächst ein entsprechendes Modell erarbeitet. Auf dieser Basis konnten geeignete Interaktionstechnologien über die Einflussmatrix ausgewählt werden. Speziell für das zentrale Bedienpanel wurde des Weiteren eine WPF-Applikation in der Verfeinerung implementiert, wobei auch das erarbeitete Konzept der Modelltransformation zum Einsatz kam. Ferner wurden die simulationsbasierten Qualitätssicherungsmaßnahmen genutzt, um sowohl die Funktionalität als auch die Gebrauchstauglichkeit des HMIs abzusichern.

Im zweiten Teil des Kapitels erfolgte eine technische und wirtschaftliche Bewertung. Aus technischer Sicht wurde überprüft, ob die formulierten Anforderungen erfüllt wurden. Darüber hinaus wurden vorhandene Nutzenpotenziale, aber auch resultierende Aufwände und vorhandene Einschränkungen herausgearbeitet. Die Wirtschaftlichkeit des Ansatzes wurde anhand der Kapitalwertmethode abgeschätzt.

7 Schlussbetrachtung

Vor dem Hintergrund der aktuell steigenden Bedeutung von Mensch-Maschine-Schnittstellen im Maschinen- und Anlagenbau ist es notwendig, diese als integralen Bestandteil eines Entwicklungsprozesses zu betrachten und nicht mehr wie bislang am Ende der Prozesskette zu implementieren. Hierdurch wird es möglich, HMIs mit innovativen Interaktionstechnologien zu realisieren sowie generell eine hohe Gebrauchstauglichkeit zu gewährleisten. Um dies zu ermöglichen, bieten aktuelle Entwicklungsansätze in der Mechatronik, u. a. die modellbasierte Entwicklung, eine geeignete Basis, da sie auf die frühzeitige Beteiligung aller Disziplinen und einen kontinuierlichen Austausch aller Gewerke fokussiert sind. Allerdings spielen in diesen Ansätzen Fragestellungen der Mensch-Maschine-Schnittstelle bislang keine oder nur eine untergeordnete Rolle. Daher ist es notwendig, diese Ansätze mit Konzepten des Usability Engineerings bzw. der nutzerzentrierten HMI-Entwicklung anzureichern und zu einem ganzheitlichen Ansatz zu verbinden.

Im Rahmen dieser Arbeit wurde daher in einem ersten Schritt ein konkreter modellbasierter Entwicklungsansatz ausgewählt, der hinsichtlich der Anforderungen an die HMI-Erweiterung eine geeignete Forschungsgrundlage bildet. Für den gewählten IMoMeSA-Ansatz von HACKENBERG ET AL. (2015) wurde zunächst analysiert, wie die HMI-Entwicklung in den bestehenden Aufbau des Modells integriert werden kann, woraus vier zentrale Aktivitäten aus Sicht der Mensch-Maschine-Schnittstelle resultierten. Die erste Aktivität adressiert dabei die Modellierung des HMIs als Teil eines interdisziplinären Systemmodells, wofür die bestehende Modellierungstechnik um HMI-spezifische Elemente erweitert wurde. Eine zweite Aktivität bildet die Auswahl einer geeigneten Interaktionstechnologie, für die eine Einflussmatrix zur Gegenüberstellung von relevanten Auswahlfaktoren und Interaktionstechnologien erarbeitet wurde. Durch die Einordnung eines Anwendungsfalls anhand der Auswahlfaktoren können konkrete Kriterien abgeleitet werden, die Interaktionstechnologien aufweisen sollten und somit bei der Technologieauswahl einbezogen werden können. Auf der Basis der HMI-Modellierung und der Auswahl von Interaktionstechnologien erfolgt in einer dritten Aktivität die Implementierung der Mensch-Maschine-Schnittstelle. Dabei kommt eine Modelltransformation zum Einsatz, um die auf Modellebene abgebildeten Inhalte automatisiert in ein Programm der Mensch-Maschine-Schnittstelle zu überführen. Parallel zum gesamten Entwicklungsprozess erfolgt die vierte Aktivität der Qualitätssicherung, um sowohl die Funktionalität als auch die Gebrauchstauglichkeit eines HMIs abzusichern. Im Rahmen dieser Arbeit wird dazu der Einsatz von Simulation vorgeschlagen, für den zwei spezielle Simulationstechniken für die HMI-Absicherung entwickelt wurden.

Um die Anwendbarkeit der erarbeiteten Methodik im industriellen Umfeld zu demonstrieren, wurden in einem ersten Schritt die technischen Lösungsbausteine in entsprechenden Softwarewerkzeugen prototypisch umgesetzt. Dies bildete die Basis für die Anwendung der Entwicklungsmethodik an einem Fallbeispiel, wofür im Rahmen dieser Arbeit ein Modell einer Einrichtung zum Stempeln von Werkstücken zum Einsatz kam. Am Beispiel der HMI-Entwicklung für diese Einrichtung konnten die einzelnen Aktivitäten erfolgreich angewandt werden. Die hierbei gewonnenen Erkenntnisse flossen schließlich in eine technische und wirtschaftliche Bewertung des Ansatzes ein.

Durch die technische Bewertung konnten bereits wichtige Maßnahmen identifiziert werden, die für einen operativen Einsatz der Entwicklungsmethodik in einem industriellen Kontext notwendig sind. Darüber hinaus ergeben sich auch aus wissenschaftlicher Sicht verschiedene Anknüpfungspunkte zur Erweiterung des Ansatzes. Hierbei ist zunächst die aktuell noch zum Teil vorhandene Beschränkung des Ansatzes auf graphische Mensch-Maschine-Schnittstellen zu nennen. Auf der Ebene des Systemmodells sind Erweiterungen im Bereich der Interaktionselemente möglich, um auch Sprach- oder Gestensteuerungen auf Modellebene abbilden und deren Funktionalität und Gebrauchstauglichkeit frühzeitig in der Simulation absichern zu können. Diese Modellierung ermöglicht wiederum konzeptionelle Erweiterungen im Bereich der Modelltransformation sowie der HMI-Verfeinerung im Allgemeinen. Einen weiteren Anknüpfungspunkt bietet schließlich das Themenfeld der mechatronischen Modularisierung. Der Aufbau der Modellierungstechnik über mechatronische Komponenten gestattet zwar generell die Wiederverwendung über Entwicklungsprojekte hinweg, allerdings wird dieser Aspekt nicht so explizit wie z. B. in den Konzepten von LÜDER ET AL. (2010) oder REINHART ET AL. (2009) betrachtet. Hierbei ist zu analysieren, welche Elemente einer Mensch-Maschine-Schnittstelle in einen mechatronischen Baukasten integriert werden können.

Ein weiteres Potenzial für zukünftige Erweiterungen des Ansatzes bietet die ganzheitliche Ausrichtung eines mechatronischen Entwicklungsprozesses an den Nutzern des Systems, wie dies bspw. von BERG ET AL. (2017) beschrieben wird. Im erarbeiteten Ansatz werden die Konzepte des Usability Engineerings ausschließlich zur Entwicklung der Mensch-Maschine-Schnittstelle genutzt. Hier ist zu prüfen, inwiefern diese Konzepte zukünftig den allgemeinen Rahmen eines Entwicklungsprozesses bilden können. In diesem Zusammenhang ist auch zu analysieren, welche Synergien zum Themenfeld der kundenorientierten Produktentwicklung geschaffen werden können, in dem zahlreiche Ansätze erarbeitet wurden, um die Bedürfnisse des Kunden durchgängig in der Entwicklung berücksichtigen zu können.

Abschließend sind auch die Auswirkungen einer weiterführenden Digitalisierung im Maschinen- und Anlagenbau als ein wichtiger Treiber für mögliche Erweiterungen der Entwicklungsmethodik zu nennen. Hierbei ist insbesondere die Vernetzung von intelligenten Maschinen und Anlagen zu nennen, die dazu führen, dass Benutzer in Zukunft ein verändertes Aufgabenspektrum zu bewältigen haben. Durch autonome Entscheidungen im Produktionsumfeld (SCHEER 2013, S. 6) und wandlungsfähige Systeme (STEEGMÜLLER & ZÜRN 2014, S. 103-104) werden Benutzer mit sich kontinuierlich verändernden Aufgaben konfrontiert. Hierfür können adaptive Mensch-Maschine-Schnittstellen, die sich an aktuelle Rahmenbedingungen anpassen können (vgl. PEISSNER 2015, S. 17), einen geeigneten Lösungsansatz bilden, für deren Entwicklung der vorliegende Ansatz zu erweitern ist. Darüber hinaus können auch die Anbindung von Maschinen und Anlagen an das Internet (KAUFMANN 2015, S. 2) oder die weiterhin zunehmende Automatisierung (HIRSCH-KREINSEN 2014, S. 5-8) einen Einfluss auf die zukünftige Interaktion von Benutzern mit mechatronischen Systemen haben. Hierbei ist allerdings zu erwähnen, dass Fachexperten auch in Zeiten der Digitalisierung von einer wichtigen Rolle des Menschen in der Produktion ausgehen, für die auch zukünftig geeignete Schnittstellen zu den Maschinen und Anlagen entwickelt werden müssen (SPATH ET AL. 2013, S. 46; REINHART ET AL. 2017, S. 51; GORECKY ET AL. 2017, S. 217-218; VOGEL-HEUSER 2017, S. 45 f.).

8 Literaturverzeichnis

ABELE & REINHART 2011

Abele, E.; Reinhart, G.: Zukunft der Produktion. Herausforderungen, Forschungsfelder, Chancen. München: Hanser 2011. ISBN: 3446425950.

ABRAS ET AL. 2004

Abras, C.; Maloney-Krichmar, D.; Preece, J.: User-centered design. In: Bainbridge, W.: Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications, S. 445-456. ISBN: 0974309125.

ACATECH 2016

acatech (Hrsg.): Innovationspotenziale der Mensch-Maschine-Interaktion (acatech IMPULS). München: Herbert Utz 2016. ISBN: 9783831644971.

AHMED & ASHRAF 2007

Ahmed, S.; Ashraf, G.: Model-based user interface engineering with design patterns. In: The Journal of Systems and Software 80 (2007), S. 1408-1422.

ALT 2012

Alt, O.: Modellbasierte Systementwicklung mit SysML. München: Hanser 2012. ISBN: 9783446430662.

ANACKER & GAUSEMEIER 2014

Anacker, H.; Gausemeier, J.: Einführung. In: Gausemeier, J. et al. (Hrsg.): Semantische Technologien im Entwurf mechatronischer Systeme. München: Hanser 2014, S. 9-24. ISBN: 9783446436305.

ANDERL ET AL. 2012

Anderl, R.; Nattermann, R.; Rollmann, T.: Das W-Modell - Systems Engineering in der Entwicklung aktiver Systeme. (2012). <<http://www.plmportal.org/de/forschung-detail/das-w-modell-systems-engineering-in-der-entwicklung-aktiver-systeme.html>> - 03.09.2017.

ARNOWITZ ET AL. 2007

Arnowitz, J.; Arent, M.; Berger, N.: Effective Prototyping for Software Makers. San Francisco: Morgan Kaufmann 2007. ISBN: 9780120885688.

AUER 2012

Auer, J.: Gestensteuerung - Abschied von den Tasten. System Integrator, Fastems Kundenzeitschrift (2012) 1, S. 7.

AUMAYR 2016

Aumayr, K.: Erfolgreiches Produktmanagement. Tool-Box für das professionelle Produktmanagement und Produktmarketing. 4. Auflage. Wiesbaden: Springer 2016. ISBN: 9783658106683.

BACKHAUS ET AL. 2016

Backhaus, J.; Scheib, J.; Pieloth, R.; Korajda, B.; Richter, C.; Reinhart, G.: Automatische Programmierung von Verarbeitungsmaschinen. In: atp – Automatisierungstechnische Praxis 58 (2016) 11, S. 36-45.

BAUMANN & LANZ 2013

Baumann, K.; Lanz, H.: Mensch-Maschine-Schnittstellen elektronischer Geräte. Leitfaden für Design und Schaltungstechnik. Berlin, Heidelberg: Springer 2013. ISBN: 9783642637384.

BAXTER ET AL. 2015

Baxter, K.; Courage, C.; Caine, K.: Understanding your users. A practical guide to user research methods. Waltham: Morgan Kaufmann 2015. ISBN: 9780128002322.

BÄUMER ET AL. 1996

Bäumer, D.; Bischofberger, W. R.; Lichter, H.; Züllighoven, H.: User Interface Prototyping - concepts, tools, and experience. In: Proceedings of the 18th international conference on Software Engineering (1996), S. 532-541.

BECERRIL ET AL. 2016

Becerril, L.; Kattner, N.; Schweigert, S.: Methoden der Funktionsmodellierung. In: Lindemann, U. (Hrsg.): Handbuch Produktentwicklung. München: Hanser 2016, S. 697-712. ISBN: 9783446445185.

BECK ET AL. 2001

Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D.: Manifest für agile Softwareentwicklung. (2001). <<http://agilemanifesto.org/iso/de/manifesto.html>> - 03.09.2017.

BECKER 2004

Becker, H. P.: Investieren und Finanzieren. In: Irgel, L. (Hrsg.): Gablers Wirtschaftswissen für Praktiker. Zuverlässige Orientierung in allen kaufmännischen Fragen. 5. Auflage. Wiesbaden: Gabler 2004, S. 179-180. ISBN: 9783322903167.

BECKER ET AL. 2011

Becker, R.; Ruckert, C.; Tappert, E.: Methodische Entwicklung von Mensch-Maschine-Schnittstellen unter Berücksichtigung nutzerzentrierter und modellbasierter Ansätze. In: 41. Fachtagung der Gesellschaft für Informatik (2011).

BENDER 2005

Bender, K.: Embedded Systems - qualitätsorientierte Entwicklung. Berlin, Heidelberg: Springer 2005. ISBN: 9783540229957.

BENGLER 2011

Bengler, K.: Interdisziplinäres Arbeiten im Bereich der Produktergonomie. In: Zeitschrift für Arbeitswissenschaft. 65 (2011) 1, S. 77-79.

BENGLER ET AL. 2012

Bengler, K.; Zimmermann, M.; Bortot, D.; Kienle, M.; Damböck, D.: Interaction Principles for Cooperative Human-Machine Systems. In: it - Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik 54 (2012) 4, S. 157-164.

BENGLER 2015

Bengler, K.: Messmethoden. In: Bubb, H. et al. (Hrsg.): Automobilergonomie. Wiesbaden: Springer Vieweg 2015, S. 617-662. ISBN: 9783834822970.

BENGLER ET AL. 2015A

Bengler, K.; Pfromm, M.; Bruder, R.: Bedienelemente. In: Winner, H. et al. (Hrsg.): Handbuch Fahrerassistenzsysteme. Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort. 3., überarbeitete und ergänzte Auflage. Wiesbaden: Springer Vieweg 2015, S. 647-657. ISBN: 9783658057336.

BENGLER ET AL. 2015B

Bengler, K.; Götze, M.; Pfannmüller, L.; Zaindl, A.: To See or not to See - Innovative Display Technologies as Enablers for Ergonomic Cockpit Concepts. In: Proceedings of the electronic displays Conference (2015).

BERG ET AL. 2017

Berg, J.; Richter, C.; Spiegelberger, B.; Reinhart, G.: Nutzerzentrierte Entwicklung mechatronischer Systeme. In: atp – Automatisierungstechnische Praxis 59 (2017) 1, S. 40-49.

BEYER 2010

Beyer, H.: User-centered agile methods. In: Synthesis lecture on human-centered informatics 3 (2010) 1, S. 1-71.

BLOCHWITZ ET AL. 2012

Blochwitz, T.; Otter, M.; Akesson, J.; Arnold, M.; Clauß, C.; Elmquist, H.; Friedrich, M.; Junghanns, A.; Mauss, J.; Neumerkel, D.; Olsson, H.; Viel A.: Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In: 9th International Modelica Conference (2012), S. 173-184.

BLUM 2017

Blum, F.: Was können VR und AR in Konstruktion und Co.?. In: ke Next (2017). <<https://www.ke-next.de/automation/sensorik-messtechnik/was-koennen-vr-und-ar-in-konstruktion-co-201.html?page=3>> - 31.10.2017.

BLUMENDORF ET AL. 2007

Blumendorf, M.; Feuerstack, S.; Albayrak, S.: Multimodal User Interaction in Smart Environments: Delivering Distributed User Interfaces. In: European Conference on Ambient Intelligence (2007), S. 113-120.

BOOCH ET AL. 2005

Booch, G.; Rumbaugh, J.; Jacobson, I.: The unified modeling language user guide. Addison-Wesley 2005. ISBN: 9780321267979.

BRACHT & FAHLBUSCH 2001

Bracht, U.; Fahlbusch, W.: Einsatz von Virtual Reality-Systemen in der Fabrik- und Anlagenplanung. (2001). <https://www.imab.tu-clausthal.de/fileadmin/homes/mf_imab/dateien/veroeffentlichungen/78.pdf> - 31.08.2017.

BRECHER ET AL. 2010

Brecher, C.; Schapp, L. O.; Groche, P.; Hoffmann, H.; Klocke, F.; Reisgen, U.: Produktionstechnik im nächsten Jahrzehnt: innovativ und integrativ. In: 10. Karlsruher Arbeitsgespräche Produktionsforschung 2010 – Produktion in Deutschland hat Zukunft. Karlsruhe, 9./10. März, V.1-15.

BREZINA ET AL. 2011

Brezina, T.; Hadas, Z.; Vetiska, J.: Using of Co-simulation ADAMS-SIMULINK for Development of Mechatronic Systems. In: MECHATRONIKA, 14th International Symposium, IEEE (2011), S. 59-64.

BROY & STOLEN 2001

Broy, M.; Stølen, K.: Specification and Development of Interactive Systems. New York: Springer 2001. ISBN: 9781461300915.

BUBB 2000

Bubb, H.: Ergonomie. Heidelberg: Spektrum Akademischer Verlag 2000. <<http://www.spektrum.de/lexikon/psychologie/ergonomie/4295>> - 27.01.2018.

BUBB 2012

Bubb, H.: Information Ergonomics. In: Stein, M. & Sandl, P. (Hrsg.): Information Ergonomics. Berlin, Heidelberg: Springer 2012, S. 23-57. ISBN: 9783642258404.

BUBB 2015

Bubb, H.: Methoden der ergonomischen Fahrzeugentwicklung. In: Bubb, H. et al. (Hrsg.): Automobilergonomie. Wiesbaden: Springer Vieweg 2015, S. 583-616. ISBN: 9783834822970.

BUDWIG ET AL. 2009

Budwig, M.; Jeong, S.; Kelkar, K.: When user experience met agile: a case study. In: CHI'09 Extended Abstracts on Human Factors in Computing Systems (2009), S. 3075-3084.

BULLINGER 1994

Bullinger, H.-J.: Ergonomie. Produkt- und Arbeitsplatzgestaltung. Wiesbaden: Springer Fachmedien Wiesbaden 1994. ISBN: 9783663120957.

BÜLTERMANN & BRANDSTÄTTER 2014

Bültermann, R.; Brandstätter, M.: Austausch von Modellen zwischen SysML und Modelica zur Simulation. In: Maurer, M. & Schulze, S.-O. (Hrsg.): Tag des Systems Engineering 2014 (2014), S. 289-298.

BUNGARTZ ET AL. 2013

Bungartz, H.-J.; Zimmer, S.; Buchholz, M.; Pflüger, D.: Modellbildung und Simulation. Eine anwendungsorientierte Einführung. 2., überarbeitete Auflage. Berlin, Heidelberg: Springer Spektrum 2013. ISBN: 9783642376559.

BURDEA & COIFFET 2003

Burdea, G. C.; Coiffet, P.: Virtual Reality Technology. 2. Auflage. Hoboken, New Jersey: John Wiley & Sons 2003.

BURMESTER 2008

Burmester, M.: Usability-Engineering. In: Weber, W. (Hrsg.): Kompendium Informationsdesign. Berlin, Heidelberg: Springer 2008, S. 321-360. ISBN: 9783540698173.

BUSCHMANN ET AL. 1998

Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M.: Pattern-orientierte Softwarearchitektur. Ein Pattern-System. Addison Wesley 1998. ISBN: 9783827312822.

CADET ET AL. 2017

Cadet, M.; Sinnwell, C.; Fischer, J.; Rosen, R.; Stephan, N.; Meissner, H.: Referenzentwicklungsprozess für cybertronische Produkte und Produktionssysteme. In: Eigner, M. et al. (Hrsg.): Modellbasierter Entwicklungsprozess cybertronischer Systeme. Berlin: Springer Vieweg 2017, S. 45-63. ISBN: 9783662551233.

CAO ET AL. 2011

Cao, Y.; Liu, Y.; Paredis, C.: System-Level Model Integration of Design and Simulation for Mechatronic Systems Based on SysML. In: Mechatronics 21 (2011) 6, S. 1063-1075.

CAO 2017

Cao, J.: What Is a Mockup: The Final Layer of UI Design. <<https://www.uxpin.com/studio/blog/what-is-a-mockup-the-final-layer-of-ui-design/>> - 01.09.2017.

CHAN & LEUNG 1996

Chan, L. W.; Leung, T. P.: Spiral Design Model for Consumer Mechatronics Products. In: Mechatronics 6 (1996) 1, S. 35-51.

COHEN ET AL. 2003

Cohen, D.; Lindvall, M.; Costa, P.: Agile Software Development. DACS SOAR Report 11 (2003).

COHN 2004

Cohn, M.: User Stories Applied. For Agile Software Development. Boston: Pearson Education 2004. ISBN: 9780321205681.

COPA-DATA 2017

COPA-DATA: Die zenon Produktfamilie. <<https://satomec.ch/produkt/zenon-logic>> - 31.08.2017.

CRUZ-NEIRA ET AL. 1992

Cruz-Neira, C.; Sandin, D. J.; DeFanti, T. A.; Kenyon, R. V.; Hart, J. C.: The CAVE: audio visual experience automatic virtual environment. In: Communications of the ACM 35 (1992) 6, S. 64-72.

CZARNECKI & HELSEN 2003

Czarnecki, K.; Helsen, S.: Classification of Model Transformation Approaches. In: Proceedings of the 2nd Workshop on Generative Techniques in the Context of MDA (2003), S. 1-17.

DA SILVA & PATON 2000

da Silva, P. P.; Paton, N. W.: UMLi: The Unified Modeling Language for Interactive Applications. In: 3rd International Conference on the Unified Modeling Language UML (2000), S. 117-132.

DANEELS & SALTER 1999

Daneels, A.; Salter, W.: What is SCADA?. International Conference on Accelerator and Large Experimental Physics Control Systems (1999), S. 339-343.

DASSAULT SYSTEMES 2017

Dassault Systèmes: DYMOLA Systems Engineering. Multi-Engineering Modellierung und Simulation auf Basis von Modelica und FMI. <<https://www.3ds.com/de/produkte-und-services/catia/produkte/dymola/>> - 01.09.2017.

DESIKAN & RAMESH 2008

Desikan, S.; Ramesh, G.: Software Testing. Principles and Practices. Pearson Education 2006. ISBN: 9788177581218.

DIN EN 62264-1

DIN EN 62264-1: Integration von Unternehmensführungs- und Leitsystemen - Teil 1: Modelle und Terminologie. Berlin: Beuth 2014.

DIN EN ISO 9241

DIN EN ISO 9241: Ergonomie der Mensch-System-Interaktion. Berlin: Beuth 2016.

DIN EN ISO 9241-11

DIN EN ISO 9241-11: Ergonomie der Mensch-System-Interaktion – Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte. Berlin: Beuth 2016.

DIN EN ISO 9241-110

DIN EN ISO 9241-110: Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung. Berlin: Beuth 2008.

DIN EN ISO 9241-210

DIN EN ISO 9241-210: Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme. Berlin: Beuth 2011.

DORAU 2011

Dorau, R.: Emotionales Interaktionsdesign. Gesten und Mimik interaktiver Systeme. Berlin, Heidelberg: Springer 2011. ISBN: 3642031013.

DOROCIAC ET AL. 2014

Dorociak, R.; Dumitrescu, R.; Gausemeier, J.; Iwanek, P.: Specification Technique CONSENS for the Description of Self-Optimizing Systems. In: Gausemeier, J. et al. (Hrsg.): Design Methodology for Intelligent Technical Systems. Berlin, Heidelberg: Springer 2014, S. 119-127. ISBN: 9783642454349.

DRATH 2010

Drath, R.: Datenaustausch in der Anlagenplanung mit AutomationML. Integration von CAEX, PLCopen XML und COLLADA. Heidelberg, New York: Springer 2010. ISBN: 9783642046735.

DUDEN 2018

Duden: Geste (2018). <<https://www.duden.de/rechtschreibung/Geste>> - 20.01.2018.

DUSTIN ET AL. 2001

Dustin, E.; Rashka, J.; Paul, J.: Software automatisch testen. Berlin, Heidelberg: Springer 2001. ISBN: 9783540676393.

DZIDA 1988

Dzida, W.: Modellierung und Bewertung von Benutzerschnittstellen. In: Software Kurier (1988) 1, S. 13-28.

EHRENSPIEL & MEERKAMM 2017

Ehrlenspiel, K.; Meerkamm, H.: Integrierte Produktentwicklung. 6., vollständig überarbeitete und erweiterte Auflage. München, Wien: Hanser 2017. ISBN: 9783446440890.

EIGNER ET AL. 2012

Eigner, M.; Anderl, R.; Stark, R.: Interdisziplinäre Produktentstehung. In: Anderl, R. et al. (Hrsg.): Smart Engineering. Interdisziplinäre Produktentstehung (acatech Diskussion), Heidelberg: Springer 2012, S. 7-16. ISBN: 9783642293719.

EIGNER 2013

Eigner, M.: Modellbasierte Virtuelle Produktentwicklung auf einer Plattform für System Lifecycle Management. In: Sandler, U. (Hrsg.): Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM. Berlin, Heidelberg: Springer 2013, S. 91-110. ISBN: 9783642369162.

EITELWEIN ET AL. 2012

Eitelwein, O.; Malz, S.; Weber, J.: Erfolg durch Modularisierung. In: Controlling & Management 56 (2012) 2, S. 79-84.

ESTEFAN 2007

Estefan, J.A.: Survey of Model-Based System Engineering (MBSE) Methodologies. In: INCOSE MBSE Focus Group 25 (2007) 8, S. 1-70.

FELDMANN ET AL. 2012

Feldmann, S.; Fuchs, J.; Vogel-Heuser, B.: Modularity, Variant and Version Management in Plant Automation – Future Challenges and State of the Art. In: 12th International Design Conference (2012), S. 1689-1698.

FELDMANN ET AL. 2014

Feldmann, S.; Kernschmidt, K.; Vogel-Heuser, B.: Combining a SysML-based modeling approach and semantic technologies for analyzing change influences in manufacturing plant models. In: Proceedings of the 47th CIRP Conference on Manufacturing Systems 17 (2014), S. 451-456.

FLANAGAN 2007

Flanagan, D.: JavaScript. Das umfassende Referenzwerk. Köln: O'Reilly 2007. ISBN: 9783897214910.

FOLLMER 2012

Follmer, M.: Systemmodelle zur Entwicklung mechatronischer Produkte. Dissertation, Johannes Kepler Universität Linz. Linz: 2012.

FRAUNHOFER IAO 1997

Fraunhofer IAO: Style Guide Werkzeugmaschinen. Stuttgart: Fraunhofer IRB 1997. ISBN: 3816751318.

FRIEDENTHAL ET AL. 2015

Friedenthal, S.; Moore, A.; Steiner, R.: A Practical Guide to SysML. The Systems Modeling Language. 3. Auflage. Waltham: Morgan Kaufmann 2015. ISBN: 9780128002025.

FRIEDRICH 2011

Friedrich, M.: Parallel Co-Simulation for Mechatronic Systems. Dissertation, Technische Universität München. München: 2011.

FRITZSON 2010

Fritzson, P.: Principles of object-oriented modeling and simulation with Modelica 2.1. John Wiley & Sons 2010. ISBN: 9780471471639.

GALITZ 2007

Galitz, W. O.: The Essential Guide to User Interface Design. An Introduction to GUI Design Principles and Techniques. Indianapolis: Wiley Publishing 2007. ISBN: 9780470053423.

GAROFALO 2011

Garofalo, R.: Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern. Microsoft Press 2011. ISBN: 9780735650923.

GAMMA ET AL. 2015

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns. Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software. Frechen: mitp 2015. ISBN: 9783826699047.

GAUSEMEIER ET AL. 2000

Gausemeier, J.; Flath, M.; Möhringer, S.: Entwicklungsumgebung Mechatronik. In: Symposium „Design for X“ 11 (2000), S. 95-100.

GAUSEMEIER & FELDMANN 2006

Gausemeier, J.; Feldmann, K.: Integrative Entwicklung räumlicher elektronischer Baugruppen. München, Wien: Hanser 2006. ISBN: 9783446404670.

GAUSEMEIER ET AL. 2013

Gausemeier, J.; Dumitrescu, R.; Steffen, D.; Czaja, A.; Wiederkehr, O.; Tschirner, C.: Systems Engineering in der industriellen Praxis (2013).

<https://www.hni.uni-paderborn.de/fileadmin/Fachgruppen/Seniorprofessur_Gausemeier/systemsengineerings/Studie_Systems-Engineering.pdf> - 31.08.2017.

GAUSEMEIER ET AL. 2014

Gausemeier, J.; Rammig, F. J.; Schäfer, W.: Design Methodology for Intelligent Technical Systems. Berlin, Heidelberg: Springer 2014. ISBN: 9783642454349.

GERKE 2015

Gerke, W.: Technische Assistenzsysteme. Berlin, München, Boston: De Gruyter 2015. ISBN: 9783110343700.

GEUß & BUBB 1994

Geuß, H.; Bubb, H.: RAMSIS - A Newly Developed Anthropometric Measuring and Analysing Technique. In: Proceedings of the 12th Triennial Congress of the International Ergonomics Association (1994), S. 444-446.

GODERIS 2008

Goderis, S.: On the Separation of User Interface Concerns: A Programmer's Perspective on the Modularisation of User Interface Code. Dissertation, Vrije Universiteit Brussel. Brüssel: VUBPRESS 2008. ISBN: 9789054874973.

GORECKY ET AL. 2017

Gorecky, D.; Schmitt, M.; Loskyll, M.: Mensch-Maschine-Interaktion im Industrie 4.0-Zeitalter. In: Vogel-Heuser, B. et al. (Hrsg.): Handbuch Industrie 4.0. Bd. 4. Allgemeine Grundlagen. 2. Auflage. Wiesbaden: Spring Vieweg 2017, S. 217-235. ISBN: 9783662532539.

GOTHELF & SEIDEN 2013

Gothelf, J.; Seiden, J.: Lean UX. Applying Lean Principles to Improve User Experience. Sebastopol: O'Reilly 2013. ISBN: 9781449366841.

GÖPFERT & STEINBRECHER 2000

Göpfert, J.; Steinbrecher, M.: Modulare Produktentwicklung leistet mehr. In: Harvard Business Manager (2000) 3, S. 1-17.

GÖTZE 1995

Götze, R.: Dialogmodellierung für multimediale Benutzerschnittstellen. Stuttgart, Leipzig: Teubner 1995. ISBN: 9783815420645.

GRABOWSKI & GEIGER 1997

Grabowski, H.; Geiger, K.: Neue Wege zur Produktentwicklung. Stuttgart: Raabe 1997. ISBN: 9783886493470.

GUALA 2002

Guala, F.: Models, Simulations, and Experiments. In: Model-Based Reasoning: Science, Technology, Values (2002), S. 59-74.

GÜNTHER & VELTEN 2014

Günther, M.; Velten, K.: Mathematische Modellbildung und Simulation. Eine Einführung für Wissenschaftler, Ingenieure und Ökonomen. Weinheim: WILEY-VCH 2014. ISBN: 9783527412174.

GÜRTLER & MEYER 2013

Gürtler, J.; Meyer, J.: Desing Thinking. Offenbach: Gabal 2013. ISBN: 9783869364865.

HABERFELLNER ET AL. 2015

Haberfellner, R.; Weck, O. de; Fricke, E.; Vössner, S.: Systems Engineering - Grundlagen und Anwendung. 13., aktualisierte Auflage. Zürich: Orell Füssli 2015. ISBN: 9783280040683.

HACKEL 2011

Hackel, M.: Auf dem Weg zum interdisziplinären mechatronischen Konstruktionsprozess: Entwickelnde Arbeitsforschung im Maschinen- und Anlagenbau. Dissertation, Fernuniversität Hagen. Frankfurt am Main: Peter Lang 2011. ISBN: 9783631610541. (Europäische Hochschulschriften, Bd. 1003).

HACKENBERG ET AL. 2015

Hackenberg, G.; Richter, C.; Zäh, M.: IMoMeSA – Abschlussbericht. Integrierte modellbasierte Entwicklung mechatronischer Systeme im Maschinen- und Anlagenbau. (2015). <<https://mediatum.ub.tum.de/attfile/1253882/incoming/2015-May/570232.pdf>> - 03.09.2017.

HACKENBERG ET AL. 2016

Hackenberg, G.; Gleirscher, M.; Stocker, T.; Richter, C.; Reinhart, G.: MaCon: Consistent Cross-Disciplinary Conception of Manufacturing Systems. In: IFAC-PapersOnLine 49 (2016) 12, S. 1175-1180.

HÄRTER 2016

Härter, H.: Ein kundenspezifisches HMI-System im Baukastenprinzip. (2016). <<http://www.elektronikpraxis.vogel.de/hmi/articles/532529/>> - 03.09.2017.

HARTMANN 1996

Hartmann, S.: The World as a Process: Simulations in the Natural and Social Sciences. In: Modelling and simulation in the social sciences from the philosophy of science point of view (1996), S. 77-100.

HECHTMAN 2017

Hechtman, S. A.: Web-based HMI: An emerging trend? <<http://www.automation.com/library/articles-white-papers/hmi-and-scada-software-technologies/web-based-hmi-an-emerging-trend>> - 31.08.2017.

HECKEL 2006

Heckel, R. Graph transformation in a nutshell. In: Electronic notes in theoretical computer science 148 (2006) 1, S. 187-198.

HEHENBERGER 2011

Hehenberger, P.: Computergestützte Fertigung. Eine kompakte Einführung. Berlin, Heidelberg: Springer 2011. ISBN: 9783642134746.

HEINECKE 2012

Heinecke, A. M.: Mensch-Computer-Interaktion. Basiswissen für Entwickler und Gestalter. 2., überarbeitete und erweiterte Auflage. Berlin, Heidelberg: Springer 2012. ISBN: 9783642135064.

HELLER 2012

Heller, J.: Experimentelle Psychologie. Eine Einführung. München: Oldenbourg 2012. ISBN: 9783486702675.

HENSEL 2011

Hensel, T.: Modellbasierter Entwicklungsprozess für Automatisierungslösungen. Dissertation, Technische Universität München. München: Utz 2011. ISBN: 9783831641673. (Forschungsberichte IWB, Nr. 258).

HERCZEG 2009

Herczeg, M.: Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme. 3. Auflage. München: Oldenbourg 2009. ISBN: 9783486587258.

HIRSCH-KREINSEN 2014

Hirsch-Kreinsen, H.: Wandel von Produktionsarbeit – Industrie 4.0. TU Dortmund (Soziologisches Arbeitspapier Nr. 38/2014).

HOGAN 2011

Hogan, B. P.: HTML5 & CSS3. Webentwicklung mit den Standards von morgen. Köln: O'Reilly 2011. ISBN: 9783897213166.

HOLM ET AL. 2013

Holm, T.; Schröck, S.; Fay, A.; Jäger, T.; Löwen, U.: Engineering von Mechatronik und Software in automatisierten Anlagen: Anforderungen und Stand der Technik. In: Software Engineering (Workshops) 2013, S. 261-272.

HOLTEN 2000

Holten, R.: Entwicklung einer Modellierungstechnik für Data Warehouse-Fachkonzepte. In: Proceedings der MobIS-Fachtagung (2000), S. 3-21.

HUMMEL 2011

Hummel, B.: Integrated Behavior Modeling of Space-Intensive Mechatronic Systems. Dissertation, Technische Universität München. München: 2011. <<https://mediatum.ub.tum.de/doc/993204/993204.pdf>> - 03.09.2017.

HÜRSCH & LOPES 1995

Hürsch, W. L.; Lopez, C.: Separation of concerns. In: Technical Report CCS-95-03 (1995).

INCOSE 2015

INCOSE: Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities. 4. Auflage. Hoboken, New Jersey: John Wiley & Sons 2015. ISBN: 9781118999400.

ISERMANN 2007

Isermann, R.: Mechatronic systems – Innovative products with embedded control. In: Control Engineering Practice 16 (2008), S. 14-29.

ISERMANN 2008

Isermann, R.: Mechatronische Systeme. Grundlagen. 2. Auflage. Berlin: Springer 2008. ISBN: 9783540323365.

ISHII & ULLMER 2012

Ishii, H.; Ullmer, B.: Tangible User Interfaces. In: Jacko, J. A. (Hrsg.): The Human-Computer Interaction Handbook. 3. Auflage. CRC Press 2012, S. 465-490.

JACKO 2012

Jacko, J. A. (Hrsg.): The Human-Computer Interaction Handbook. 3. Auflage. CRC Press 2012. ISBN: 9781439829448.

JANSCHER 2010

Janschek, K.: Systementwurf mechatronischer Systeme. Methoden – Modelle – Konzepte. Berlin, Heidelberg: Springer 2010. ISBN: 9783540788768.

JOOS ET AL. 2003

Joos, M.; Rötting, M.; Vliechkovsky, B. M.: Bewegungen des menschlichen Auges: Fakten, Methoden und innovative Anwendungen. In: Rickheit, G. et al. (Hrsg.): Psycholinguistik. Berlin: Walter de Gruyter 2003, S. 142-167.

KAKLANIS ET AL. 2016

Kaklanis, N.; Biswas, P.; Mohamad, Y.; Gonzalez, M. F.; Peissner, M.; Langdon, P.; Tzovaras, D.; Jung, C.: Towards standardisation of user models for simulation and adaptation purposes. In: Universal Access in the Information Society 15 (2016) 1, S. 21-48.

KALLMEYER 1998

Kallmeyer, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Dissertation, Universität Paderborn. Paderborn: Heinz-Nixdorf-Institut 1998. ISBN: 3931466418. (HNI-Verlagsschriftenreihe 42).

KAUFMANN 2015

Kaufmann, T.: Geschäftsmodelle in Industrie 4.0 und dem Internet der Dinge. Wiesbaden. Springer Vieweg 2015. ISBN: 9783658102715.

KAUSLER 2008

Kausler, B.: Automatisierte Bedienung graphischer Benutzungsoberflächen beim Systemfunktionsfunktionstest. Dissertation, Technische Universität München. München: 2008.

KEMPA & MANN 2005

Kempa, M.; Mann, Z. A.: Model Driven Architecture. In: Informatikspektrum 28 (2005) 4, S. 298-302.

KERNSCHMIDT & VOGEL-HEUSER 2013

Kernschmidt, K.; Vogel-Heuser, B.: An interdisciplinary SysML based modeling approach for analyzing change influences in production plants to support the engineering. In: IEEE International Conference on Automation Science and Engineering (2013), S. 1113-1118.

KERNSCHMIDT ET AL. 2014

Kernschmidt, K.; Vogel-Heuser, B.; Barbieri, G.; Fantuzzi, C.: Interdisziplinäre modellbasierte Entwicklung mechatronischer Systeme basierend auf SysML zur Steigerung der Wiederverwendung. In: VDI-Kongress Automation 2014, S. 1-13.

KEBLER 2014

Keßler, A.: Computer-Aided Software Engineering. In: Eigner, M. et al. (Hrsg.): Modellbasierte virtuelle Produktentwicklung. Berlin, Heidelberg: Springer Vieweg 2014, S. 161-174. ISBN: 9783662438152.

KLEIN 2016

Klein, T.: Agile Engineering im Maschinen- und Anlagenbau. Dissertation, Technische Universität München. München: Utz 2016. ISBN: 9783831645985. (Forschungsberichte IWB, Nr. 323).

KLEMMER ET AL. 2000

Klemmer, S. R.; Sinha, A. K.; Chen, J.; Landay, J. A.; Aboobaker, N.; Wang, A.: Suede: a Wizard of Oz prototyping tool for speech user interfaces. In: Proceedings of the 13th annual ACM symposium on user interface software and technology (2000), S. 1-10.

KNOLL 2011

Knoll, A.: Vom Smartphone an die Maschine. (2011).
<<http://www.elektroniknet.de/automation/sonstiges/artikel/81092/>> - 31.08.2017.

KNOLL 2014

Knoll, A.: Taster und Touchscreen auf einem Panel. (2014).
<<http://www.computer-automation.de/steuerungsebene/bedienen-beobachten/artikel/114147/>> - 31.08.2017.

KOCH ET AL. 1991

Koch, M.; Reiterer, H.; Tjoa, A. M.: Software-Ergonomie. Gestaltung von EDV-Systemen - Kriterien, Methoden und Werkzeuge. Wien, New York: Springer 1991. ISBN: 9783211822883.

KOLSTER 2014

Kolster, D.: Handlungsorientierte, multimodale Werkzeugmaschinen-Benutzerschnittstellen. Dissertation, RWTH Aachen. Aachen: Apprimus 2014. ISBN: 9783863592141.

KRAUß 2003

Krauß, L.: Entwicklung und Evaluation einer Methodik zur Untersuchung von Interaktionsgeräten für Maschinen- und Prozessbediensysteme mit grafischen Benutzungsoberflächen. Dissertation, Technische Universität Kaiserslautern. Kaiserslautern: 2003. (Fortschrittsberichte pak, Bd. 7).

KRAUSE ET AL. 2007

Krause, F.-L.; Franke, H.-J.; Gausemeier J.: Innovationspotenziale in der Produktentwicklung. München, Wien: Hanser 2007. ISBN: 3446406670.

LAMBRECHT 2014

Lambrecht, J.: Natürlich-räumliche Industrieroboterprogrammierung auf Basis markerloser Gestenerkennung und mobiler Augmented Reality. Dissertation, Technische Universität Berlin. Berlin: 2014.

LANGMANN 2017

Langmann, R.: Taschenbuch der Automatisierung. 3., neu bearbeitete Auflage. München: Hanser 2017. ISBN: 9783446446649.

LAURIG 1990

Laurig, W.: Grundzüge der Ergonomie. Erkenntnisse und Prinzipien. 3., vollständig überarbeitete Auflage. Berlin, Köln: Beuth 1990. ISBN: 9783410365808.

LEVIN ET AL. 2000

Levin, E.; Pieraccini, R.; Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. In: IEEE Transactions on Speech and Audio Processing 8 (2000) 1, S. 11-23.

LINGUATEC 2017

Linguatec: Voice Pro 12. <<https://www.linguatec.de/spracherkennung/voice-pro-12-standard/>> - 31.08.2017.

LITTO ET AL. 2004

Litto, M.; Korajda, I.; Mangold, C.; Angerbauer, R.; Hils, W.; Lerche, M.:
Baukastenbasiertes Engineering mit Föderal. Verband Deutscher Maschinen- und
Anlagenbau. Frankfurt am Main: VDMA 2004. ISBN: 9783816304784.

LOPEZ-JAQUERO ET AL. 2003

López-Jaquero, V.; Montero, F.; Molina, J. P.; Fernández-Caballero, A.; González,
P.: Model-based Design of Adaptive User Interfaces through Connectors. In:
International Workshop on Design, Specification, and Verification of Interactive
Systems. Berlin, Heidelberg: Springer 2003, S. 245-257.

LÜCKEL ET AL. 2000

Lückel, J.; Koch, T.; Schmitz, J.: Mechatronik als integrative Basis für innovative
Produkte. In: VDI-Tagung: Mechatronik - Mechanische/ Elektrische
Antriebstechnik (2000), S. 1-26.

LÜDER ET AL. 2010

Lüder, A.; Hundt, L.; Foehr, M.; Holm, T.; Wagner, T.; Zaddach, J.-J.:
Manufacturing System Engineering with Mechatronical Units. In: IEEE
International Conference on Emerging Technologies and Factory Automation
(2010).

LUYTEN 2004

Luyten, K.: Dynamic User Interface Generation for Mobile and Embedded Systems
with Model-Based User Interface Development. PhD Thesis, Transnational
University Limburg: School of Information Technology. Limburg: 2004.

MACHATE ET AL. 2014

Machate, J.; Schäffler, A.; Ackermann, S.: A touch of future – Einsatzbereiche für
Multi-Touch-Anwendungen. In: Schlegel, T. (Hrsg.): Multi-Touch. Interaktion
durch Berührung. Berlin, Heidelberg: Spring Vieweg 2014, S. 13-44. ISBN:
9783642361128.

MARKOWITZ 1996

Markowitz, J. A.: Using Speech Recognition. Prentice Hall PTR 1996.
ISBN: 9780131863217.

MARSCHALL 2005

Marschall, F.: Modelltransformationen als Mittel der modellbasierten Entwicklung
von Software-Systemen. Dissertation, Technische Universität München. München:
2005. <<https://mediatum.ub.tum.de/doc/601766/601766.pdf>> - 01.09.2017.

MÄRTIN ET AL. 2013

Märtin, C.; Herdin, C.; Engel, J.: Patterns and Models for Automated User Interface Construction – In Search of the Missing Links. In: Kurosu, M. (Hrsg.): Human-Computer Interaction, Part I, HCII 2013. Berlin, Heidelberg: Springer 2013, S. 401-410. ISBN: 9783642392313.

MAUDERER 2010

Mauderer, M.: Ein Beitrag zur Planung und Entwicklung von rekonfigurierbaren mechatronischen Systemen – am Beispiel von starren Fertigungssystemen. Dissertation, Technische Universität München. München: Utz 2010. ISBN: 9783831641260. (Forschungsberichte IWB, Nr. 250).

MAYER & PANTFÖRDER 2014

Mayer, F.; Pantförder, D.: Unterstützung des Menschen in Cyber-Physical-Production-Systems. In: Bauernhansl, T. et al. (Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Wiesbaden: Springer Vieweg 2014, S. 481-491. ISBN: 9783658046811.

MAYHEW 1999

Mayhew, D.: The Usability Engineering Lifecycle. A Practitioner's Handbook for User Interface Design. Morgan Kaufmann 1999. ISBN: 9781558605619.

MEHLER-BICHLER & STEIGER 2014

Mehler-Bichler, A.; Steiger, L.: Augmented Reality: Theorie und Praxis. 2., überarbeitete Auflage. München: Oldenbourg 2014. ISBN: 9783110353846.

MEIXNER & GÖRLICH 2008

Meixner, G.; Görlich, D.: Aufgabenmodellierung als Kernelement eines nutzerzentrierten Entwicklungsprozesses für Bedienoberflächen. In: Fachtagung Modellierung Berlin, Workshop „Verhaltensmodellierung: Best Practices und neue Erkenntnisse“ (2008).

MEIXNER 2010

Meixner, G.: Entwicklung einer modellbasierten Architektur für multimodale Benutzungsschnittstellen. Dissertation, Technische Universität Kaiserslautern. Kaiserslautern: 2010. (Fortschrittsberichte pak, Bd. 21).

MEIXNER ET AL. 2011

Meixner, G.; Seissler, M.; Breiner, K.: Model-driven Useware Engineering. In: Hussmann, H. et al. (Hrsg.): Model-Driven Development of Advanced User Interfaces. Berlin, Heidelberg: Springer 2011, S. 1-26. ISBN: 9783642145612.

MILLAN 2002

Millan, J.: Brain-computer interfaces. In: Arbib, M. A. (Hrsg.): Handbook of Brain Theory and Neural Networks. Cambridge: MIT Press 2002.

NATHAN 2014

Nathan, A.: WPF 4.5 Unleashed. Pearson Education 2014. ISBN: 9780672336973.

NATTERMANN & ANDERL 2013

Nattermann, R.; Anderl, R.: The W-Model Using Systems Engineering for Adaptronics. In: Conference on Systems Engineering Research (2013), S. 937-946.

NAUMANN ET AL. 2014

Naumann, M.; Dietz, T.; Kuss, A.: Mensch-Maschine-Interaktion. In: Bauernhansl, T. et al. (Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Wiesbaden: Spring Vieweg 2014, S. 509-523. ISBN: 9783658046811.

NAVARRE ET AL. 2009

Navarre, D.; Palanque, P.; Ladry, J.-F.; Barboni, E.: ICOs: A Model-Based User Interface Description Technique dedicated to Interactive Systems Addressing Usability, Reliability and Scalability. In: ACM Transactions on Computer-Human Interaction 16 (2009) 4.

NEWMAN 1995

Newman, R.: Head-up Displays: Designing the way ahead. Avebury Aviation. ISBN: 9780291398116.

NICKLAS 2003

Nicklas, F.: Modulare Baukastensysteme. (2003). <http://www.frank-nicklas.de/download/2003_09_29_Modularisierung_VDI_V4.pdf> - 06.09.2017.

NIELSEN 1993

Nielsen, J.: Usability Engineering. San Francisco: Morgan Kaufmann 1993. ISBN: 9780125184069.

OEHMEN 2016

Oehmen, J.: Risiko- und Chancenmanagement in der Produktentwicklung. In: Lindemann, U. (Hrsg.): Handbuch Produktentwicklung. München: Hanser 2016, S. 59-98. ISBN: 9783446445185.

OETINGER 1988

Oetinger, R.: Benutzergerechte Software-Entwicklung. Berlin, Heidelberg: Springer 1988. ISBN: 9783540191353.

OMG 2010

Object Management Group: OMG Unified Modeling Language™ (OMG UML), Infrastructure. (2010). <<http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/>> - 05.09.2017.

OPPERMANN & REITERER 1992

Oppermann, R.; Reiterer, H.: Evaluation von Benutzerschnittstellen.
In: Wirtschaftsinformatik 34 (1992) 3, S. 283-293.

OVIATT 2012

Oviatt, S.: Multimodal Interfaces. In: Jacko, J. A. (Hrsg.): The Human-Computer Interaction Handbook. 3. Auflage. CRC Press 2012, S. 405-430.

PAHL ET AL. 2003

Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K. H.: Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung. 5., neu bearbeitete und erweiterte Auflage. Berlin, Heidelberg: Springer 2003. ISBN: 9783662091876.

PATERNO ET AL. 1997

Paterno, F.; Mancinic, C.; Meniconi, S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In: Human-Computer Interaction INTERACT'97 (1997), S. 362-369.

PATERNO ET AL. 2009

Paterno, F.; Santoro, C.; Spano, L. D.: MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. In: ACM Transactions on Computer-Human Interaction 16 (2009) 4.

PEISSNER & HIPPEL 2013

Peissner, M.; Hipp, C.: Potenziale der Mensch-Technik Interaktion für die effiziente und vernetzte Produktion von morgen. Studie des Fraunhofer Institut für Arbeitswissenschaft und Organisation. Fraunhofer Verlag, 2013. ISBN: 9783839605639.

PEISSNER 2014

Peissner, M.: Entwurfsmusterbasierter Ansatz für adaptive Benutzungsschnittstellen zur Überwindung von Nutzungsbarrieren. Dissertation, Universität Stuttgart. Stuttgart: Fraunhofer Verlag 2014. ISBN: 9783839607695. (Schriftenreihe zu Arbeitswissenschaft und Technologiemanagement, Bd. 12).

PETRASCH 2007

Petrasch, R.: Model Based User Interface Design: Model Driven Architecture™ und HCI Patterns. In: GI Softwaretechnik-Trends. Mitteilungen der Gesellschaft für Informatik 27 (2007) 3, S. 5-10.

PIETQUIN 2004

Pietquin, O.: A framework for unsupervised learning of dialogue strategies. Dissertation, Faculté Polytechnique de Mons. Mons: 2004.

PILLAGE ET AL. 1999

Pillage, T. L.; Rohrer, R. A.; Visweswariah, C.: Electronic Circuit & Systems Simulation Methods. McGraw-Hill 1999. ISBN: 0070501696.

PONN & LINDEMANN 2011

Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Heidelberg; Dordrecht; London; New York: Springer 2011. ISBN: 9783642205798.

POPPER 1962

Popper, K.: Die Logik der Sozialwissenschaften. In: Kölner Zeitschrift für Soziologie und Sozial-Psychologie 14 (1962), S. 233-248.

POSSEL-DÖLKEN 2010

Possel-Dölken, F.: Mechatronische Modularisierung im Sondermaschinenbau. (2010). <http://www.transmechatronic.de/uploads/tx_vitramemberadmin/literature/B34_113.pdf> - 02.09.2017.

PREIM & DACHSELT 2015

Preim, B.; Dachselt, R.: Interaktive Systeme. Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces. 2. Auflage. Berlin Heidelberg: Springer Vieweg 2015. ISBN: 9783642452468.

PRITSCHOW 1997

Pritschow, G.: HÜMNOS – Trendwende in der Steuerungstechnik. Kurzbericht im Rahmen der EMO (Werkzeugmaschinen-Weltausstellung) 1997, ISW Universität Stuttgart, 1997.

PUERTA 1997

Puerta, A. R.: A Model-Based Interface Development Environment. In: IEEE Software 14 (1997) 4, S. 40-47.

PURSCHKE ET AL. 1998

Purschke, F.; Schulze, M.; Zimmermann, P.: Virtual reality: new methods for improving and accelerating the development process in vehicle styling and design. In: Computer Graphics International IEEE Proceedings 1998, S. 789-797.

QAMAR ET AL. 2009

Qamar, A.; Doring, C.; Wikander, J.: Designing mechatronic systems, a model-based perspective, an attempt to achieve SysML-Matlab/Simulink model integration. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics 2009, S. 1306-1311.

REIF ET AL. 2007

Reif, R.; Walch, D.; Wulz, J.: Einsatz von Virtual und Augmented Reality: Studie zur menschengerechten Simulation und Prozessunterstützung im logistischen Umfeld. (2007).

<http://www.fml.mw.tum.de/fml/images/Publikationen/ForLog_Studie_VR-AR_2007_Abgabe.pdf> - 31.08.2017.

REINHART & PATRON 2003

Reinhart, G.; Patron, C.: Integrating augmented reality in the assembly domain. In: CIRP Annals - Manufacturing Technology 52 (2003) 1, S. 5-8.

REINHART ET AL. 2009

Reinhart, G.; Meling, F.; Zuber, E.; Grimmer, K.: Baukastenbasiertes disziplinübergreifendes Engineering von Produktionsanlagen. Herausforderungen und Lösungsansätze. In: Gauemeier, J. et al. (Hrsg.): Entwurf mechatronischer Systeme. 6. Paderborner Workshop 2009.

REINHART ET AL. 2010

Reinhart, G.; Hensel, T.; Meling, F.: Automatisierung der HMI-Projekterstellung. (2010). <<https://www.copadata.com/?eID=download&pluginName=CpdownloadsSingle&controller=Download&action=download&uuid=47af7a90-802b-41d7-8156-c016b1a6ec01>> - 03.09.2017.

REINHART ET AL. 2017

Reinhart, G.; Bengler, K.; Dollinger, C.; Intra, C.; Lock, C.; Popova-Dlogosch, S.; Rimpau, C.; Schmidtler, J.; Teubner, S.; Vernim, S.: Der Mensch in der Produktion von Morgen. In: Reinhart, G. (Hrsg.): Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. München: Hanser 2017, S. 51-88. ISBN: 9783446454583.

REN ET AL. 2011

Ren, Z.; Meng, J.; Yuan, J.; Zhang, Z.: Robust hand gesture recognition with kinect sensor. In: Proceedings of the 19th ACM international conference on Multimedia (2011), S. 759-760.

REUTH ET AL. 2002

Reuth, R.; Künzer, A.; Boldt, T.; Schmidt, L.; Luczak, H.; Murrenhoff, H.: Modellbasierte Gestaltung einer multimodalen Benutzungsschnittstelle zur Unterstützung von Greif- und Spannprozessen beim 3D-Laserschweißen. In: Marzi, R. et al. (Hrsg.): Bedienen und Verstehen. Düsseldorf: VDI-Verlag 2002, S. 55-70.

REUTHER 2003

Reuther, A.: useML – systematische Entwicklung von Maschinenbediensystemen mit XML. Dissertation, Technische Universität Kaiserslautern. Kaiserslautern: 2003. (Fortschrittsberichte pak, Bd. 8).

RICHTER & FLÜCKIGER 2013

Richter, M.; Flückiger, M.: Usability Engineering kompakt – Benutzbare Produkte gezielt entwickeln. 3. Auflage. Berlin, Heidelberg: Springer 2013.
ISBN: 9783642348310.

RICHTER & REINHART 2014

Richter, C.; Reinhart, G.: Interdisciplinary and Model-Based Development of User Interfaces for Production Plants. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics (2014), S. 536-541.

RICHTER & REINHART 2015

Richter, C.; Reinhart, G.: Klassifikation von Bedientechnologien für den Maschinen- und Anlagenbau. Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF) 110 (2015) 5, S. 261-266.

RICHTER 2016

Richter, C.: Entwicklung innovativer Bedienkonzepte für den Maschinen- und Anlagenbau. In: iwb Newsletter 24 (2016) 1, S. 11.

RICHTER ET AL. 2016

Richter, C.; Ahrens, M.; Hehenberger, P.; Krottil, S.; Stich, P.; Reinhart, G.; Wiesinger, A.; Wimmer, A.: Model-based Development and Virtual Commissioning in Practice: A Novel Approach to Establish Innovative Development Methods in Industrial Environments. In: Proceedings of TMCE (2016), S. 93-104.

RIES 2012

Ries, E.: Lean Startup. Schnell, risikolos und erfolgreich Unternehmen gründen. Redline Verlag, 2012. ISBN: 9783864142710.

RODDECK 2016

Roddeck, W.: Einführung in die Mechatronik. 5. überarbeitete und erweiterte Auflage. Wiesbaden: Springer Vieweg 2016. ISBN: 9783658158439.

RÖDIGER 1987

Rödiger, K.-H.: Arbeitsorientierte Gestaltung von Dialogsystemen im Büro- und Verwaltungsbereich. Dissertation, Technische Universität Berlin, Berlin 1987.

ROBMANN & HEINZE 2010

Roßmann, J.; Heinze, F.: Modeling and Simulation of Malfunctions in Automation Systems. In: IEEE International Conference on Emerging Technologies and Factory Automation (2010).

ROBMANN 2012A

Roßmann, J.: eRobotics: The Symbiosis of Advanced Robotics and Virtual Reality Technologies. In: ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (2012), S. 1395-1402.

ROBMANN 2012B

Roßmann, J.: Control by 3D Simulation – A New eRobotics Approach to Control Design in Automation. In: International Conference on Intelligent Robotics and Applications (2012), S. 186-197.

RÜCKERT & PAECH 2009

Rückert, J.; Paech, B.: Kontextsensitive Dialogmodellierung. In: Software Engineering (2009), S. 63-74.

SAUTER 2010

Sauter, T.: The Three Generations of Field-Level Networks - Evolution and Comptibility Issues. In: IEEE Transactions on Industrial Electronics 57 (2010) 11, S. 3585-3595.

SCHEER 2013

Scheer, A.W.: Industrie 4.0 – Wie sehen Produktionsprozesse im Jahr 2020 aus?. IMC AG, 2013.

SCHENK & RIGOLL 2010

Schenk, J.; Rigoll, G.: Mensch-Maschine-Kommunikation. Grundlagen von sprach- und bildbasierten Benutzerschnittstellen. Berlin, Heidelberg: Springer 2010. ISBN: 9783642054563.

SCHNELL 2003

Schnell, G. (Hrsg.): Bussysteme in der Automatisierungs- und Prozesstechnik. Grundlagen, Systeme und Trends der industriellen Kommunikation. 5., überarb. und erw. Auflage. Wiesbaden: Vieweg 2013. ISBN: 9783663094722.

SCHOLZ 2013

Scholz, M: HMI mit neuer Klasse. (2013).

<<http://www.computer-automation.de/steuerungsebene/industrie-pc/artikel/97047/>>
- 31.08.2017.

SCHRECK 2014

Schreck, A.: Die Rolle des HMI für die Industrie 4.0. viernull-Magazin (2014) 2, S. 49-54.

SCHRÖDER 2003

Schröder, J.: Benchmarking von Entwicklungsbereichen im Maschinenbau. Dissertation, RWTH Aachen. Aachen: 2003.

SCHWABER & BEEDLE 2002

Schwaber, K.; Beedle, M.: Agile Software Development with Scrum. Prentice Hall 2002. ISBN: 9780130676344.

SCHWEITZER 1989

Schweitzer, G.: Mechatronik - Aufgaben und Lösungen. In: VDI Bericht 787 (1989), S. 1-15.

SELLS & GRIFFITHS 2007

Sells, C.; Griffiths, I.: Programming WPF. 2. Auflage. Sebastopol: O'Reilly 2007. ISBN: 9780596510374.

SHARMA 2011

Sharma, K. L. S.: Overview of Industrial Process Automation. Waltham: Elsevier 2011. ISBN: 9780124157798.

SHELLY & ROSENBLATT 2011

Shelly, G.; Rosenblatt, H.: Systems analysis and design. 9. Auflage. Boston: Cengage Learning 2011.

SHNEIDERMAN ET AL. 2017

Shneiderman, B.; Plaisant, C.; Cohen, M.; Jacobs, S.: Designing the User Interface. Strategies for Effective Human-Computer Interaction. 5. Auflage. Pearson Education 2017. ISBN: 9780134748566.

SIEMENS 2017

Siemens: Maschinen- und prozessnahes Bedienen und Beobachten: SIMATIC WinCC flexible. <<http://w3.siemens.com/mcms/human-machine-interface/de/visualisierungssoftware/wincc-flexible/seiten/default.aspx>> - 31.08.2017.

SIEMON 2001

Siemon, E.: Über den Entwurf von Benutzungsschnittstellen technischer Anwendungen mit visuellen Spezifikationsmethoden und Werkzeugen. Dissertation, Technische Universität Darmstadt. Darmstadt: 2001.

SKROTZKI 2011

Skrotzki, R.: Hardware-Ergonomie. (2011). <http://www.imtm-iaw.ruhr-uni-bochum.de/wp-content/uploads/sites/5/2011/08/hardware_ergonomie2011.pdf> - 31.08.2017.

SNYDER 2003

Snyder, C.: Paper Prototyping. The fast and easy way to design and refine user interfaces. San Francisco: Morgan Kaufmann 2003. ISBN: 9781558608702.

SONG ET AL. 2011

Song, Y.; Demirdjian, D.; Davis, R.: Tracking Body and Hands for Gesture Recognition: NATOPS Aircraft Handling Signals Database. In: IEEE International Conference on Automatic Face & Gesture Recognition (2011), S. 500-506.

SORENSEN & MIHAILESCU 2010

Sorensen, E.; Mihailescu, M. I.: Model-View-ViewModel (MVVM) Design Patten using Windows Presentation Foundation (WPF) Technology. MegaByte Journal 9 (2010) 4, S. 1-19.

SPATH ET AL. 2013

Spath, D.; Ganscher, O.; Gerlach, S.; Hämmerle, M.; Krause, T.; Schlund, S.: Produktionsarbeit der Zukunft – Industrie 4.0. Studie des Fraunhofer Instituts für Arbeitswissenschaft und Organisation (IAO). Fraunhofer Verlag 2013.

SPATH & DANGELMAIER 2016

Spath, D.; Dangelmaier, M.: Produktentwicklung Quo Vadis. In: Lindemann, U. (Hrsg.): Handbuch Produktentwicklung. München: Hanser 2016, S. 3-8. ISBN: 9783446445185.

STEGERWALD 2014

Stegerwald, F.: Webfähige HMI - modular und auf unterschiedliche Plattformen übertragbar. (2014). <<http://www.gti.de/wp-content/uploads/2014/11/Webfaehige-HMI-modular-und-kundenspezifisch-auf-unterschiedliche-Plattformen-portierbar.pdf>> - 31.08.2017.

STONE 2001

Stone, R.: Virtual Reality for interactive training: an industrial practitioner's viewpoint. In: International Journal of Human-Computer Studies 55 (2001) 4, S. 699-711.

STONE ET AL. 2005

Stone, D. L.; Jarrett, C.; Woodroffe, M.; Minocha, S.: User interface design and evaluation. San Francisco: Morgan Kaufmann 2005. ISBN: 9780120884360.

SUH 2001

Suh, N. P.: Axiomatic Design. Advances and Applications. Oxford University Press 2001. ISBN: 9780195134667.

SY 2007

Sy, D.: Adapting Usability Investigations for Agile User-centered Design. In: Journal of Usability Studies 2 (2007) 3, S. 112-132.

TAI 2007

Tai, L.-C.: The GUI Toolkit, Framework Page. <<http://www.atai.org/guitool/>> - 31.08.2017.

TH WILDAU 2017

TH Wildau: Gestensteuerung. <<http://www.tfh-wildau.de/sbruntha/Material/VR/Websites-T09/Datenhandschuh/gesten.html>> - 31.08.2017.

THANNER ET AL. 2010

Thanner, T.; Schäfer, V.; Rao, I. S.: Multitouch Displays mit Infrarot. (2010). <<http://www.elektronikpraxis.vogel.de/multitouch-displays-mit-infrarot-a-302791/>> - 31.08.2017.

THIEL ET AL. 2008

Thiel, K.; Meyer, H.; Fuchs, F.: MES - Grundlage der Produktion von morgen. Effektive Wertschöpfung durch die Einführung von Manufacturing Execution Systems. München: Oldenbourg 2008. ISBN: 9783835631403.

TIDWELL 2011

Tidwell, J.: Designing Interfaces. Patterns for Effective Interaction Design. 2. Auflage. Sebastopol: O'Reilly 2011. ISBN: 9781449379704.

TIMPE ET AL. 2002

Timpe, K.-P.; Jürgensohn, T.; Kolrep, H.: Mensch-Maschine-Systemtechnik. Konzepte, Modellierung, Gestaltung, Evaluation. 2. Auflage. Düsseldorf: Symposion 2002. ISBN: 9783933814838.

TÖNNIS 2010

Tönnis, M.: Augmented Reality: Einblicke in die erweiterte Realität. Heidelberg: Springer 2010. ISBN: 9783642141799.

TREDER 2016

Treder, M.: Wireframing, Prototyping, Mockuping – What's the Difference?. (2016). <<https://designmodo.com/wireframing-prototyping-mockuping/>> - 01.09.2017.

ULRICH & HILL 1976

Ulrich, P.; Hill, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil I). In: Wirtschaftswissenschaftliches Studium: Zeitschrift für Ausbildung und Hochschulkontakt 5 (1976) 7, S. 304-309.

ULRICH ET AL. 1984

Ulrich, H.; Dyllick, T.; Probst, G.: Management. Bern: P. Haupt 1984.

URBAS ET AL. 2011

Urbas, L.; Hennig, S.; Hager, H.; Doherr, F.; Braune, A.: Towards context adaptive HMIs in process industries. In: IEEE International Conference on Industrial Informatics (2011), S. 244-249.

VAJNA ET AL. 2009

Vajna, S.; Weber, C.; Bley, H.; Zeman, K.: CAx für Ingenieure. Eine praxisbezogene Einführung. 2. Auflage. Berlin, Heidelberg: Springer 2009. ISBN: 9783540360384.

VANDERDONCKT 2008

Vanderdonckt, J.: Model-driven engineering of user interfaces: Promises, successes, failures, and challenges. In: Proceedings of 5th Annual Romanian Conference on Human-Computer Interaction (2008), S. 1-10.

VANDERDONCKT & CALLEROS 2008

Vanderdonckt, J.; Calleros, J.: UsiXML, a User Interface Model and Language Engineering approach. In: W3C Model Based UI (2008).

VDI/VDE 2013

VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (Hrsg.):
Automation 2020 – Bedeutung und Entwicklung der Automation bis zum Jahr
2020. Thesen und Handlungsfelder.
<https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/GMA_Automation_2020_Internet_2Auflage.pdf> - 28.08.2017.

VDI/VDE 2015

VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (Hrsg.):
Automation 2025 – Thesen und Handlungsfelder.
<http://jahresbericht.vdi.de/fileadmin/user_upload/VDI-Thesen_und_Handlungsfelder_Automation_2025.pdf> - 29.08.2017.

VDI 2206

VDI 2206: VDI Richtlinie: Entwicklungsmethodik mechatronischer Systeme.
Berlin: Beuth 2004.

VDI 2221

VDI 2221: VDI Richtlinie: Methodik zum Entwickeln und Konstruieren technischer
Systeme und Produkte. Berlin: Beuth 1993.

VDI/VDE 2422

VDI/VDE 2422: VDI/VDE Richtlinie: Entwicklungsmethodik für Geräte mit
Steuerung durch Mikroelektronik. Berlin: Beuth 1994.

VDI 3633

VDI 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen.
Berlin: Beuth 2014.

VDI 3633 – BLATT 1

VDI 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen.
Blatt 1: Grundlagen. Berlin: Beuth 2014.

VDI/VDE 3850

VDI/VDE 3850: VDI/ VDE Richtlinie: Gebrauchstaugliche Gestaltung von
Benutzungsschnittstellen für technische Anlagen. Berlin: Beuth 2014.

VDI/VDE 3850 – BLATT 1

VDI/VDE 3850: VDI/ VDE Richtlinie: Gebrauchstaugliche Gestaltung von
Benutzungsschnittstellen für technische Anlagen. Blatt 1: Konzepte, Prinzipien und
grundsätzliche Empfehlungen. Berlin: Beuth 2014.

VDMA 2004

VDMA: Leitfaden Software-Ergonomie – Gestaltung von Bedienoberflächen. Frankfurt am Main: VDMA 2004. ISBN: 9783816304807.

VDMA 2005

VDMA: ProMiS: Projektmanagement für interdisziplinäre Systementwicklungen. Frankfurt am Main: VDMA 2005. ISBN: 9783816305033.

VDMA 2010

VDMA: aquimo – Ein Leitfaden für Maschinen- und Anlagenbauer. Frankfurt am Main: VDMA 2010. ISBN: 9783816305897.

VDMA 2015

VDMA: MEPRONA – Anforderungen und Methoden im mechatronischen Engineering. Frankfurt am Main: VDMA 2015. ISBN: 9783816306788.

VERL ET AL. 2009

Verl, A.; Müller, V.; Haubelt, A.: Baukastenbasiertes simulationsgestütztes Engineering. In: A&D Kompendium 2009/2010. München: Publish-Industry 2009, S. 32-34.

VERMEIR 2013

Vermeir, N.: Windows 8 App Projects. XAML and C# Edition. Apress 2013. ISBN: 9781430250654.

VOGEL-HEUSER 2013

Vogel-Heuser, B.: Automation als Enabler für Industrie 4.0 in der Produktion auf Basis von Cyber-Physical Systems. In: Vogel-Heuser, B. (Hrsg.): Engineering von der Anforderung bis zum Betrieb. Kassel: kassel university press GmbH 2013, S. 1-4. ISBN: 9783862193783.

VOGEL-HEUSER 2017

Vogel-Heuser, B.: Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik. In: Vogel-Heuser, B. et al. (Hrsg.): Handbuch Industrie 4.0. Bd. 4. Allgemeine Grundlagen. 2. Auflage. Wiesbaden: Spring Vieweg 2017, S. 37-48. ISBN: 9783662532539.

VOGL 2009

Vogl, W.: Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern. Dissertation, Technische Universität München. München: Utz 2009. ISBN: 9783831608690. (Forschungsberichte IWB, Nr. 228).

WACHSMUTH ET AL. 2008

Wachsmuth, I.; Lenzen, M.; Knoblich, G.: Embodied communication in humans and machines. New York: Oxford University Press 2008. ISBN: 9780199231751.

WEILKIENS 2016

Weilkiens, T.: SYSMOD – The Systems Modeling Toolbox. Pragmatic MBSE with SysML. 2. Auflage. MBSE4U 2016. ISBN: 9783981787580.

WIESEND 2016

Wiesend, S.: AR- & VR-Lösungen im Unternehmen. (2016).

<<https://www.computerwoche.de/a/ar-und-vr-loesungen-im-unternehmen,3217983>>
- 31.08.2017.

WINKLER ET AL. 2007

Winkler, M.; Heinrich, M.; Behring, A.; Steinmetz, J.; Dargie, W.: EMODE – ein Ansatz zur werkzeugunterstützten Modellierung multimodaler, adaptiver Benutzerschnittstellen. In: GI Jahrestagung 1 (2007), S. 437-442.

WINTER ET AL. 2012

Winter, D.; Holt, E.-M.; Thomaschewski, J.: Persona driven agile development. Build up a vision with personas, sketches and persona driven user stories. In: Proceedings of the 7th Conference on Information Systems and Technologies (2012).

WUCHERER 2004

Wucherer, K.: Mechatronik – Gemeinsame Herausforderung für Maschinenbau, Steuerungstechnik und Informatik. In: Klocke, F.; Pritschow, G. (Hrsg.): Autonome Produktion. Berlin, Heidelberg: Springer 2004, S. 177-192. ISBN: 9783642621437.

WÜNSCH 2007

Wünsch, G.: Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme. Dissertation, Technische Universität München. München: Utz 2007. ISBN: 9783831607952. (Forschungsberichte IWB, Nr. 215).

ZAFIROV & ROUBANOV 2014

Zafirov, R.; Roubanov, D.: Elektrik und Elektronik. In: Eigner, M. et al. (Hrsg.): Modellbasierte virtuelle Produktentwicklung. Berlin, Heidelberg: Springer Vieweg 2014, S. 137-160. ISBN: 9783662438152.

ZHENG ET AL. 2011

Zheng X. S.; Wang, M.; Matos, G.; Zhang, S.: Streamlining User Experience Design and Development: Roles, Tasks and Workflow of Applying Rich Application Technologies. In: Jacko, J. A. (Hrsg.): Human-Computer Interaction, Part I. HCII 2011. Lecture Notes in Computer Science 6761. Berlin, Heidelberg: Springer 2011. ISBN: 9783642216015.

ZÜHLKE 2004

Zühlke, D.: Useware-Engineering für technische Systeme. Berlin, Heidelberg: Springer 2004. ISBN: 9783540206477.

ZÜHLKE 2012

Zühlke, D.: Nutzergerechte Entwicklung von Mensch-Maschine-Systemen. Useware Engineering für technische Systeme. 2., neu bearb. Auflage. Berlin, Heidelberg: Springer 2012. ISBN: 9783642220739.

Anhang

A1 Detailbetrachtung des IMoMeSA-Ansatzes von HACKENBERG ET AL.

Aufbauend auf der Einführung der modellbasierten Entwicklungsmethodik von HACKENBERG ET AL. (2015) in Abschnitt 3.3.2 erfolgt in diesem Abschnitt eine Detailbetrachtung des IMoMeSA-Ansatzes, die zum Verständnis der vorliegenden Arbeit beitragen kann. Dabei konzentrieren sich die nachfolgenden Ausführungen auf eine Beschreibung der einzelnen Elemente der erarbeiteten Modellierungstechnik und des definierten Prozessablaufs für die Konzeption von mechatronischen Systemen. Für die weiteren Aspekte des IMoMeSA-Ansatzes, die nicht direkt zum Verständnis der vorliegenden Arbeit notwendig sind, sei auf HACKENBERG ET AL. (2015) verwiesen.

Elemente der Modellierungstechnik

Die nachfolgende Erläuterung der einzelnen Elemente der Modellierungstechnik von HACKENBERG ET AL. (2015) gliedert sich in eine textuelle Beschreibung, um deren Einsatz während der Konzeption mechatronischer Systeme zu definieren, sowie eine Spezifikation auf einer Metamodellebene. Für die Darstellung des Metamodells wird das Klassendiagramm aus der UML verwendet, wobei zur Visualisierung der Beziehungen die Assoziation, die Komposition und die Generalisierung zum Einsatz kommen. Weiterhin sind innerhalb der einzelnen Modellierungselemente die relevanten Attribute mit ihrem Datentyp skizziert.

Das zentrale Element des IMoMeSA-Ansatzes bildet die *Komponente*, die für eine hierarchische Zerlegung eines mechatronischen Systems in einzelne, voneinander losgelöste und wiederverwendbare Systembestandteile zuständig ist. Auf der Ebene des Metamodells wird bewusst nicht zwischen mechatronischen und disziplinspezifischen Komponenten differenziert, da sich diese Unterscheidung im Verlauf der Entwicklung automatisch durch die jeweiligen Inhalte einer Komponente (Ports, Verhalten, Subkomponenten, etc.) ergibt. Üblicherweise sind mechatronische Komponenten jedoch eher im oberen Bereich der Komponentenhierarchie vorzufinden, während disziplinspezifische Komponenten den Unterbau eines mechatronischen Systems bilden. Intern ist jede Komponente über einen Namen und eine optionale textuelle Beschreibung definiert (vgl. Abbildung A-1). Ferner kann jede Komponente eine beliebige Anzahl weiterer Komponenten beinhalten, um dadurch eine hierarchische Struktur eines mechatronischen Systems aufzubauen.

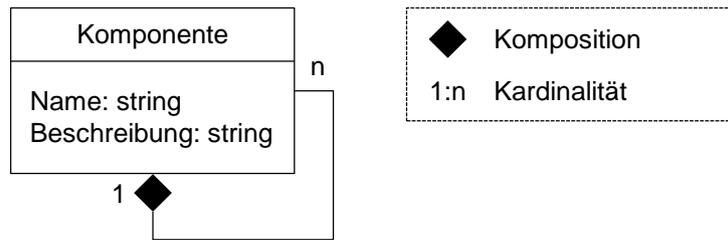


Abbildung A-1: Metamodell des Modellierungselements „Komponente“

Innerhalb von Komponenten bildet das Modellierungselement *Anforderung* eine erste Möglichkeit, um Kundenwünsche und -ansprüche an ein mechatronisches System auf eine informelle Art schnell und einfach zu dokumentieren. Dabei werden Anforderungen innerhalb von Komponenten beliebiger Hierarchiestufen modelliert, um sowohl übergreifende Anforderungen an das Gesamtsystem als auch spezifische Anforderungen an einzelne Komponenten formulieren zu können. Neben dem Aufbau einer Anforderung über einen Namen und eine textuelle Beschreibung besteht weiterhin die Möglichkeit, Anforderungen untereinander zu verknüpfen, um auf eventuelle Abhängigkeiten hinzuweisen (vgl. Abbildung A-2).

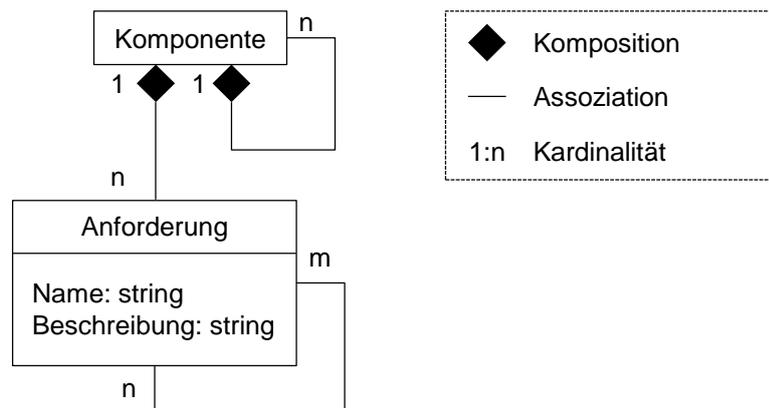


Abbildung A-2: Metamodell des Modellierungselements „Anforderung“

Weiterhin definiert die erarbeitete Modellierungstechnik das Element *Port*, über das die Schnittstelle einer Komponente zu ihrer Umgebung formal beschrieben werden kann. Dazu setzt sich jeder Port zunächst aus einem Namen, einer Beschreibung sowie einer Richtung (Eingabe, Ausgabe oder Ein- und Ausgabe) zusammen (vgl. Abbildung A-3). Weiterhin ist für jeden Port ein konkreter Typ zu wählen, wobei zwischen Material-, Energie-, Daten- und Eventports unterschieden wird. *Materialports* lassen sich dabei weiter in Material Entry und Material Exit Ports unterteilen, an denen Bauteile erzeugt oder entfernt werden können. Darüber hinaus existieren Material Detector und Material Binding Ports, die zum Erkennen von Bauteilen (z. B. an einer Lichtschranke) bzw. zum Schaffen von kinematischen Verbindungen

zwischen einzelnen Körpern (z. B. auf einem Förderband) verwendet werden. Um seine Position und Größe festzulegen, beinhaltet jeder Materialport weiterhin ein einfaches oder zusammengesetztes Volumen. Demgegenüber wird mit *Energieports* die Schnittstelle einer Komponente im Hinblick auf die übertragene elektrische, kinematische Energie oder die Bearbeitungsenergie spezifiziert. Eine Besonderheit bilden die kinematischen Energieports, für die auch die Richtung des Energieflusses anhand einer entsprechenden Achse festgelegt wird. Schließlich definiert die Modellierungstechnik auch *Daten-* und *Eventports*. Erstere kommen zum Einsatz, um die steuerungstechnische Schnittstelle einer Komponente festzulegen. Eventports werden hingegen verwendet, um Einflüsse der Umgebung abzubilden, was u. a. für die Modellierung eines Komponentenausfalls benötigt wird.

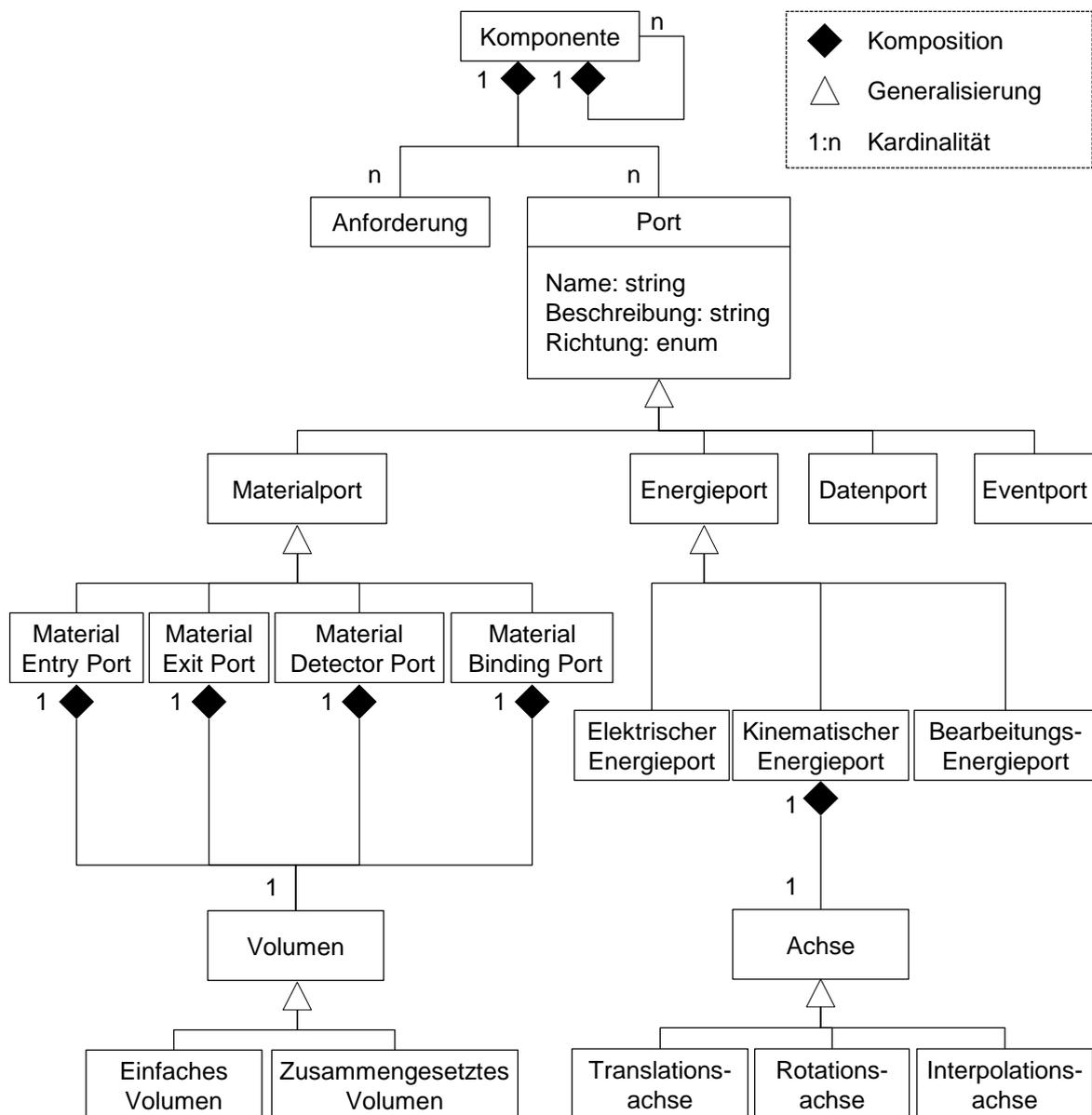


Abbildung A-3: Metamodell des Modellierungselements „Port“

Basierend auf dem Konzept der Ports besteht über das Modellierungselement *Eigenschaft* die Möglichkeit, den an einem Port anliegenden Wert zu überwachen, wie bspw. den Verbrauch an einem elektrischen Energieport. Dabei kann generell zwischen den beiden Eigenschaftsformen *Bedingung* und *Qualität* unterschieden werden. Bedingungen liefern dabei ein binäres Ergebnis, ob der aktuelle Wert an einem Port akzeptabel ist oder nicht. Mit Qualitäten wird demgegenüber ein numerischer Wert zwischen 0 und 1 berechnet, der den Grad der Akzeptanz der aktuellen Portbelegung zum Ausdruck bringt. Für beide Berechnungsschritte kommen boolesche und numerische Ausdrücke sowie Mengenausdrücke zum Einsatz, die selbst miteinander kombiniert werden können, um auch komplexere Berechnungen zu ermöglichen. Innerhalb der Ausdrücke wird weiterhin der zu beobachtende Port referenziert, was im Metamodell in Abbildung A-4 durch die Assoziation zwischen den Ausdrücken und den Ports visualisiert wird.

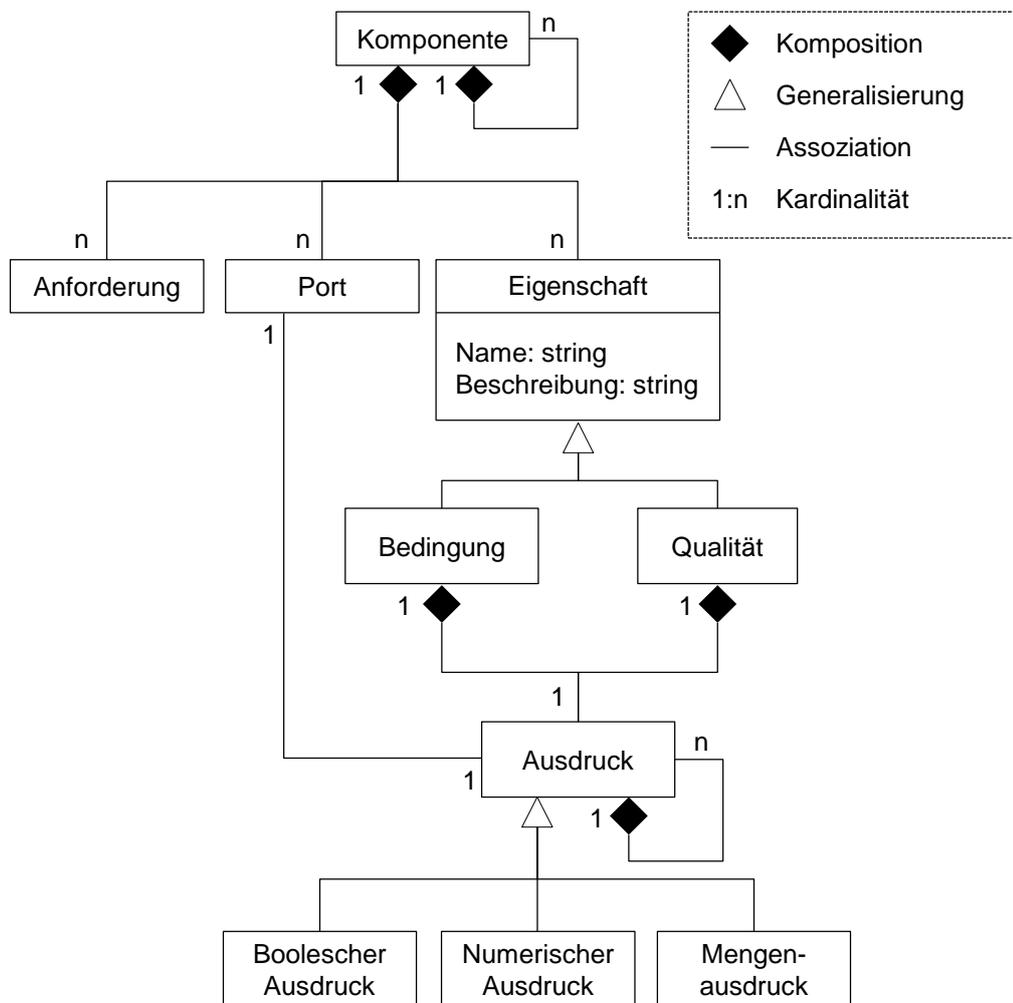


Abbildung A-4: Metamodell des Modellierungselements „Eigenschaft“

Neben den eingeführten Eigenschaften stellt das Modellierungselement *Szenario* den zweiten wichtigen Baustein zur Formalisierung von Anforderungen dar. Dazu wird in Szenarien ein Nachrichten-Reihenfolge-Diagramm modelliert, das definiert, in welcher Abfolge eine betrachtete Komponente mit ihrer Umgebung Material, Energie und Daten austauscht, um eine bestimmte Aufgabe zu erfüllen. Das zugehörige Metamodell (vgl. Abbildung A-5) sieht dazu zunächst Schritte vor, um den grundsätzlichen Ablauf eines Szenarios festzulegen. Der Wechsel zwischen zwei Schritten wird über Transitionen mit entsprechenden Übergangsbedingungen (sog. Wächtern) spezifiziert, wobei die Ausgabeports der betrachteten Komponente beobachtet werden. Sowohl während eines Schritts als auch bei einem Übergang können Aktionen ausgeführt werden, mittels derer die Werte an den Eingabeports der betrachteten Komponente gesetzt werden. An Schritte können auch Eigenschaften angeheftet werden, um bspw. eine maximale Verweildauer für einen Zustand festzulegen. Für die innerhalb der Szenarien notwendigen Berechnungen werden boolesche oder numerische Ausdrücke bzw. Mengen-Ausdrücke verwendet, über die wie bei den Eigenschaften der Bezug zu den Ports der Komponente hergestellt wird. Wichtig ist dabei zu erwähnen, dass ein Szenario auch selbst Material Entry und Exit Ports definieren kann, um während des Ablaufs des Szenarios Bauteile erzeugen oder entfernen zu können. Schließlich können Variablen genutzt werden, um u. a. ein Zwischenspeichern von Informationen zu ermöglichen.

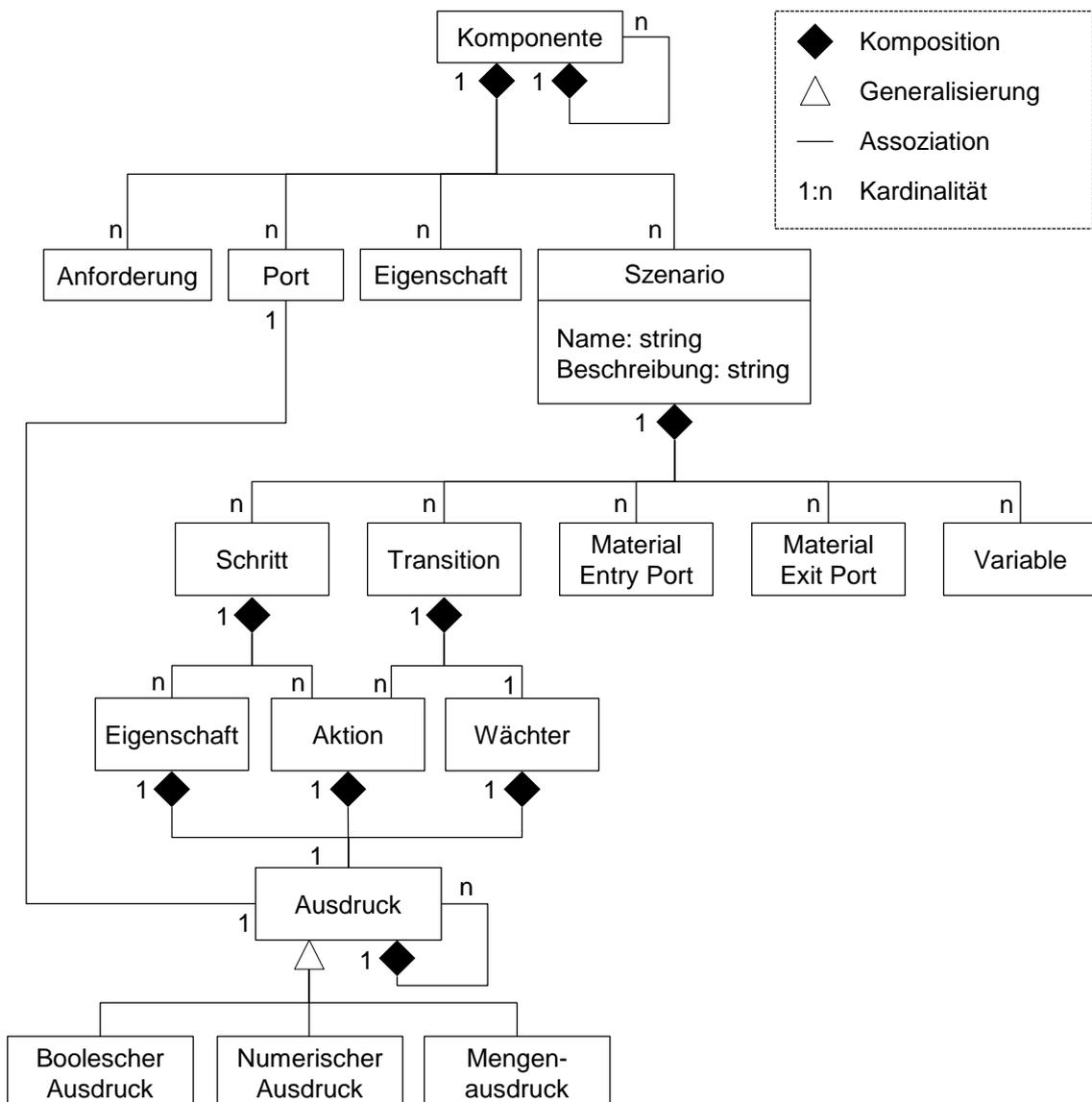


Abbildung A-5: Metamodell des Modellierungselements „Szenario“

Während anhand der zuvor betrachteten Modellierungselemente vor allem Anforderungen an ein mechatronisches System bzw. einzelne Komponenten abgebildet wurden, ermöglicht das Modellierungselement *Monitor* die Gestaltung des technischen Prozesses, der auf einem System oder einer Komponente ablaufen soll. Dabei wird der Prozess ähnlich wie bei Szenarien mit Aktivitäten (anstelle von Schritten) und Transitionen sowie entsprechenden Zustandsübergängen (Wächtern) spezifiziert (vgl. Abbildung A-6). Generell besteht auch die Möglichkeit, Aktionen innerhalb von Aktivitäten oder an Transitionen zu modellieren, allerdings werden diese nur genutzt, um interne Variablen zu beeinflussen, während das Beeinflussen von Ports nicht vorgesehen ist, da Monitore grundsätzlich einen beobachtenden Charakter haben. Daher dienen Monitore vor allem auf übergeordneten Ebenen

dazu, Sollabläufe zu spezifizieren, die bei der Gestaltung der disziplinspezifischen Komponenten einzuhalten sind. Daher bietet sich bei der Verwendung von Monitoren auch der Einsatz von Eigenschaften an, um bspw. die maximale Dauer oder den maximalen Energieverbrauch für einen Prozessschritt zu überwachen.

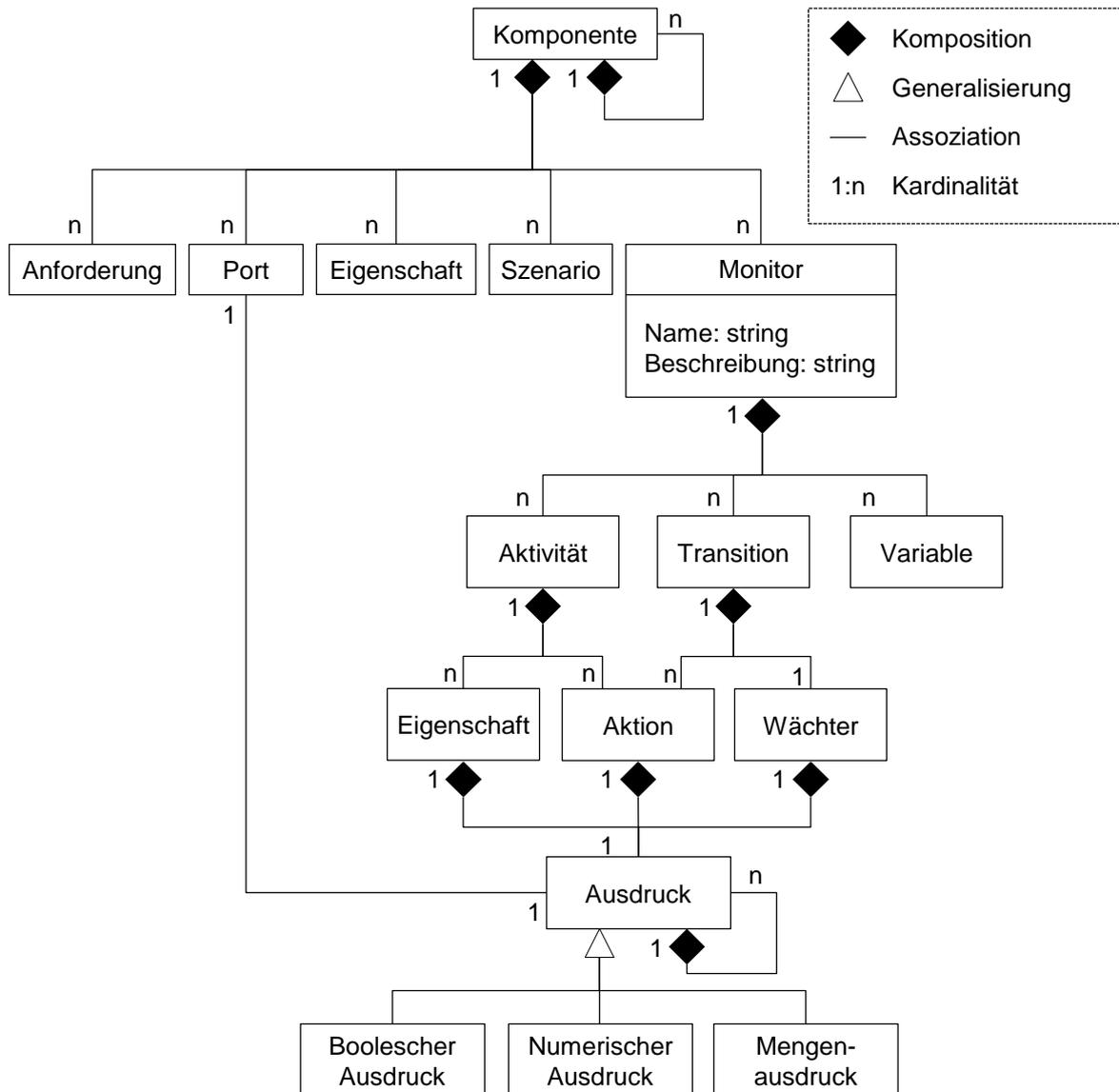


Abbildung A-6: Metamodell des Modellierungselements „Monitor“

Neben ihrer überwachenden Funktion können Monitore auch verwendet werden, um zu entscheiden, ob eine Komponente in entsprechende Subkomponenten unterteilt werden soll. Im Fall einer Dekomposition ist es in der Regel zielführend, die einzelnen Aktivitäten eines Monitors für die Strukturierung der Subkomponenten zu verwenden. Um die abgeleiteten Subkomponenten miteinander zu verbinden, wird das Modellierungselement *Kanal* eingesetzt. In Anlehnung an die Porttypen wird

dabei generell zwischen Energie-, Daten- und Eventkanälen differenziert (vgl. Abbildung A-7). Bei Materialports sind hingegen keine expliziten Verbindungen über Kanäle erforderlich, da sich diese implizit durch Überschneidungen der jeweiligen Volumina im 3D-Modell ergeben.

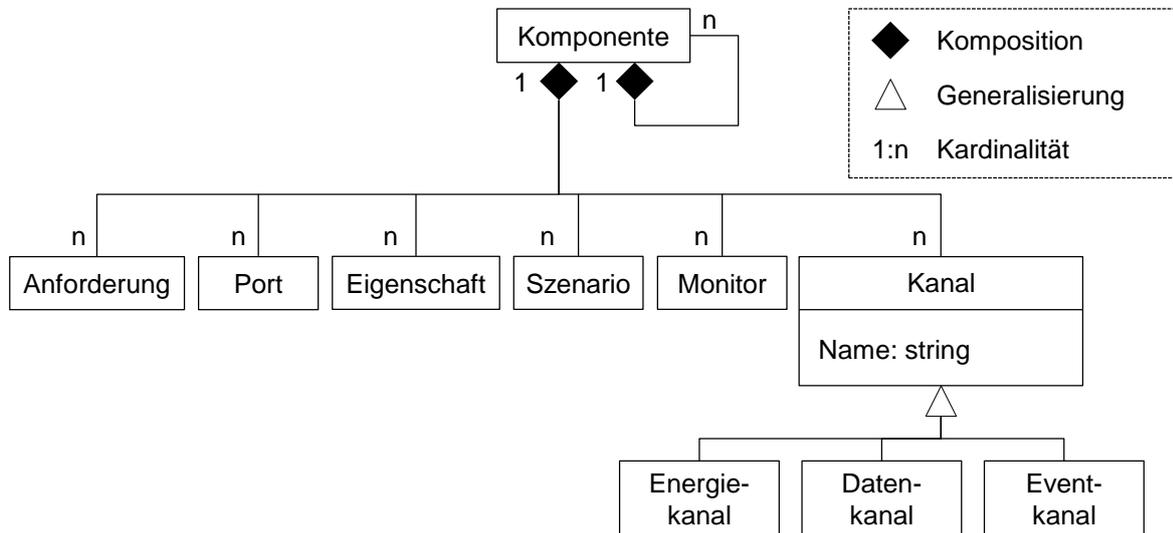


Abbildung A-7: Metamodell des Modellierungselements „Kanal“

Für alle Komponenten, für die eine Zerlegung in Subkomponenten nicht notwendig ist, wie dies z. B. bei Sensoren der Fall ist, kann die jeweilige Funktionsweise mit dem Modellierungselement *Verhalten* festgelegt werden. Auf oberster Ebene wird dabei grundsätzlich zwischen unterschiedlichen Verhaltensmodi unterschieden, die im Wesentlichen zum Einsatz kommen, um zwischen einem Verhalten im Normalfall und möglichen Fehlerfällen zu unterscheiden. Zwischen diesen Modi erfolgt ein Übergang durch Transitionen, die von Eventports aktiviert werden. In jedem Modus wird das eigentliche Verhalten durch einen unterlagerten Zustandsautomaten spezifiziert, der analog zu den Monitoren aufgebaut ist (vgl. Abbildung A-8). An den Wächtern werden die Werte der Eingabe-Port der Komponente ausgewertet, wodurch Zustandsübergänge initiiert werden. Innerhalb der Zustände oder an den Transitionen können Aktionen modelliert werden, durch die Werte an den Ausgabe-Ports der jeweiligen Komponente entsprechend beeinflusst werden. Im Unterschied zu Szenarien und Monitoren ist im Modellierungselement „Verhalten“ bewusst keine Abbildung von Eigenschaften mehr vorgesehen, da alle Anforderungen an ein zu entwickelndes System bereits auf den übergeordneten Ebenen in die Szenarien und Monitore integriert werden können.

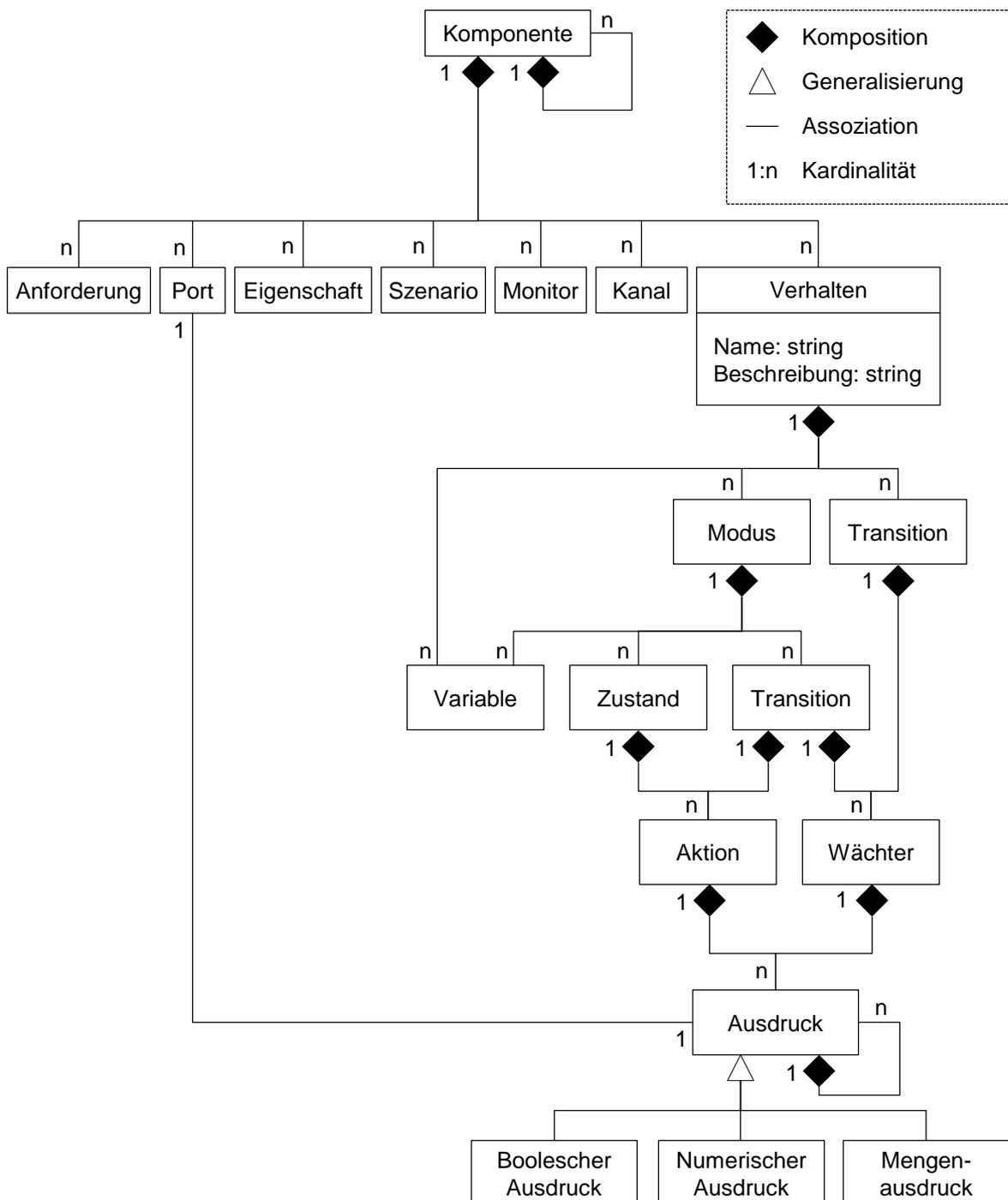


Abbildung A-8: Metamodell des Modellierungselements „Verhalten“

Neben einem Verhalten besitzen spezifische Komponenten auch eine Repräsentation im 3D-Geometriemodell, die mit dem Modellierungselement *Bauteil* abgebildet werden kann. Dabei steht die Modellierung einfacher Geometrien im Vordergrund, um den erforderlichen Bauraum zu definieren und einen kollisionsfreien Materialfluss zu ermöglichen. Dazu kommen wie bei Materialports unterschiedliche

Volumina zum Einsatz. Grundsätzlich stützt sich das Konzept dabei auf den Constructive-Solid-Geometry-Ansatz, der die Definition von einfachen Volumina mit Quadern, Kugeln und Zylindern sowie deren Kombination mit booleschen Operationen, wie der Vereinigung, der Schnittmenge oder der Differenz, ermöglicht (vgl. Abbildung A-9). Über entsprechende Transformationen können die Volumina schließlich im Raum angeordnet sowie in ihrer Orientierung verändert werden.

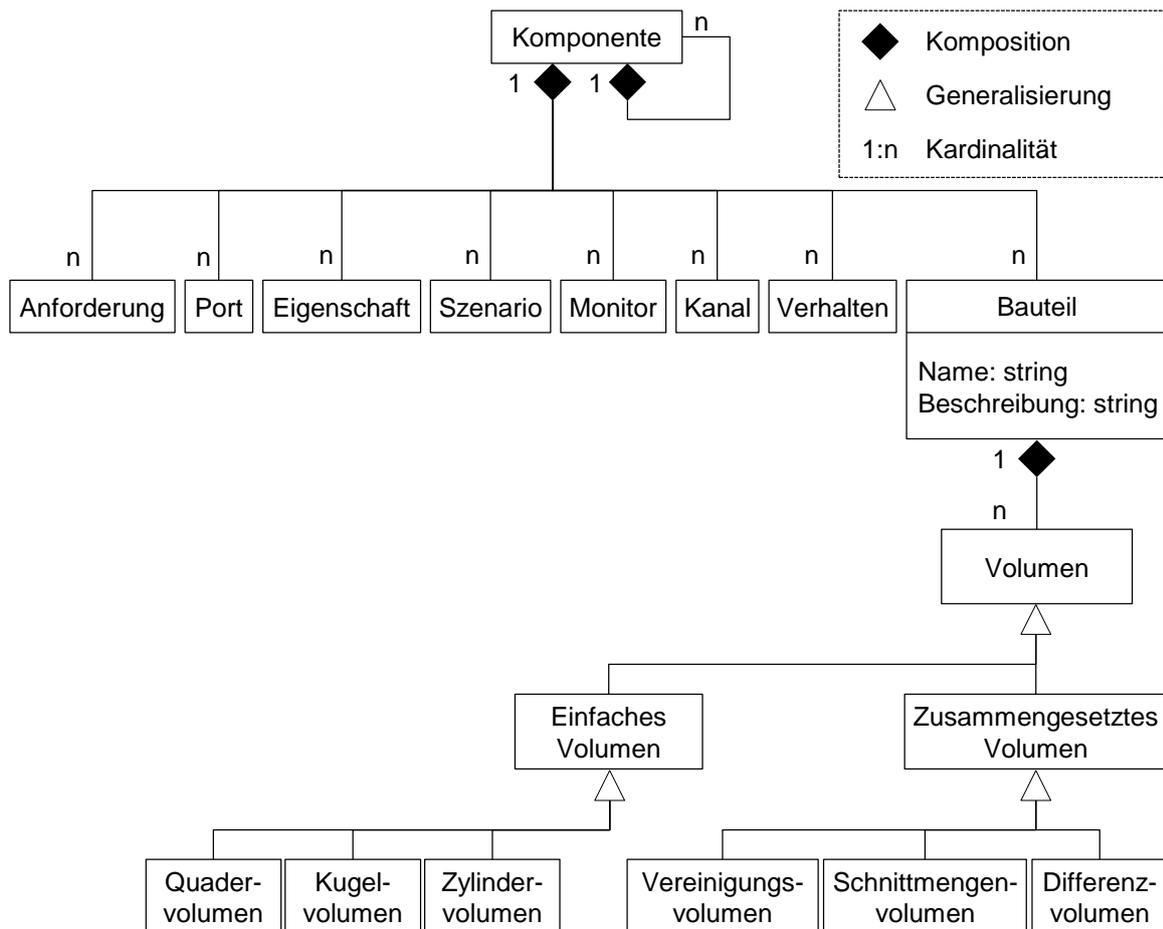


Abbildung A-9: Metamodell des Modellierungselements „Bauteil“

Prozessablauf für die Konzeption mechatronischer Systeme

Basierend auf der übergeordneten Gliederung eines mechatronischen Entwicklungsprozesses in die Phasen der Konzeption, der Verfeinerung und des physikalischen Aufbaus, lassen sich für jede Entwicklungsphase einzelne Schritte identifizieren, die im Nachfolgenden speziell für die Konzeption näher betrachtet werden. Dabei gibt zunächst Abbildung A-10 einen Überblick über diese Schritte, die im Einzelnen aus dem Aufbau der mechatronischen Komponentenstruktur, der Ausgestaltung von modellierten Komponenten sowie der Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten bestehen.

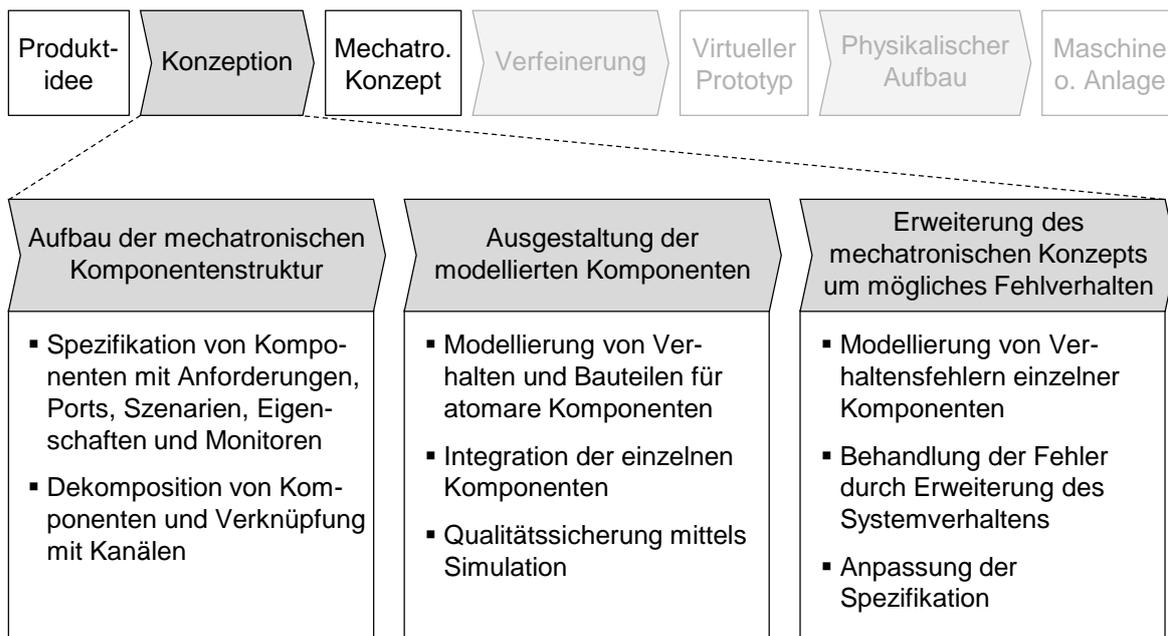


Abbildung A-10: Teilschritte bei der Konzeption mechatronischer Systeme

Auf Basis einer Produktidee für eine neue Maschine oder Anlage, die sich bspw. durch einen Kundenauftrag oder eine interne Marktrecherche ergeben kann, erfolgt zunächst der *Aufbau der mechatronischen Komponentenstruktur*, die zur Realisierung der Produktidee zielführend ist. Dabei wird in einem ersten Schritt das mechatronische System selbst als Komponente modelliert, für das anschließend die Anforderungen gesammelt und dokumentiert werden. Weiterhin können die Anforderungen anhand der Ports, Szenarien und Eigenschaften formalisiert werden, um in der Simulation eine automatisierte Qualitätssicherung zu ermöglichen. Sobald alle Anforderungen auf dieser Ebene abgebildet sind, kann anhand eines Monitors der technische Prozess festgelegt werden, der zur Realisierung der einzelnen Anforderungen benötigt wird. Die Komplexität dieses Prozesses ist dabei ein wichtiger Indikator für die Entscheidung, ob das System in einzelne Subkomponenten zerlegt werden soll. Im Fall einer Dekomposition werden entsprechende Subkomponenten erzeugt, für die im Anschluss die zuvor beschriebenen Modellierungselemente erneut eingesetzt werden, um Anforderungen an diese Komponenten zu definieren und zu formalisieren sowie Teilprozesse anhand von Monitoren festzulegen. Diese Monitore bilden wiederum die Basis für die Entscheidung nach einer erneuten Dekomposition in Subkomponenten. Aus diesem rekursiven Vorgehen resultiert schließlich eine mechatronische Komponentenstruktur des zu entwickelnden Systems, die bei der Ebene endet, auf der eine mechatronische Komponente in ihre rein disziplinspezifischen Bestandteile unterteilt werden kann.

Dadurch besteht die erste Aufgabe in der anschließenden *Ausgestaltung der modellierten Komponenten* in der Aufteilung der mechatronischen Komponenten der untersten Ebene in ihre disziplinspezifischen Bestandteile, wobei erneut Komponenten verwendet werden. Innerhalb dieser Komponenten werden allerdings keine Anforderungen mehr dokumentiert oder formalisiert, vielmehr liegt der Fokus auf der Modellierung des Verhaltens sowie ggf. entsprechender Bauteile für das 3D-Geometriemodell des zu entwickelnden Systems. Sobald alle disziplinspezifischen Bestandteile einer mechatronischen Komponente auf diese Weise modelliert sind, kann die Komponente in der Simulation getestet werden. Dabei fungieren die definierten Szenarien als Testfälle, für die geprüft wird, ob die jeweilige Komponente korrekt auf die Eingaben aus dem Szenario reagiert und ob dabei alle formulierten Bedingungen und Monitore eingehalten werden. Auf diese Weise können die mechatronischen Komponenten der untersten Hierarchieebene losgelöst voneinander erprobt und optimiert werden. Sobald dabei alle Komponenten eine hinreichende Qualität besitzen, können diese auf der nächsthöheren Ebene integriert werden. Bei diesem Integrationsschritt ist es oft notwendig, die einzelnen Subkomponenten durch neue disziplinspezifische Komponenten zu verknüpfen, wie dies bspw. erforderlich wird, wenn zwei Steuerungskomponenten nur durch eine zusätzliche Gesamtsteuerung miteinander kommunizieren können. Nach der Verbindung aller Subkomponenten kann in der Simulation geprüft werden, ob das resultierende Teilsystem den auf dieser Ebene modellierten Szenarien, Bedingungen und Monitoren gerecht wird. Analog zur untersten Hierarchieebene können auch auf dieser Ebene alle mechatronischen Komponenten losgelöst voneinander getestet und optimiert werden, bis alle modellierten Anforderungen entsprechend berücksichtigt sind. Der Integrationsschritt endet, sobald die oberste Hierarchieebene erreicht ist und auch hier die Erfüllung der Anforderungen in der Simulation nachgewiesen werden kann.

Während die ersten beiden Schritte ausschließlich die Modellierung des Normalverhaltens eines mechatronischen Systems adressieren, fokussiert der dritte Schritt die *Erweiterung des mechatronischen Konzepts um mögliches Fehlverhalten*, um die Robustheit eines Systems im späteren Betrieb zu steigern und das System gegen mögliche Verhaltensfehler einzelner Komponenten, wie z. B. einen Sensorausfall, abzusichern. Dazu wird zunächst bewusst ein Verhaltensfehler in einer disziplinspezifischen Komponente modelliert, wofür die im Metamodell vorgesehenen Modi im Modellierungselement „Verhalten“ genutzt werden. Für den Fehlermodus kann anschließend ein eigener Zustandsautomat modelliert werden, der das Verhalten der Komponente im Fehlerfall definiert. Weiterhin wird ein spezieller Eventport

modelliert, der mit der Transition zwischen den Modi der Komponente verknüpft wird, um den Übergang zwischen den zwei alternativen Verhaltensweisen auszulösen. Dieser Eventport ist weiterhin in ein sog. Fehlerszenario zu integrieren, das als Variante eines bestehenden Szenarios definiert ist. Bei der Simulation kann anschließend überprüft werden, ob das Szenario trotz des eingebrachten Fehlers bis zum Ende ablaufen kann und ob dabei alle modellierten Bedingungen und Monitore eingehalten werden. Wenn dies nicht der Fall ist, muss eine entsprechende Fehlerbehandlung in das Systemverhalten integriert werden, die bspw. das Wiederholen eines Teilablaufs anstößt oder das System stoppt und eine entsprechende Fehlermeldung generiert. Da trotz der Fehlerbehandlung in der Regel die ursprünglich modellierten Szenarien mit ihren Bedingungen nicht erfüllt werden können, ist in einem letzten Teilschritt eine Anpassung der Spezifikation erforderlich. Dabei sind insbesondere Szenarien, Eigenschaften und Monitore zu adaptieren, um das erweiterte Systemverhalten zur Behandlung von Komponentenfehlern zu berücksichtigen. Das eben eingeführte Vorgehen mit den Teilschritten zur Fehlermodellierung, -behandlung und anschließenden Spezifikationsanpassung kann für eine beliebige Anzahl von möglichen Komponentenfehlern wiederholt werden, um die Robustheit eines Systems auf ein gewünschtes Niveau anzuheben. Sobald eine ausreichende Anzahl von Komponentenfehlern im Hinblick auf einen konkreten Anwendungsfall modelliert und behandelt wurde, liegt ein vollständiges mechanisches Konzept des zu entwickelnden Systems vor, das anschließend in der Phase der Verfeinerung weiterführend umgesetzt werden kann.

A2 Metamodell der Modellierungstechnik

In Ergänzung zur Beschreibung der einzelnen HMI-spezifischen Elemente der Modellierungstechnik in Abschnitt 5.2.1 wird im Folgenden das zugehörige Metamodell dieser Elemente beschrieben, das für eine Umsetzung der Modellierungstechnik in einem Softwarewerkzeug benötigt wird. Dabei werden im Wesentlichen die Klassendiagramme der einzelnen Modellierungselemente auf Basis der UML in Anlehnung an die Darstellung aus Anhang A1 präsentiert. Weiterhin erfolgt eine textuelle Erläuterung der einzelnen Modellierungselemente, um den Zusammenhang zwischen den jeweiligen Beschreibungen in Abschnitt 5.2.1 und der zugehörigen Umsetzung im Metamodell zu erläutern. Wichtig ist dabei zu erwähnen, dass die Modellierungselemente „HMI-Komponenten“ und „Verhalten“ nicht explizit aufgeführt werden. Im Hinblick auf HMI-Komponenten liegt dies daran, dass sie aus Sicht des Metamodells mit allgemeinen Komponenten identisch sind und sich lediglich über ihre Bestandteile (Ports, Verhalten, etc.) auf eine semantische Weise als HMI-Komponenten definieren. Hinsichtlich des Modellierungselements „Verhalten“ ist dies damit zu begründen, dass das zugehörige Metamodell direkt aus dem ursprünglichen IMoMeSA-Ansatz übernommen wurde, das bereits in Anhang A1 dargestellt wurde.

Das erste Element zur Modellierung von Mensch-Maschine-Schnittstellen stellen die *Benutzerrollen* dar, die sich inhaltlich aus einem Namen der Rolle sowie einer textuellen Beschreibung des jeweiligen Rollenverständnisses zusammensetzen (vgl. Abbildung A-11). Darüber hinaus können Rollen in einer Eltern-Kind-Beziehung zueinanderstehen, was im Metamodell über eine entsprechende Assoziation zum Ausdruck gebracht wird. Generell sind User Stories beliebigen Komponenten zugeordnet, um auch den Fall zu berücksichtigen, dass einzelne Benutzer nur für Teile eines mechatronischen Systems verantwortlich sind.

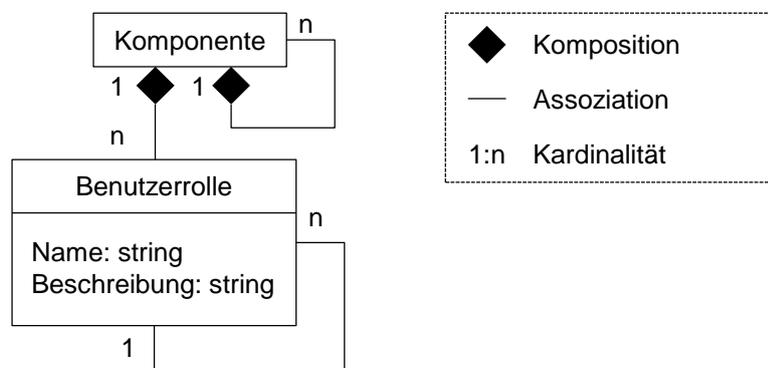


Abbildung A-11: Metamodell des Modellierungselements „Benutzerrolle“

Für die einzelnen Benutzerrollen können die Anforderungen im Umgang mit einer Mensch-Maschine-Schnittstelle anhand des Elements *User Story* modelliert werden. Der standardisierte Aufbau einer User Story über eine Benutzerrolle, ein Ziel und einen Nutzen ist im Metamodell über zwei entsprechende Attribute sowie eine Assoziation abgebildet (vgl. Abbildung A-12). Analog zu den allgemeinen Anforderungen sind User Stories den Komponenten zugeordnet, wodurch sie sich entlang der Komponentenhierarchie entsprechend verteilen können.

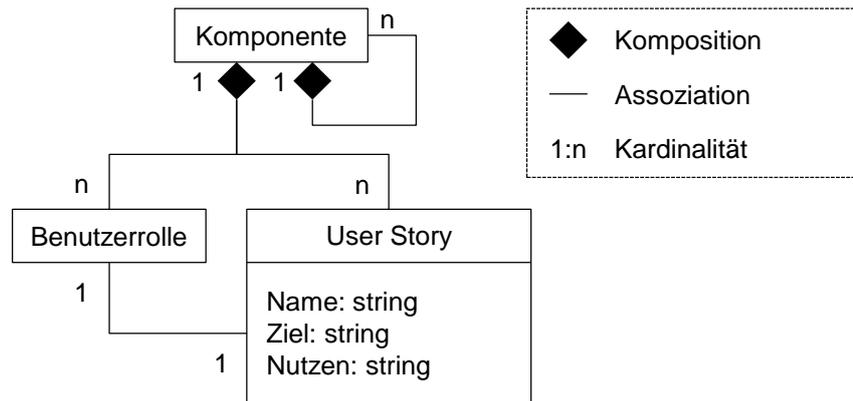


Abbildung A-12: Metamodell des Modellierungselements „User Story“

Zur Formalisierung von User Stories kommen des Weiteren sog. *HMI-Ports* zum Einsatz. Diese sind als Spezialform der allgemeinen Ports definiert, wofür das Metamodell von Ports lediglich um das Attribut „HMI-Flag“ zu ergänzen ist, um einen Port als HMI-Port zu kennzeichnen (vgl. Abbildung A-13). Sobald dieses Flag gesetzt ist, kann zudem eine Benutzerrolle verknüpft werden, um grundsätzliche Verantwortlichkeiten bereits auf Modellebene festzulegen. Darüber hinaus kann das Metamodell von Ports mit seiner Unterstruktur über Material-, Energie-, Daten- und Eventports unverändert erhalten bleiben, weswegen diese Struktur in Abbildung A-13 nicht mehr explizit dargestellt wird.

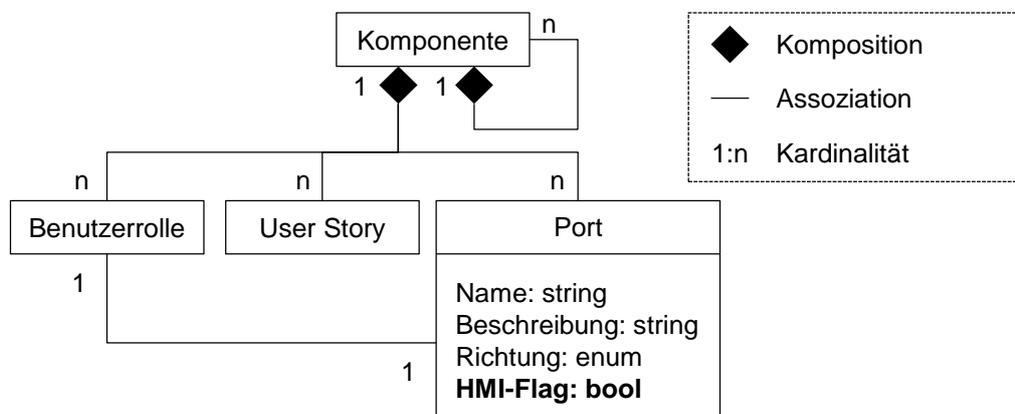


Abbildung A-13: Metamodell des Modellierungselements „HMI-Port“

Um die Interaktion eines Benutzers mit einer Komponente anhand der eben eingeführten HMI-Ports festzulegen, kann das Modellierungselement *Benutzerdialog* genutzt werden. Dieses Modellierungselement ist als Spezialform von Szenarien definiert und erweitert deren Funktionalität dahingehend, dass in einem Dialog mehrere Benutzer als Entitäten auftreten können. Dazu sieht das zugehörige Metamodell (vgl. Abbildung A-14) zunächst die Spezifikation aller an einem konkreten Dialog beteiligten Benutzer vor, die mit den Benutzerrollen verknüpft werden. Für jeden Benutzer können dann jeweils eigene Schritte, Transitionen, etc. definiert werden, um deren jeweilige Interaktion mit dem System über HMI-Ports zu spezifizieren.

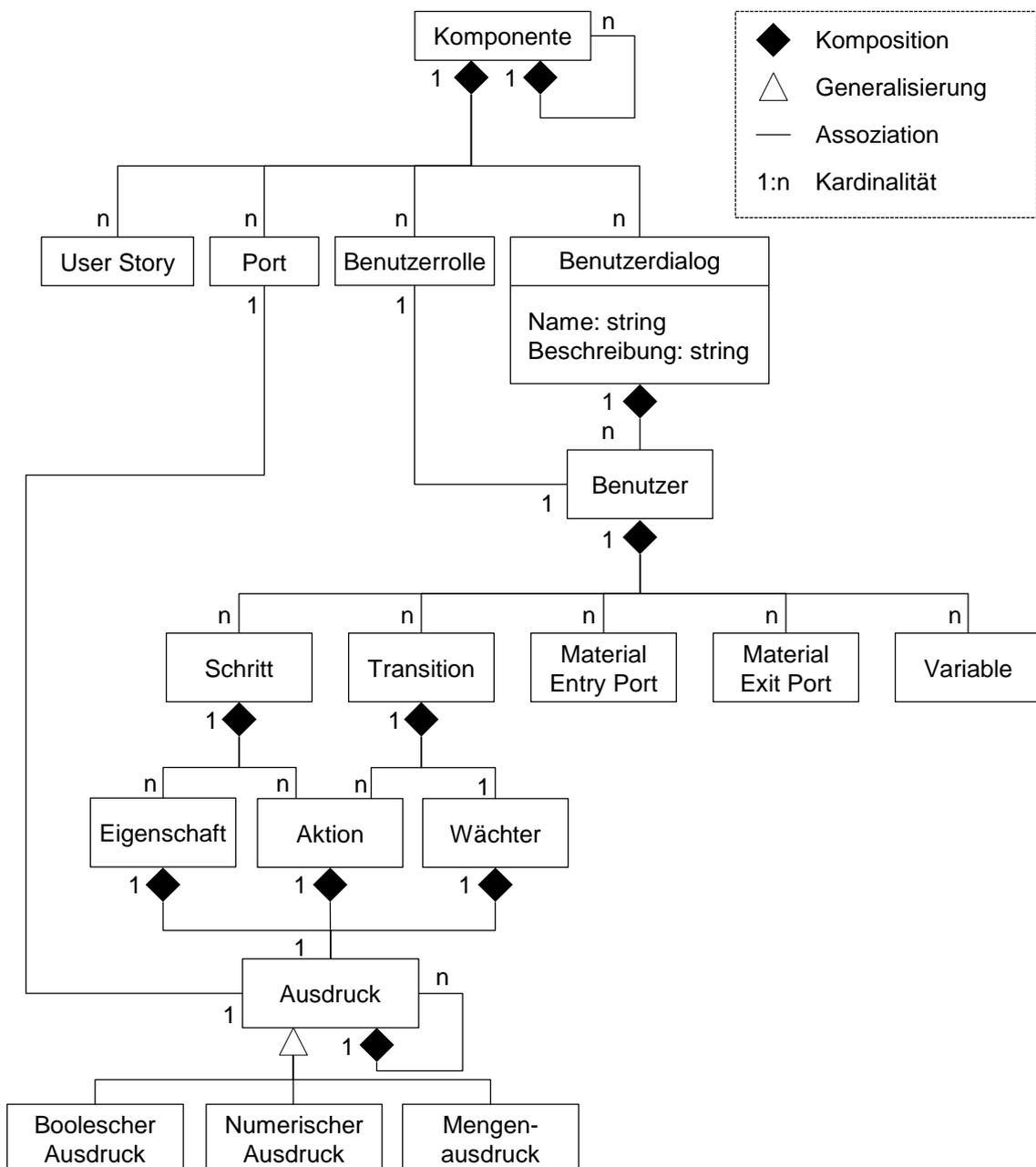


Abbildung A-14: Metamodell des Modellierungselements „Benutzerdialog“

Eine letzte Erweiterung der IMoMeSA-Modellierungstechnik zur Abbildung von HMIs bilden die *Interaktionselemente*. Deren generelle Unterteilung in Objekt- und Screen-Repräsentationen findet sich auch im Metamodell wieder (vgl. Abbildung A-15). Objekt-Repräsentationen können dabei wie Materialports und Bauteile über Volumina definiert werden. Demgegenüber setzen sich Screen-Repräsentationen aus Controls zusammen, über deren Attribute die Position auf dem Screen, die Größe und die Farbe festgelegt wird. Ferner lassen sich Controls in Bedien- und Anzeige-, Container- sowie Navigationselemente weiterführend unterteilen. Die Bedien- und Anzeigeelemente sind dabei mit entsprechenden HMI-Ports verknüpft und setzen sich aus den in Abbildung A-15 gezeigten Bausteinen zusammen. Mit Containerelementen werden hingegen einzelne Bereiche auf einem Screen definiert. Navigationselemente dienen schließlich dazu, zwischen einzelnen Screens zu wechseln, wofür eine entsprechende Assoziation im Metamodell vorgesehen ist.

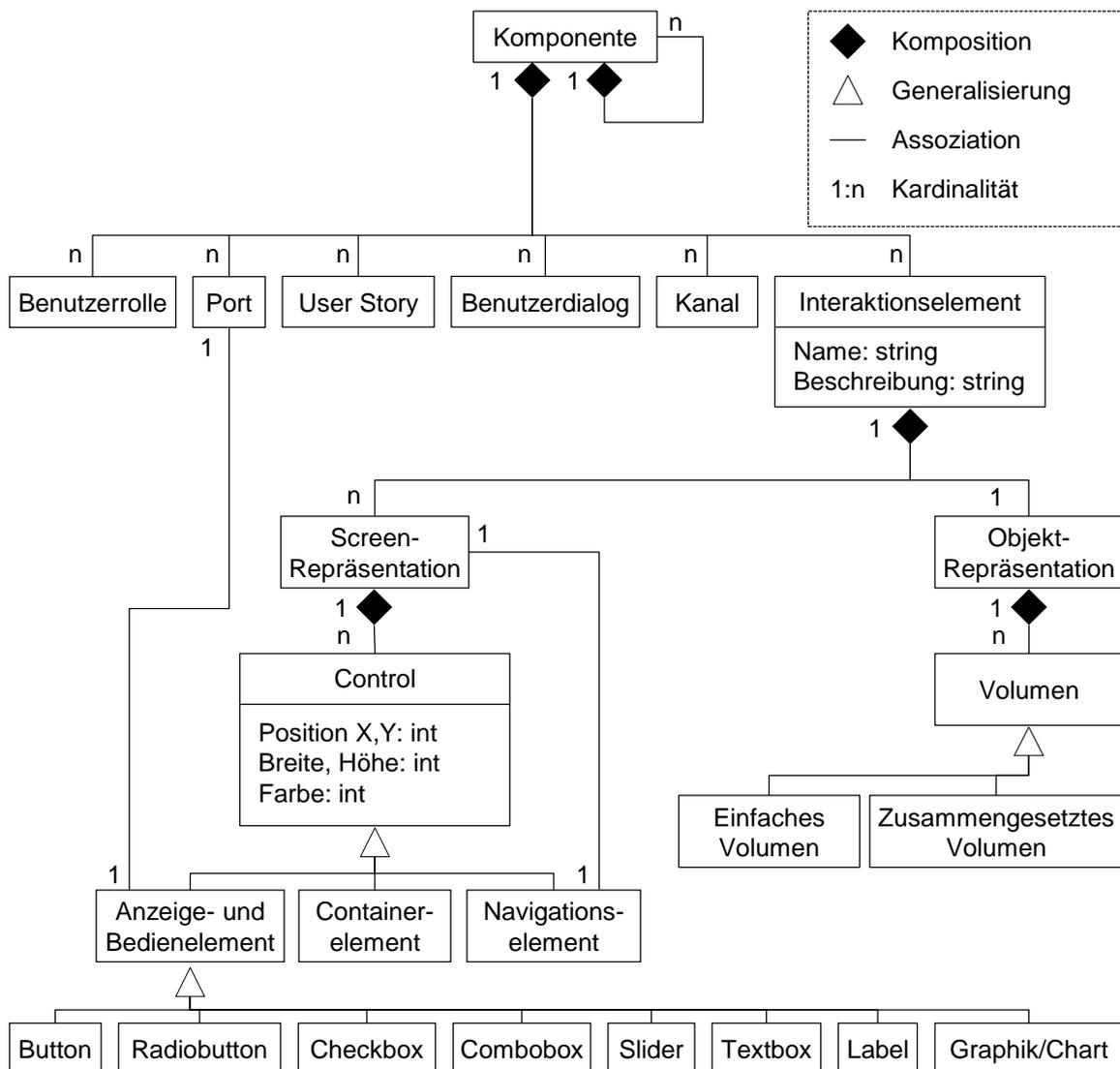


Abbildung A-15: Metamodell des Modellierungselements „Interaktionselement“

A3 Auswahlfaktoren und Klassifikationskriterien

In den Abbildungen A-16 bis A-21 sind die Auswahlfaktoren und Klassifikationskriterien mit ihren jeweiligen Ausprägungen dargestellt, die in der erarbeiteten Einflussmatrix verwendet werden. Dabei erfolgt eine Gliederung anhand der übergeordneten Stakeholder für die Auswahlfaktoren bzw. der einzelnen Technologieklassen für die Klassifikationskriterien.

Entwicklung		Vertrieb		
Größe des Entwicklungsteams	1 Person	Distributionspolitik	HMI als Bestandteil der Maschine bzw. Anlage	
	2 - 5 Personen		HMI in unterschiedlichen Ausbaustufen erwerbbar	
	> 5 Personen		Verkauf von Updates in der Betriebsphase	
HMI-Design	Durch Programmierer	Fertigungstyp	Einzelfertigung	
	Durch Usability Experten		Serienfertigung	
Vorhandenes Programmier-Know-how	Hochsprachen-Kenntnisse		Massenfertigung	Interaktions-technologien im Produktportfolio
	SPS-Kenntnisse	Gemischt		
	Konfigurations- bzw. Baukastenkenntnisse	Vorwiegend einfache Taster und Anzeigen		
Rechtmanagement	Vernachlässigbar	Ausschließlich einfache Taster und Anzeigen	Vorwiegend graphische Benutzerschnittstellen	
	Einfaches Rechte-management ausreichend	Ausschließlich graphische Benutzerschnittstellen	Ausschließlich graphische Benutzerschnittstellen	
	Detaillierte Eingriffsbeschränkungen erforderlich	Vorwiegend Sprachsteuerungen	Vorwiegend Sprachsteuerungen	
Notwendigkeit von Fernwartungen	Fernwartung unbedingt erforderlich	Ausschließlich Sprachsteuerungen	Ausschließlich Sprachsteuerungen	
	Fernwartung optional	Vorwiegend Körper-gesteuerungen	Vorwiegend Körper-gesteuerungen	
	Servicetechniker vor Ort, keine Fernwartung nötig	Ausschließlich Körper-gesteuerungen	Ausschließlich Körper-gesteuerungen	
Innovations-bereitschaft	Wahrung des Status Quo	Vorwiegend Augmented Reality	Ausschließlich Augmented Reality	
	Inkrementelle Innovation	Ausschließlich Augmented Reality	Konfigurations-möglichen für Kunden	
	Disruptive Innovation	Erforderlich		
Bedeutung der optischen Erscheinung	Hoch	Gewünscht		
Kosten für Entwicklung	Gering	Nicht gewünscht	Einkauf	
	Kosten für Entwicklung	Minimierung erforderlich		Verfügbarkeit der Technologie
Marktumfeld im Kerngeschäft	Investitionen möglich	Monopolstellung		
	Geringe Konkurrenz		Geringe Konkurrenz	Mittelfristig ausreichend
	Starke Konkurrenz	Starke Konkurrenz	Langfristig gewünscht	
Preisstrategie	Kosten gering halten	Preisstrategie	Langfristig gefordert	
	Keine Vorgaben		Gefordertes Erscheinungsdatum	„Freitexteingabe“
	Hochpreisiges Segment anstreben		Geforderter Gewähr-leistungszeitraum	„Freitexteingabe“
		Herstellervorgaben	„Freitexteingabe“	

Abbildung A-16: Auswahlfaktoren der herstellerseitigen Stakeholder

<i>Benutzer</i>	
Bewegungsradius	< 1m
	1 - 10m
	10 - 100m
	> 100m
Bildungsstand	Analphabeten
	Geringe Bildung
	Mittlere Bildung
	Hohe Bildung
Mehrmaschinenbedienung	Ja, mehr als 5 Maschinen
	Ja, bis zu 5 Maschinen
	nein
Anzahl unterschiedlicher Benutzergruppen	> 3
	2 - 3
	1
Anzahl unterschiedlicher Benutzer	> 5
	3 - 5
	1 - 2
Benutzerfluktuation	Häufig neues Personal
	Selten neues Personal
Verfügbarkeit der Hände	Beide Hände belegt
	Eine Hand belegt
	Beide Hände frei
Persönliche Schutzausrüstung	Handschuhe
	Helm
	Schutzbrille
	Gehörschutz

<i>Produktionsleiter</i>	
Geforderte Datensicherheit	Gering
	Mittel
	Hoch
Örtliche Verfügbarkeit der Daten	Nur an der Maschine
	In der Nähe der Maschine
	Im gesamten Fabrikumfeld
Gewünschte Betriebssysteme	Windows-basierte Systeme
	Apple-basierte Systeme
	Linux-basierte Systeme
Minimale Lebensdauer	„Freitexteingabe“
Geforderte Kommunikationstechnologien	TCP/IP
	Profibus
	Profinet
	Modbus
	CANOpen
	OPC UA
	WLAN
	Bluetooth
Bediengeschwindigkeit	Schnelle Bedienung erforderlich
	Keine Relevanz
Wirtschaftliche Folgen von Fehlbedienungen	Geringe Auswirkungen
	Maschinenstillstand
	Produktionsstillstand
Diebstahlgefahr	Geringe Gefahr
	Mittlere Gefahr
	Hohe Gefahr
Bedienung mit Stiften oder Griffeln	Gewünscht bzw. gefordert
	Möglich
	Nicht möglich
Interaktionstechnologien im Maschinenpark	Gemischt
	Vorwiegend einfache Taster und Anzeigen
	Ausschließlich einfache Taster und Anzeigen
	Vorwiegend graphische Benutzerschnittstellen
	Ausschließlich graphische Benutzerschnittstellen
	Vorwiegend Sprachsteuerungen
	Ausschließlich Sprachsteuerungen
	Vorwiegend Körpergestensteuerungen
	Ausschließlich Körpergestensteuerungen
	Vorwiegend Augmented Reality
Ausschließlich Augmented Reality	

<i>Management</i>	
Verfügbarkeit der Technologie	Kurzfristig ausreichend
	Mittelfristig ausreichend
	Langfristig gewünscht
	Langfristig gefordert
Kosten der Technologie	Günstiges Preissegment
	Mittleres Preissegment
	Hohes Preissegment

<i>Einkauf</i>	
Verfügbarkeit der Technologie	Kurzfristig ausreichend
	Mittelfristig ausreichend
	Langfristig gewünscht
	Langfristig gefordert
Kosten der Technologie	Günstiges Preissegment
	Mittleres Preissegment
	Hohes Preissegment
Herstellervorgaben	„Freitexteingabe“

Abbildung A-17: Auswahlfaktoren der kundenseitigen Stakeholder

Normen/Gesetze		Maschine/Anlage	
Einzuhaltende Zertifizierungen	CE	Interaktionsart	Bedienen
	GL		Beobachten
	ABS		Bedienen und Beobachten
	BV	Interaktionshäufigkeit	< 10 p. Tag
	DNV		10 - 50 p. Tag
	UL		> 50 p. Tag
	CSA	Anzahl unterschiedlicher Bedienaufgaben	< 10
	Ulus		10 - 30
	C-TICK		> 30
	NEMA		
Stoßfestigkeitsklasse	IK00	Arten von Eingaben	Binäre Einstellungen
	IK01		Werteinstellungen
	...		Bewegungssteuerung
	IK10		Parametereingaben
	Alphanumerische Eingaben (einzelne Wörter)		
Explosionsschutzbestimmungen	ATEX Zone 22		Alphanumerische Eingaben (ganze Sätze)
	ATEX Zone 21	Arten von Ausgaben	Fehler, Warnungen und Informationen
	ATEX Zone 20		Maschinenstatus
	ATEX Zone 2		Maschinenparameter
	ATEX Zone 1		Grafiken und Animationen
ATEX Zone 0	Komplexität der Bedienaufgaben	Schwierige Aufgaben	
		Mittlere Aufgaben	
		Einfache Aufgaben	
Stromversorgung	Gleichstrom (DC)	Systemgröße	Anlage
	Wechselstrom (AC)		Produktionszelle
Lautstärke im Produktionsumfeld	Laute Geräusche		
	Normale Geräusche	Verfügbare Platz	„Freitexteingabe“
	Geringe Geräusche		
Spanaufkommen	Stark	Bedienung in Maschinennähe	Ja, erforderlich
	Gering		Nein, auch aus der Ferne möglich
	Keine Späne	Bedienung ohne Sichtkontakt zum HMI	Ja, erforderlich
Schmutzaufkommen im Produktionsumfeld	Stark		Nein, Sichtkontakt immer möglich
	Gering		
Feuchtigkeit im Produktionsumfeld	Reinraumbedingungen	Anzahl unterschiedlicher Bedienstellen	> 3
	Untertauchen möglich		2 - 3
	Strahlwasser möglich		1
	Spritzwasser möglich	Anordnung und Erreichbarkeit der Bedienstellen	Zentral und einfach erreichbar
Kein Wasserkontakt	Zentral, aber schwierig erreichbar		
Elektromagnetische Verträglichkeit	notwendig		Dezentral, aber einfach erreichbar
	Nicht notwendig		Dezentral und schwierig erreichbar
Maximale Temperatur	„Freitexteingabe“	Gefährdungspotenzial durch Fehlbedienung	Groß
Minimale Temperatur	„Freitexteingabe“		Mittel
Vorherrschende Lichtverhältnisse	Hohe Sonneneinstrahlung		Gering
	Geringe Sonneneinstrahlung		
	Keine Sonneneinstrahlung		

Abbildung A-18: Auswahlfaktoren der Rahmenbedingungen

<i>Interaktionstechnologien</i>	
Erscheinungsdatum	‚Freitexteingabe‘
Hersteller	‚Freitexteingabe‘
Unterstützte Betriebssysteme	Windows
	Apple
	Linux
Verfügbarkeit	Langfristig sichergestellt
	Mittlere Zeitspanne
	Kurze Zeitspanne
Komplexität	Eignung für Analphabeten
	Gering
	Mittel
Preis	Günstiges Preissegment
	Mittleres Preissegment
	Hohes Preissegment

<i>Interaktionsgeräte</i>	
Informationsrichtung	Eingabe
	Ausgabe
	Ein- und Ausgabe
Spiegelungs-empfindlichkeit	Hoch
	Bedingt
	Keine
Feuchtigkeitsschutz	Gegen Untertauchen
	Gegen Strahlwasser
	Gegen Spritzwasser
	Kein Schutz
Schmutzempfindlichkeit	Hoch
	Bedingt
	Keine
Schallempfindlichkeit	Hoch
	Bedingt
	Keine

<i>Hardware</i>	
Portabilität	Mobil
	Stationär
Unterstützung mehrerer Applikationen	Gegeben
	Eingeschränkt
	Nicht möglich
Stromversorgung	Batterie
	Netzkabel
	USB
Hitzebeständigkeit	‚Freitexteingabe‘
Kältebeständigkeit	‚Freitexteingabe‘

Vorhandene Zertifizierungen	CE
	GL
	ABS
	BV
	DNV
	UL
	CSA
	Ulus
Stoßfestigkeitsklasse	C-TICK
	NEMA
	IK00
	IK01
Gewährleisteter Explosionsschutz	...
	IK10
	ATEX Zone 22
	ATEX Zone 21
	ATEX Zone 20
	ATEX Zone 2
	ATEX Zone 1
	ATEX Zone 0
Befestigung am Körper	Auf den Augen
	Auf dem Kopf
	Am Körper
	Externe Trageeinrichtung notwendig
Platzbedarf	Nicht vorgesehen
	‚Freitexteingabe‘
	‚Freitexteingabe‘
Lebensdauer	‚Freitexteingabe‘
	‚Freitexteingabe‘
	‚Freitexteingabe‘
Unterstützte Kommunikationstechnologien	TCP/IP
	Profibus
	Profinet
	Modbus
	CANOpen
	OPC UA
	WLAN
	Bluetooth
	USB
	Anbindung externer Systeme
Nicht möglich	
Stromversorgungsart	Wechselstrom (AC)
	Gleichstrom (DC)
Elektromagnetische Verträglichkeit	kritisch
	unkritisch
Latenzzeit	Keine Echtzeit
	Weiche Echtzeit
	Harte Echtzeit

Abbildung A-19: Klassifikationskriterien (1/3)

<i>Eingabegeräte</i>	
Eingabemodalität	Visuell
	Akustisch
	Haptisch
Anzahl möglicher Operationen	Stark eingeschränkt
	Eingeschränkt
	Unbegrenzt
Kratzempfindlichkeit	Hoch
	Bedingt
	Keine
Benötigte Hände	Keine
	Eine Hand
	Beide Hände
Sicherheit gegenüber Fehlbedienung	Gering
	Mittel
	Hoch

<i>Visuelle Eingabegeräte</i>	
Notwendige Entfernung	< 1m
	1 - 10m
	> 10m
Erkennung	Statisch (Erkennung von Positionen)
	Dynamisch (Erkennung von Bewegungen)
Erfassbare Körperbereiche	Gesamter Körper
	Hände
	Füße
	Kopf
	Augen

<i>Akustische Eingabegeräte</i>	
Typ	Spracheingabe
	Sonstige Signale
Durchgängigkeit	Kontinuierlich (Satzweise)
	Diskret (Wort für Wort)
Abhängigkeit vom Sprecher	Ja
	Nein

<i>Ausgabegeräte</i>	
Ausgabemodalität	Visuell
	Akustisch
	Haptisch
Anzahl anzuzeigender Informationen	Stark eingeschränkt
	Eingeschränkt
	Unbegrenzt

<i>Ein- und Ausgabegeräte</i>	
Technologisches Prinzip	Kapazitiv
	Resistiv
	Induktiv
	Optisch
Bedienbarkeit mit Handschuhen	Möglich
	Nicht möglich
Bedienbarkeit mit Griffeln oder Stiften	Notwendig
	Optional
	Nicht möglich
Haptisches Feedback	Ja
	Nein
Touch-Eigenschaften	Single-Touch
	Dual-Touch
	Multi-Touch

<i>Haptische Eingabegeräte</i>	
Bedienbarkeit mit Handschuhen	Möglich
	Nicht möglich
Feedback vorhanden	Ja
	Nein
Bauformen	Tastatur
	Befehlsgeber
	Positioniergerät

<i>Visuelle Ausgabegeräte</i>	
Bauformen	Bildschirm
	Signalleuchte
	Augmented Reality
	2D-Projektion
Auflösung	3D-Projektion
	Hoch
	Mittel
	Gering

Abbildung A-20: Klassifikationskriterien (2/3)

<i>Akustische Ausgabegeräte</i>	
Bauformen	Lautsprecher
	Kopfhörer
Signallautstärke	Leise
	Mittel
	Laut
Typ	Sprachausgabe
	Sonstige Signale

<i>Tastaturgeräte</i>	
Vorhandene Tasten	Buchstaben (ausgewählt)
	Buchstaben (vollständig)
	Ziffern
Tastengröße	Klein
	Mittel
	Groß

<i>Haptische Ausgabegeräte</i>	
Typ	Vibration
	Force Feedback
	Touch Feedback
Beeinflusster Bereich	Hand
	Arm
	Gesicht
	Torso
	Beliebig

<i>Positioniergeräte</i>	
Positionierart	Absolut
	Relativ
Bauformen	Computermaus
	Touchpad
	Trackball
	Joystick
	Tablett
	3D-Maus
Freiheitsgrade	Bat
	Eindimensional
	Zweidimensional
	Dreidimensional
	Mehr-Dimensional (>3)

<i>Befehlsgeber</i>	
Wertebereich	Diskret (<5 Werte)
	Diskret (>5 Werte)
	Kontinuierlich
Bedienung mit	Hand
	Fuß
	Anderer Körperteil

<i>Software</i>	
Applikationsart	Eigenständiges Programm
	Web HMI
	Industrie-App
Branchenfokus	Stark eingeschränkt
	Eingeschränkt
Berechnungszeit	Hoch
	Mittel
	Gering
Erforderliches Programmier-Know-how	Hochsprache
	SPS-Kenntnisse
	Baukastensysteme / Konfigurationen
Verfügbare Funktionalitäten	Benutzer- und Rechtemanagement
	Konfiguration im Betrieb
	Remote-Zugriff
	Datenverschlüsselung
	Einspielen von Updates
	Virtuelle Tastaturen
	Navigation
	Nutzerdialoge
Unterstützte Entwicklungsparadigmen	Verteilte Entwicklung
	Trennung von Funktion und Design
	Frameworks und Bibliotheken

Abbildung A-21: Klassifikationskriterien (3/3)

A4 Verzeichnis verwendeter Software

- Citavi® 4
- Eclipse® Mars Release 4.5.0
- Microsoft® Excel 2010
- Microsoft® Power Point 2010
- Microsoft® Visual Studio Express 2015
- Microsoft® Word 2010

A5 Verzeichnis betreuter Studienarbeiten

Der Autor der vorliegenden Dissertation hat in den Jahren von 2012 bis 2017 unter wesentlicher wissenschaftlicher, fachlicher und inhaltlicher Anleitung die im Folgenden aufgeführten studentischen Arbeiten am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) der Technischen Universität München (TUM) betreut. Die entstandenen Ergebnisse sind teilweise in das vorliegende Dokument eingeflossen. Der Autor dankt allen Studierenden für ihre Unterstützung bei dieser wissenschaftlichen Arbeit.

Marvin Becker: Entwicklung einer Methodik für die formale Überführung eines Funktionsmodells in die mechanische CAD-Konstruktion. Bachelorarbeit. Bearbeitungszeitraum: 01.05.2013 - 31.10.2013

Josef Bernöcker: Evaluierung einer Funktionsmodellierungstechnik für mechatronische Entwicklungsprozesse anhand der Entwicklung eines falt- und klebmoduls einer Verpackungsanlage. Semesterarbeit. Bearbeitungszeitraum: 01.04.2013 - 18.10.2013

Johannes Deist: Entwicklungsmethode zur zielgerichteten Analyse und Optimierung von bestehenden mechatronischen Systemen. Semesterarbeit. Bearbeitungszeitraum: 01.05.2014 - 31.10.2014

Julian Fichtner: Konzeption erweiterter mechatronischer Module zur Realisierung konfigurierbarer Benutzerschnittstellen. Semesterarbeit. Bearbeitungszeitraum: 16.03.2015 - 15.09.2015

Michael Gerigk: Integrierte Entwicklung von Benutzerschnittstellen im Kontext mechatronischer Systeme. Semesterarbeit. Bearbeitungszeitraum: 15.01.2015 - 14.07.2015

Annemarie Hofmann: Bewertungssystematik zur Unterstützung der Auswahl von Schnittstellentechnologien für industrielle Anwendungen. Semesterarbeit. Bearbeitungszeitraum: 15.09.2015 - 25.05.2016

Markus Hörmann: Betriebsstrategie für eine Ressourcenallokation in Cyber-Physischen-Produktionssystemen. Masterarbeit. Bearbeitungszeitraum: 17.08.2015 - 31.03.2016

Wolfgang Liertz: Konzept für ein konfigurierbares Cockpit eines Condition Monitoring-Systems zum Visualisieren von Zuständen in Werkzeugmaschinen. Diplomarbeit. Bearbeitungszeitraum: 01.02.2014 - 31.07.2014

Guillaume Mativon: Konzept zur teil-automatisierten Erstellung von 3D-Physiksimulationen von mechatronischen Systemen. Masterarbeit. Bearbeitungszeitraum: 15.11.2016 - 14.05.2017

Martina Müller: Entwicklung einer Methodik für ein integratives Anforderungsmanagement im Maschinen- und Anlagenbau. Semesterarbeit. Bearbeitungszeitraum: 15.11.2013 - 14.05.2014

Florian Pichler: Entwicklung einer automatisierten Testumgebung für SPS-basierte SW-Bibliotheken. Masterarbeit. Bearbeitungszeitraum: 15.04.2014 - 14.10.2014

Georg Räßle: Methodik zur Entwicklung von Bedienoberflächen begleitend zum mechatronischen Anlagenentstehungsprozess. Semesterarbeit. Bearbeitungszeitraum: 01.05.2014 - 31.10.2014

Hendrik Weber: Informationstechnische Integration mechatronischer Anlagenmodule mit individuellen Steuerungskonzepten. Semesterarbeit. Bearbeitungszeitraum: 01.10.2014 - 31.03.2015

Josef Xu: Entwicklung einer Methode zur Auswahl von Bedientechnologien für produktionstechnische Anwendungsfälle. Bachelorarbeit. Bearbeitungszeitraum: 15.04.2014 - 14.10.2014

Patrick Zimmermann: Vorgehensmodell zur Bestimmung einer I4.0-Vernetzungsarchitektur im Produktionsumfeld. Masterarbeit. Bearbeitungszeitraum: 16.05.2017 - 15.11.2017