



FAU Studien aus dem Maschinenbau 396

Thomas Johann Luft

Komplexitätsmanagement in der Produktentwicklung

Holistische Modellierung, Analyse,
Visualisierung und Bewertung komplexer Systeme

Thomas Johann Luft

Komplexitätsmanagement in der Produktentwicklung - Holistische
Modellierung, Analyse, Visualisierung und Bewertung komplexer
Systeme

FAU Studien aus dem Maschinenbau

Band 396

Herausgeber der Reihe:

Prof. Dr.-Ing. Jörg Franke
Prof. Dr.-Ing. Nico Hanenkamp
Prof. Dr.-Ing. habil. Tino Hausotte
Prof. Dr.-Ing. habil. Marion Merklein
Prof. Dr.-Ing. Michael Schmidt
Prof. Dr.-Ing. Sandro Wartzack

Thomas Johann Luft

Komplexitätsmanagement in der Produktentwicklung – Holistische Modellierung, Analyse, Visualisierung und Bewertung komplexer Systeme

Dissertation aus dem Lehrstuhl für Konstruktionstechnik (KTmfk)
Prof. Dr.-Ing. Sandro Wartzack

Erlangen
FAU University Press
2022

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Bitte zitieren als

Luft, Thomas Johann. 2022. *Komplexitätsmanagement in der Produktentwicklung - Holistische Modellierung, Analyse, Visualisierung und Bewertung komplexer Systeme*. FAU Studien aus dem Maschinenbau Band 396. Erlangen: FAU University Press. DOI: 10.25593/978-3-96147-541-4.

Das Werk, einschließlich seiner Teile, ist urheberrechtlich geschützt.
Die Rechte an allen Inhalten liegen bei ihren jeweiligen Autoren.
Sie sind nutzbar unter der Creative-Commons-Lizenz BY-NC.

Der vollständige Inhalt des Buchs ist als PDF über den OPUS-Server der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:
<https://opus4.kobv.de/opus4-fau/home>

Verlag und Auslieferung:
FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck: docupoint GmbH

ISBN: 978-3-96147-540-7 (Druckausgabe)
eISBN: 978-3-96147-541-4 (Online-Ausgabe)
ISSN: 2625-9974
DOI: 10.25593/978-3-96147-541-4

**Komplexitätsmanagement in der
Produktentwicklung**
– Holistische Modellierung, Analyse, Visualisierung
und Bewertung komplexer Systeme

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg

zur
Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von

Thomas Johann Luft, Dipl.-Wirtsch.-Ing.

aus Bamberg

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen

Prüfung: 13.09.2021

Gutachter: Prof. Dr.-Ing. Sandro Wartzack
Prof. Dr.-Ing. Kristin Paetzold, TU Dresden

Vorwort

„Der Fortschritt lebt vom Austausch des Wissens.“

ALBERT EINSTEIN (1879 – 1955)

Die vorliegende Arbeit entstand größtenteils während meiner Tätigkeit als Akademischer Rat am Lehrstuhl für Konstruktionstechnik (KTmfk) der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). Zuerst möchte ich mich an dieser Stelle bei allen Kollegen, Projektpartnern, Studierenden, Freunden sowie meiner Familie sehr herzlich bedanken, die auf vielfältigste Art und Weise zum Gelingen dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. Sandro Wartzack für die Betreuung meines Promotionsvorhabens. Sein entgegengebrachtes Vertrauen sowie seine fordernde als auch fördernde Betreuung hat maßgeblich zum erfolgreichen Gelingen der vorliegenden wissenschaftlichen Arbeit beigetragen. Dafür besten Dank, Sandro! Top!

Frau Prof. Dr.-Ing. Kristin Paetzold, Inhaberin der Professur für Virtuelle Produktentwicklung an der TU Dresden, möchte ich für das Interesse an meiner Arbeit und die Übernahme des Zweitgutachtens herzlich danken. Zudem danke ich Herrn Prof. Dr. Lutz Schröder für sein die Bereitschaft, die Funktion des fachfremden Prüfers zu übernehmen. Herrn Prof. Dr.-Ing. Nico Hanenkamp möchte ich für die Übernahme des Prüfungsvorsitzes bei der Promotionsprüfung meinen Dank aussprechen.

Meinen ehemaligen Kolleginnen und Kollegen des KTmfk danke ich für die gute gemeinsame Zeit, die hilfreiche Unterstützung, das konstruktive Feedback sowie die fachlichen und persönlichen Gespräche. Alle KTmfk'ler haben auf mannigfache Art und Weise zum Gelingen dieser Dissertation beigetragen. Besonders hervorheben möchte ich meinen Kollegen Christof Küstner für die gute Zusammenarbeit und insb. dafür, dass er mir Python ans Herz gelegt hat. Zudem möchte ich mich bei Evelyn Weyrich für die organisatorische und bei Thomas Niering für die informationstechnische Unterstützung bedanken.

Danken möchte ich auch Wissenschaftlern anderer Universitäten für den fachlichen und freundschaftlichen Austausch. Stellvertretend möchte ich Daniel Roth und Alexander Crostack von der Universität Stuttgart sowie Markus Mörtl und Christian Schmied von der TU München nennen.

Weiterer Dank gilt auch der Deutschen Forschungsgemeinschaft (DFG) für die finanzielle Förderung insb. der zwei Forschungsprojekte zum produktorientierten Prozessmanagement. Zudem möchte ich mich beim Verein Deutscher Ingenieure (VDI) für die gute Zusammenarbeit im Rahmen der Erstellung der VDI-Richtlinie „Wissensmanagement im Ingenieurwesen – Wissensbasierte Konstruktion (KBE)“ bedanken.

Wesentliche und wertvolle Unterstützung bei meinen Forschungsarbeiten haben zahlreiche Studierende geleistet, deren studentische Arbeiten ich betreuen durfte. Dafür möchte ich mich ganz herzlich bedanken, vor allem bei M. Beck, J. Bochmann, M. Diekkrüger, D. Feldmann, M. Hittinger, M. Hofmann, M. Kahl sowie S. Schlenz.

Außerdem möchte ich mich auch bei meinem aktuellen Arbeitgeber, der Voith Group, für die zeitlichen Freiräume und das Verständnis während der nebenberuflichen Fertigstellung dieser Arbeit bedanken. Ich freue mich besonders, dass ich bereits einige Erkenntnisse aus meiner Lehrstuhlzeit und meinen Forschungsarbeiten im industriellen Umfeld anwenden und gewinnbringend nutzen konnte.

Abschließend möchte ich mich ganz herzlich bei allen meinen Familienmitgliedern für ihre fortdauernde Unterstützung und Förderung bedanken. Dies ist die wichtige Grundlage für meine bisherige akademische Laufbahn. Schließlich gebührt meiner Andrea sowohl herzlichster Dank als auch höchster Respekt für die wertvolle Unterstützung als auch den nötigen Rückhalt.

Burgebrach, im Februar 2022

Thomas J. Luft

An dieser Stelle möchte der Autor darauf hinweisen, dass in dieser Arbeit das generische Maskulinum (wie z. B. Ingenieur) aus Gründen der besseren Lesbarkeit und auch zur sprachlichen Vereinfachung stets verwendet wird. Selbstverständlich sind immer Personen **männlichen, weiblichen und diversen Geschlechts gleichermaßen** gemeint.

Inhaltsverzeichnis

Formelzeichen- und Abkürzungsverzeichnis	ix
1 Einleitung	1
1.1 Ausgangssituation und Rahmenbedingungen	1
1.2 Problemstellung und Herausforderungen	4
1.3 Zielsetzung und Aufbau der Arbeit	5
1.4 Rolle des Komplexitätsingenieurs	7
1.5 Beispielhaftes Anwendungsszenario	10
2 Grundlagen zur Modellierung komplexer Systeme	15
2.1 System und Systemverständnis	15
2.1.1 Systemtheorie und Systembegriff.....	16
2.1.2 Systemtechnik und Theorie technischer Systeme	18
2.2 Komplexität und Komplexitätsmanagement	21
2.2.1 Begriffserklärung von Komplexität	21
2.2.2 Grade und Treiber der Komplexität	24
2.2.3 Strategien des Komplexitätsmanagements	29
2.3 Modellverständnis und Modellierung.....	31
2.3.1 Modellbegriff und Modellarten	32
2.3.2 Modellraum und Navigationsmodell der Entwicklung.....	34
2.3.3 Modellierungsmethoden und -sprachen	37
3 Stand der Forschung und Technik	41
3.1 Grundlagen domänenspezifischer Modelle.....	41
3.1.1 Produktorientierte Modellierungsansätze	43
3.1.2 Prozessorientierte Modellierungsansätze	44
3.2 Grundlagen domänenübergreifender Modelle	45
3.2.1 Modellbasiertes Systems Engineering	46
3.2.2 Ontologiebasierte Modelle	51
3.2.3 Matrixbasierte Modelle	55
3.3 Ansätze für das matrixbasierte Komplexitätsmanagement in der Produktentwicklung	61
3.3.1 Matrixbasierte Modellierungsansätze.....	62
3.3.2 Graphbasierte Visualisierungsansätze	71
3.3.3 Mathematische Analyseansätze	79
3.3.4 Ansätze zur Softwareunterstützung	86

4	Handlungsbedarfe und Ziele.....	93
4.1	Diskussion des Stands der Forschung und Technik.....	93
4.1.1	Forschungsbedarf aus Sicht der Wissenschaft	96
4.1.2	Handlungsbedarf aus Sicht der Wirtschaft.....	101
4.2	Ziele und Anforderungen	104
4.2.1	Forschungsfragen und Ziele	105
4.2.2	Anforderungen an das zu entwickelnde Konzept	106
5	Konzept für ein Assistenzsystem zum Komplexitätsmanagement in der PE.....	111
5.1	Gesamtkonzept des Assistenzsystems	112
5.1.1	Vorgehen beim Management komplexer Systeme.....	112
5.1.2	Prinzipieller Ablauf der Modellierung und Analyse	115
5.1.3	IT-Architektur und Module des Assistenzsystems	118
5.2	Automatisierte Modellierung von DSMs.....	122
5.2.1	Produktmodellierung	123
5.2.2	Prozessmodellierung	132
5.3	Computerunterstützte Modellierung von DMMs	137
5.3.1	Grafische Modellierung.....	138
5.3.2	Domain Merger	142
5.4	Graphbasierte Visualisierung und Kommunikation	145
5.4.1	Visualisierungsmodul.....	146
5.4.2	Visualisierungsprogramme	151
5.5	Mathematische Analyse und Bewertung.....	153
5.5.1	Analysemodul	153
5.5.2	Requirements Analysis Module.....	156
5.5.3	Cost Analysis Module	160
5.6	Holistische Manipulation und Modifikation	165
5.6.1	Modifying Module	166
5.6.2	Functional Analysis Module	169
6	Praktische Anwendungsbeispiele	173
6.1	Matrixbasierte Produktmodellierung	174
6.1.1	Modellierung einer DSM für ein einfaches Einzelteil.....	175
6.1.2	Modellierung einer DSM für eine komplexe Baugruppe	177
6.2	Matrixbasierte Prozessmodellierung	179
6.2.1	Modellierung einer DSM für einen einfachen Prozess	179
6.2.2	Modellierung einer DSM für einen komplexen Prozess	180

6.3	Modellierung von DMMs	183
6.3.1	Grafische Modellierung einer DMM	183
6.3.2	Modellierung mit dem Domain Merger	185
6.4	Visualisierung von matrixbasierten Modellen	186
6.4.1	Graphbasiertes Visualisierungsmodul	186
6.4.2	Externes Visualisierungsprogramm Tulip	188
6.5	Analyse von matrixbasierten Modellen	190
6.5.1	Mathematisches Analysemodul	190
6.5.2	Requirements Analysis Module	192
6.5.3	Cost Analysis Module	194
7	Kritische Diskussion und Reflexion.....	199
7.1	Diskussion aus Sicht der Wissenschaft.....	199
7.2	Diskussion aus Sicht der Wirtschaft	202
8	Zusammenfassung und Ausblick.....	205
8.1	Zusammenfassung und Schlussfolgerungen	205
8.2	Ausblick und weiterführende Forschungsfragen	207
9	Summary and outlook.....	211
9.1	Summary and conclusions	211
9.2	Outlook and further research questions.....	213
	Anhang.....	217
A.1	Definition und Bestimmung von Begriffen	217
A.2	Grundlagen und Erläuterungen zu VUCA.....	218
A.3	Kompetenzen des Komplexitätsingenieurs	230
A.4	Rechenbeispiel zum Anwendungsszenario Rohr	233
A.5	Begriffe, Prinzipien der Systemtheorie und eine Klassifikationssystematik von Systemen	240
A.6	Bedeutung, Bestandteile und Klassifikation technischer Systeme	254
A.7	Grundlagen zum Komplexitätsverständnis	266
A.8	Arten, Treiber und Grade von Komplexität	275
A.9	Auswirkungen von Komplexität auf Kosten	297
A.10	Strategien und Ziele des Komplexitätsmanagements	304
A.11	Klassifikationen und Arten von Modellen	314
A.12	Grundlagen zum Modellverständnis und zur Modellierung	323
A.13	Modellierung und Modellierungsansätze.....	335
A.14	Modellierungsmethoden und -sprachen.....	342

A.15 Referenzmodelle und Metamodelle.....	346
A.16 Überblick über domänenspezifische Modelle	353
A.17 Grundlagen zu ontologiebasierten Modellen	371
A.18 Grundlagen zu matrixbasierten Modellen	382
A.19 Grundlagen zur graphbasierten Visualisierung	389
A.20 Grundlagen zur mathematischen Analyse	391
A.21 Vorgehensweise zur Erstellung, Visualisierung und Analyse von matrixbasierten Modellen	397
A.22 Ergänzungen zur Produktmodellierung	401
A.23 Ergänzungen zur Prozessmodellierung	406
A.24 Ergänzungen zur Visualisierung	415
A.25 Ergänzungen zur Analyse.....	419
A.26 Ergänzungen zum Assistenzsystem	429
Literaturverzeichnis	435
Verzeichnis promotionsbezogener, eigener Publikationen	498
Verzeichnis promotionsbezogener, studentischer Arbeiten	503

Formelzeichen- und Abkürzungsverzeichnis

Formelzeichen

<i>Symbol</i>	<i>Einheit</i>	<i>Beschreibung</i>
A	mm^2	Querschnittsfläche
BS	Nmm^2	Biegesteifigkeit (EI)
DS	N	Dehnsteifigkeit (EA)
E	N/mm^2	Elastizitätsmodul (E-Modul des Werkstoffs)
E		Element
F_N / F_G	N	Normalkraft und Gewichtskraft (Last)
G	-	Graph, z. B. Graph $G=(E, R)$, d. h. ein Graph, welcher Elementen E und Relationen R umfasst
H_{ab}	-	Matrix H_{ab}
I_y / I_z	mm^4	Flächenträgheitsmoment
R	-	Relation(en)
S_{ab}	-	Matrix S_{ab}
T_{ba}	-	Matrix T_{ba}
V	mm^3	Volumen
W_y	mm^3	Widerstandsmoment
X	-	Teilmenge X
Y	-	Teilmenge Y
d	-	Distanz (distance) ohne spezifische Einheit
d_a	mm	Außendurchmesser
d_i	mm	Innendurchmesser
l	mm	Länge
m	kg	Masse
n	-	Index (z. B. für $n \times n$ Matrix)
s	mm	Wandstärke
ε	-	Längsdehnung
ρ	g/mm^3	Dichte (z. B. des Werkstoffs)
ϑ	$^{\circ}\text{C}$	Temperatur in $^{\circ}\text{C}$ Grad Celsius
ω_{max}	mm	Durchbiegung

Symbol	Einheit	Beschreibung
ω'_{max}	-	Neigung
€	Euro	Symbol für die Geldeinheit Euro
∞	-	Symbol für unendlich

Abkürzungen

Abkürzung	Beschreibung
AK	Akkumulierte Kosten
AN	Analyse
AND	und-Verknüpfung
API	Application Programming Interface
ASM	Dateiformat eines CAD-Programms (assembly)
AS4MAVEN ²	Bezeichnung des entwickelten Assistenzsystems, d. h. agiles, adaptives und analytisches Assistenzsystem Modelling, Analysis, Visualisation, Evaluation, Navigation und Negotiation)
BPMN	Business Process Model and Notation
BG.E	Baugruppeneigenschaften
BoM	Bill of Materials (Stückliste)
C	Connectoren (Konnektoren)
C&C-Ansatz	Contact and Cannel-Ansatz
CAD	Computer-aided Design
CAM	Cambridge Advanced Modeller der University of Cambridge in England
CAX	Computer-aided x, wobei x ein Platzhalter für diverse computerunterstützte IT-Systeme und Technologien ist
COM	Component Object Model
CoP	Community of Practice
CPM	Characteristics-Properties Modelling
Common-KADS	Common-Knowledge Acquisition and Documentation Structuring
CPI	Change Prediction Index
CRM	Customer Relations Management

Abkürzung	Beschreibung
DMAIC-Zyklus	Define, Measure, Analyze, Improve, Control Zyklus
DMM	Domain Mapping Matrix
DSM	Design Structure Matrix
EK	Elementkosten
ET.M	Einzelteilmerkmalen
EIG	Eigenschaft(en) des Produkts
EPK	Ereignisgesteuerte Prozesskette (Modellierungssprache)
F	Funktion (function)
FAU	Friedrich-Alexander-Universität Erlangen-Nürnberg
FBS	Function-Behaviour-Structure (Framework)
FEM	Finite-Elemente-Methode
FMEA	Fehlermöglichkeits- und -einflussanalyse
FU	Funktionsumfang
GUI	Graphical User Interface
graphml	XML-basiertes Dateiformat für die Repräsentation von Graphen
HDDSM	High-Definition DSM
HIKE	Hybride Intelligente Konstruktionselemente
ID	Identification (Identifikation)
INCOSE	International Council on Systems Engineering
IPE	Integrierte Produktentwicklung
IC	Input in columns
IR	Input in rows
IVE	Integrated Value Engineering
KID	Knowledge, Information, Data (Wissen, Informationen, Daten)
LSSs	Leitstützstrukturen
MBSE	Model-based Systems Engineering
MDM	Multi Domain Matrix
MER	Merkmal(e) des Produkts
MKM	Münchener Produktkonkretisierungsmodell
MO	Modellierung

Abkürzung	Beschreibung
MOKA	Methodology and tools Oriented to Knowledge-based Applications
MS	Microsoft
MSX-CDSM	Multi-Solution eXtended Conceptual Design Semantic Matrix
NASA	National Space Administration
NX	3D-CAD-System Siemens NX
OMG	Object Management Group
OG	Organisationsdomäne
OOSE	Object-Oriented Systems-Engineering Method
OPM	Object-Process Methodology
OR	oder-Verknüpfung
P (PBR)	Property (Eigenschaft, i. d. R. eines Produktelements)
PDCA	Plan-Do-Check-Act-Zyklus
PDD	Property-Driven Development/Design
PE	Produktentwicklung
PEP	Produktentwicklungsprozess
PLM	Product Lifecycle Management
prt	Dateiformat eines CAD-Programms (part)
PTC	Parametric Technology Corporation
q	quantified (quantifiziert)
QFD	Quality Function Deployment ist eine bekannte Methode der Qualitätssicherung
RASCI	Responsible, Accountable, Support, Consulted, Informed und dieses englischsprachige Akronym steht für eine Methode zur Analyse bzw. Darstellung von Verantwortlichkeiten
RE	Requirements Engineering
SA	Studentische Arbeiten, die vom Autor betreut wurden
SE	Systems Engineering
SIPOC	Supplier, Inputs, Process, Output, Customer und dieses englischsprachige Akronym steht für eine Methode im Rahmen der Prozessmodellierung
SysML	Systems Modeling Language
SQL	Structured Query Language (strukturierte Abfrage-Sprache)
SZ	Spezifikationsdomäne

Abkürzung	Beschreibung
TOTE	Test-Operate-Test-Exit-Schema
tlp	Dateiformat des Visualisierungsprogramms Tulip
TRIZ	Teoria reschenija isobretatjelskich sadatsch und dieses russische Akronym steht für die Theorie des erfinderischen Problemlösens
UDT	Universal Design Theory
UI	User Interface
UML	Unified Modeling Language
UMG	Umgebungsbedingungen
UX	Benutzerfreundlichkeit
VB API	Visual Basic Application Programming Interface
VDI	Verein Deutscher Ingenieure
VER	Verhalten des Produktelements
VI	Visualisierung
VOC	Voice of the Customer (Stimme des Kunden) ist eine Methode
VUCA	Volatility, Uncertainty, Complexity Ambiguity und dieses englischsprachige Akronym beschreibt Rahmenbedingungen
W, WI	Wissens- und Informationsdomäne
WFs	Wirkflächen
WFPs	Wirkflächenpaare
X-CDSM	eXtended Conceptual Design Semantic Matrix
XMI	Dateiformat Extended Multiple Instrument Digital Interface
XML	Auszeichnungssprache Extensible Markup Language
XLSX	Dateiformat von MS Excel
XT	eXtreme Tailoring
XOR	Exklusive-oder-Verknüpfung

1 Einleitung

“I think the next century [21st] will be the century of complexity.”

STEPHEN W. HAWKING (1942-2018)

Komplexität ist eine der großen Herausforderungen unserer Zeit. So ist auch nach EHRENSPIEL & MEERKAMM Komplexität ein zentrales Phänomen der Natur, was sich u. a. in den Naturwissenschaften von der Astronomie bis zur Zellteilung wahrnehmen lässt [151]. Neben Komplexität ist unsere heutige Welt vor allem auch durch Volatilität bzw. Unbeständigkeit, Unsicherheit und Mehrdeutigkeit geprägt, was oft mit dem Akronym VUCA für die englischen Begriffe ‚volatility‘, ‚uncertainty‘, ‚complexity‘, ‚ambiguity‘ bezeichnet wird [399]. Der Terminus VUCA wurde erstmals in den späten 1990er Jahren vom Militär verwendet [281] und wird inzwischen zur Beschreibung der gegenwärtigen, allgemeinen Rahmenbedingungen in vielen Publikationen (vgl. u. a. [45], [282], [253]) herangezogen.

Diese **zunehmende Volatilität, Unsicherheit, Komplexität und Mehrdeutigkeit** führt in nahezu allen Branchen zu einer Veränderung der Märkte und damit auch der Wettbewerbssituation [130], [217]. Auch Unternehmen sind immer mehr mit der sich verändernden VUCA-Welt konfrontiert [122]. Als Konsequenz müssen Entscheider VUCA in ihren Entscheidungsprozessen mit einbeziehen [207].

Ähnlich wie in MACK & KHARE [362] werden auch in der vorliegenden Arbeit **Komplexität als Ursprung** und die anderen drei Dimensionen als ihre Folgen angesehen (s. Anhang A.2). Nicht nur in zahlreichen Lebensbereichen und Branchen, sondern v. a. auch in der Produktentwicklung (PE) nimmt die Komplexität stetig zu [151]. Diese Arbeit will daher einen Beitrag für ein holistisches, effektives und effizientes Komplexitätsmanagement in der integrierten Produktentwicklung (IPE) leisten und dadurch Entwickler bei ihrer komplexen Arbeit unterstützen.

1.1 Ausgangssituation und Rahmenbedingungen

Die Ziele der PE leiten sich von den jeweiligen Unternehmenszielen ab [151]. Eines der wichtigsten **Unternehmensziele** ist der Gewinn, also die Differenz zwischen Erlösen (bzw. Erträgen) und Kosten (bzw. Aufwand) in einer zeitlichen Periode (z. B. einem Geschäftsjahr). Für ein erfolgreiches Unternehmen sind daher gewinnbringende, erfolgreiche Produkte eine der wichtigsten Voraussetzungen für die Erreichung der Unternehmensziele

[151] und im Allgemeinen auch für eine florierende Wirtschaft [350]. Da der Unternehmenserfolg durch den Produkterfolg bestimmt wird und die Erfolgsverantwortung für das Produkt vornehmlich im Produktentwicklungsbereich liegt, ist die Bedeutung des Entwickelns in produzierenden Unternehmen, die in dieser Arbeit im Fokus stehen, beträchtlich [151].

Für den **Produkterfolg** ist grundsätzlich die ausreichende Nachfrage auf Kundenseite (marktspezifische Ziele bzw. externe Sicht) genauso wichtig wie die wirtschaftliche Leistungserbringung auf Anbieterseite (unternehmensspezifische Ziele bzw. interne Sicht) [350]. Aus Sicht der PE sind daher für die Kundenbefriedigung Produkte mit Eigenschaften zu entwickeln, die für Käufer und/oder Nutzer einen – häufig gegenüber der Konkurrenz – (Mehr-)Wert haben (z. B. benötigte Funktionalität, gute Ergonomie, geringe Emissionen) [151]. Diese marktspezifischen Ziele lassen sich grundsätzlich mit hoher Qualität, niedrigen Preisen und kurzer Lieferzeit zusammenfassen [151], wobei Kunden immer mehr auf ein angemessenes bzw. möglichst gutes Verhältnis zwischen den benötigten Funktionen und den drei Seiten des magischen Dreiecks Qualität, Zeit und Preis (bzw. Kosten) achten. Zudem lassen sich aus den Unternehmenszielen für die PE auch unternehmensspezifische (auch als herstellerspezifische bzw. als interne bezeichnete) Ziele ableiten [151]. Dazu gehört das zentrale Ziel, die unternehmensinternen direkten und indirekten Kosten möglichst zu verringern oder zumindest gering zu halten. Dabei sind aus Sicht der PE sowohl die in der Entwicklung verursachten Kosten (z. B. Aufwand für den Konstruktionsprozess, wie Personal- und Arbeitsmittelkosten) und die in der PE festgelegten Kosten (z. B. Kosten für die Herstellung oder den Einkauf von Teilen) als auch die Änderungskosten zu beachten. Neben der Einhaltung des Entwicklungsbudgets sind i. d. R. von der PE auch qualitative und zeitliche Ziele zu erreichen.

Diese Ziele müssen zahlreiche Unternehmen in der gegenwärtigen Ausgangssituation erreichen, die durch einen intensiven Wettbewerb und VUCA – insb. weiter **zunehmende, hohe Komplexität** – geprägt ist [354] (vgl. auch Studienergebnisse zum Umgang mit Komplexität in [468]). So steigt auch in der PE die unternehmensexterne und -interne Komplexität, die von Ingenieuren zu beherrschen ist (vgl. Kapitel 1.4) [151]. In [355] und [151] wird die große Bedeutung von Informationen und Wissen im Produktentwicklungsprozess (PEP) an vielen Stellen (z. B. Wissensexplosion als ein wesentlicher Grund der IPE als Knotenpunkt des Informationsumsatzes [151] und somit der Wissenskomplexität) hervorgehoben (vgl. [169], [350]). Mit dieser zunehmenden Wissenskomplexität in Folge des allgemeinen Entwicklungsfortschritts (v. a. in Technologieunternehmen) sind insb.

Entwickler immer mehr gefordert diese Wissenskomplexität zu beherrschen [470]. Ferner kann die interne Wissenskomplexität auch als (z. T. erforderliche) Folge der stetig steigenden externen Informationskomplexität angesehen werden (vgl. [328], [481]). Daher werden die drei internen Arten der Komplexität, d. h. die Produkt-, Prozess- und Organisationskomplexität, nach [355] und [151] in der folgenden Darstellung (s. Bild 1) durch die unternehmensinterne Wissenskomplexität und die unternehmensexterne Informationskomplexität komplementiert. Zu den acht unterschiedlichen Arten der Komplexität werden jeweils auch entsprechende beispielhafte Quellen bzw. (externe) Einflüsse angeführt. Weitere Komplexitätsarten sind in [480] zu finden.

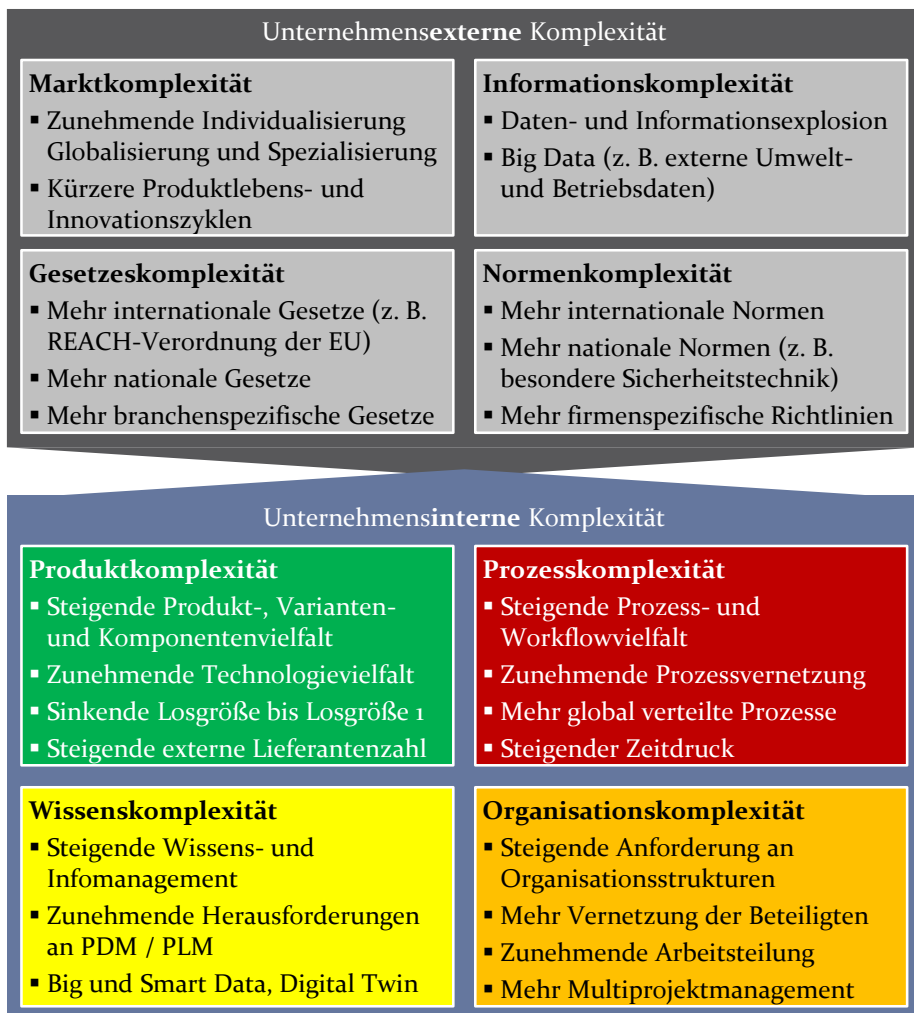


Bild 1: Zunehmende externe und interne Komplexität (in Anlehnung an [355] und [151])

1.2 Problemstellung und Herausforderungen

Wie in Bild 1 zu erkennen ist, betrifft die zunehmende Komplexität nicht nur die unternehmensinterne Produkt-, Prozess-, Wissens- und Organisationskomplexität, sondern auch das gesamte Umfeld des Unternehmens, die Märkte, die Gesellschaft und die ganze Welt (vgl. auch Erläuterung zu VUCA-Welt in Anhang A.2). Aus diesem Grund sind Entwickler mit einer Vielzahl an unterschiedlichen Herausforderungen konfrontiert, wie z. B. der Erarbeitung der Entwicklungsstrategie und -roadmap, der Ideenfindung, dem Verstehen von Patent- und Rechtsfragen, der Verfolgung von Technologieentwicklungen, der Produktneuentwicklung und -weiterentwicklung, der Produktpflege, des Änderungsmanagements [351]. Daraus lässt sich folgern, dass Produktentwickler mit einer zunehmenden **Komplexität umgehen bzw. diese beherrschen** müssen [151].

Aufgrund der Bedeutung, die den Entscheidungen des Entwicklers für den Produkt- und Unternehmenserfolg zukommt, ist es wesentlich Entwickler bei den großen und weiter steigenden Komplexitäts Herausforderungen zu unterstützen. Beim **Umgang mit Komplexität** kann generell zwischen den drei Strategien Komplexitätsvermeidung, Komplexitätsreduzierung und Komplexitätsbeherrschung unterschieden werden [610] (s. Kapitel 2.2.3).

Die **Schaffung von Transparenz** für die verschiedenen Interdependenzen bzw. Wechselwirkungen zwischen Elementen einer und/oder mehrerer Domäne(n) im Rahmen der Entwicklung ist die zentrale Herausforderung bei der Beherrschung von Komplexität. Dabei können unternehmensintern grob vier Komplexitätsarten unterschieden werden: Produkt-, Prozess-, Wissens- und Organisationskomplexität (s. Bild 1 und Details Anhang A.7). Für die Schaffung von Transparenz ist das Verständnis des Kontextes der IPE erforderlich, welcher von einer Vielzahl teilweise miteinander interagierender Einflussfaktoren abhängig ist [211], [421].

Zusammenfassend lässt sich daher schlussfolgern, dass die stark **steigende Komplexität** des Entwicklungssystems sowohl aus den Interdependenzen innerhalb als auch zwischen den unterschiedlichen Domänen bzw. Teilsystemen resultiert. Die Höhe der Komplexität, mit der Entwickler konfrontiert sind, lässt sich verdeutlichen, in dem nicht nur die stetig steigende Anzahl an variantenreichen Produkten, verteilten Prozessen, verwendetem Wissen und spezialisierten Personen, sondern auch deren Wechselwirkungen untereinander sowie die hohe Dynamik und Unsicherheiten eines komplexen Systems betrachtet werden.

Um die vielen verschiedenen Elemente der **komplexen (Teil-)Systeme** in einem Modell geeignet abbilden zu können und um Produktentwicklern eine multikriterielle Bewertung bestimmter Handlungsalternativen zu ermöglichen, sind domänenübergreifende, integrative bzw. holistische Modellierungsansätze erforderlich, welche jeweils kontextbezogen nützliche, mathematische Analyse- sowie Visualisierungsmöglichkeiten erlauben. Derzeit bestehende Ansätze eignen sich hierfür jedoch u. a. aufgrund methodischer Defizite, ihres hohen Modellierungsaufwands, eingeschränkter Analyse- und Visualisierungsmöglichkeiten oder ihrer geringen Unterstützung bei der multikriteriellen Bewertung nur eingeschränkt (s. Kapitel 4).

1.3 Zielsetzung und Aufbau der Arbeit

Zielsetzung der vorliegenden Arbeit ist eine Methodik, mit der Entwickler beim holistischen Komplexitätsmanagement in der PE unterstützt werden, indem kontextbezogen jeweils geeignete Methoden zur Modellierung, Analyse, Visualisierung und Bewertung von komplexen Systemen für eine zielgerichtete Entscheidungsfindung zur Verfügung gestellt werden. Zur effektiven und effizienten Anwendung der Methodik ist auch ein IT-Werkzeug erforderlich, das Entwickler bei deren herausfordernden und für den Produkterfolg entscheidenden Aufgaben im Rahmen des holistischen Komplexitätsmanagements unterstützt.

Zur Erreichung dieser Zielsetzung sind zunächst die Grundlagen für die Modellierung komplexer Systeme zu erarbeiten. Darauf aufbauend erfolgt die Auseinandersetzung mit dem Stand der Forschung und Technik zu domänenspezifischen und -übergreifenden Modellen. Da die matrixbasierte Modellierung die Basis der vorliegenden Arbeit ist, werden Ansätze, Methoden und IT-Werkzeuge des matrixbasierten Komplexitätsmanagements detailliert erläutert und evaluiert. Aus dem daraus abgeleiteten Handlungsbedarf wird eine umfassende Methodik zum Komplexitätsmanagement in der PE sowie ein **Gesamtkonzept für ein unterstützendes Assistenzsystem mit mehreren Modulen zur Modellierung, Analyse, Visualisierung und Bewertung** vorgestellt. Dieses Konzept wird implementiert, um anhand von Anwendungsbeispielen verifiziert und validiert zu werden (vgl. [588]). Dabei haben die kritisch diskutierten Ergebnisse und Erkenntnisse aus der Anwendung des Assistenzsystem gezeigt, dass dieses sowohl funktions- und leistungsfähig ist als auch Entwicklern hilft und befähigt effizientere, effektivere und zielgerichtete Entscheidungen zu treffen. Abschließend erfolgt ein Fazit und Ausblick auf weitere Forschungsfragen. Ein Überblick des inhaltlichen Aufbaus der Arbeit ist in Bild 2 dargestellt.

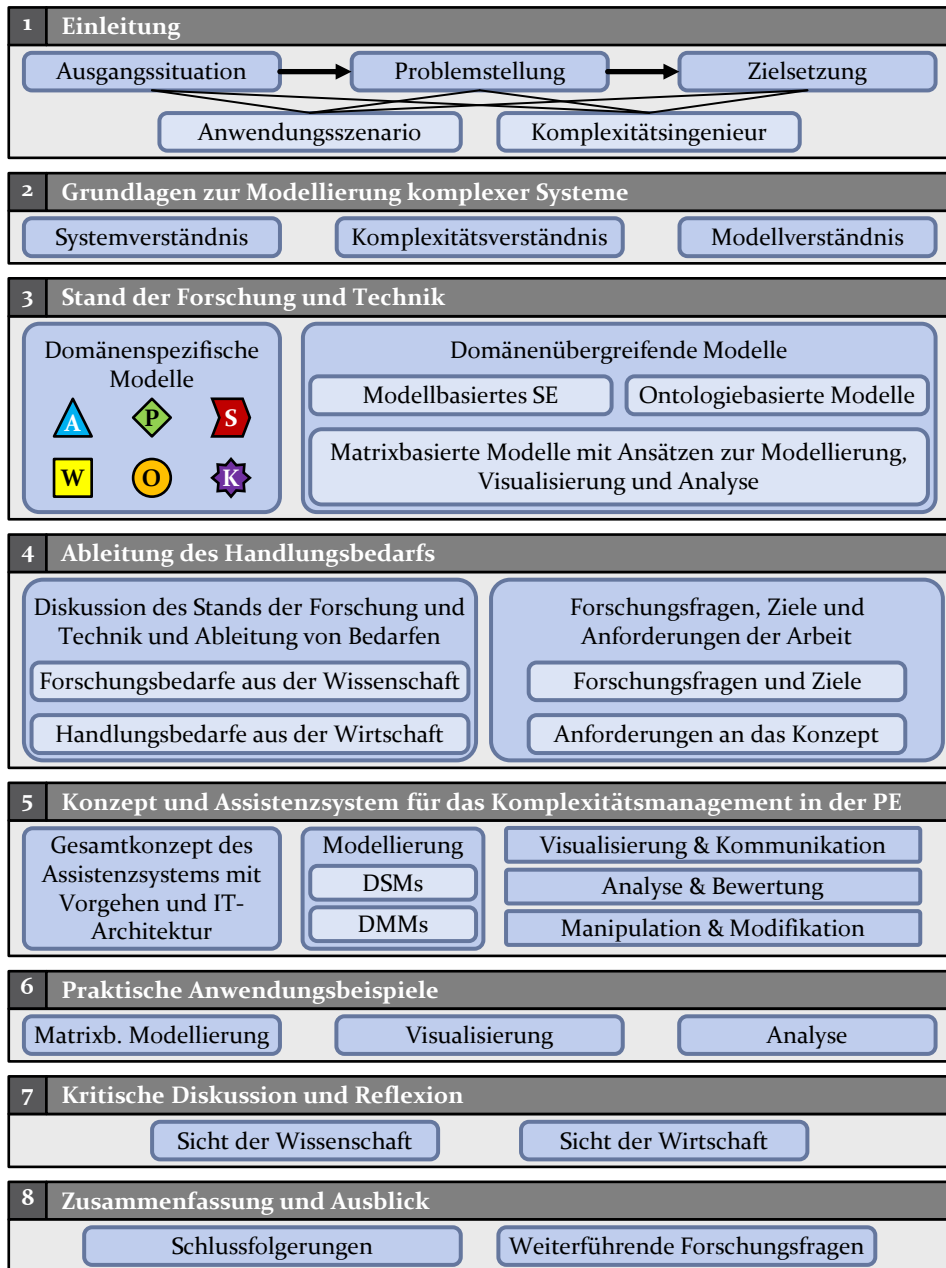


Bild 2: Überblick zum inhaltlichen Aufbau und zur strukturellen Vorgehensweise

1.4 Rolle des Komplexitätsingenieurs

Komplexes Problemlösen bzw. das **Lösen komplexer Probleme** ist laut Studien (vgl. [347], [348]) des World Economic Forums eine seit Jahren der wichtigsten und daher gefragtesten Fähigkeiten von Menschen in Wissenschaft und Wirtschaft. In diesen Studien wird unter dieser Problemlösekompetenz die Fähigkeit verstanden, komplexe Probleme zu erkennen und die damit in Beziehung stehenden Informationen zu analysieren, um v. a. Optionen zu entwickeln und zu bewerten sowie dann darauf aufbauende Lösungen umzusetzen [348].

Die große Wichtigkeit der **Problemlösungskompetenz** in der und für die IPE wird auch in EHRENSPIEL & MEERKAMM thematisiert [151]. Dies liegt v. a. daran, dass der PEP i. d. R. durch hohe Komplexität geprägt ist, die sich z. B. aus der hohen Anzahl verschiedener Aufgaben und zu lösender Problemstellungen sowie aus den teilweise wechselseitigen Abhängigkeiten dieser ergibt [550].

Diese zunehmend wichtiger werdenden Aufgaben gehören nach Auffassung des Autors in den Verantwortungsbereich der Rolle „Komplexitätsingenieur“, welche sich zusätzlich zu den grundlegenden Fähigkeiten eines Entwicklers insb. durch die oben skizzierten Kompetenzen auszeichnet (s. Anhang A.3). Die **Rolle des Komplexitätsingenieurs** wird erläutert, da an dieser viele Überlegungen in dieser Arbeit festgemacht werden. Durch ein „Denken in Rollen“ im Gegensatz zu einem „Denken in Personen“ wird v. a. von der Unternehmensgröße abstrahiert [P7], d. h., dass in großen Unternehmen eine Rolle von mehreren Personen ausgeübt wird und in kleinen Unternehmen eine Person mehrere Rollen wahrnehmen kann [151].

Komplexitätsingenieure sind insb. Mittelmanager (s. Bild 3): Einerseits zwischen der Vision, den Strategien und Zielen im Hinblick auf Komplexität der Unternehmensführung und den diversen operativ(er) arbeitenden Unternehmensbereichen (vertikale Hierarchiesicht), andererseits zwischen den verschiedenen Fachbereichen mit deren jeweiligen spezifischen Komplexitätsarten (vgl. Bild 1) und deren integrativen Interdependenzen (horizontale Hierarchiesicht). In Bild 3 sind die vier Domänen sowie das Anforderungs- und Kostenmanagement farblich hervorgehoben, da diese im Fokus dieser Arbeit stehen. So interagiert der **Komplexitätsingenieur als Mittelmanager mit anderen Akteuren** aus unterschiedlichen Unternehmensreichen (s. für Details Anhang A.3). In Bild 3 sind z. B. die Rollen Produkt-, Prozess-, Wissens- und Organisationsingenieur als Vertreter der vier bereits in Bild 1 skizzierten Domänen dargestellt. Diese Rollen werden zusammen mit domänenspezifischen Modellen im Anhang A.16 erläutert.

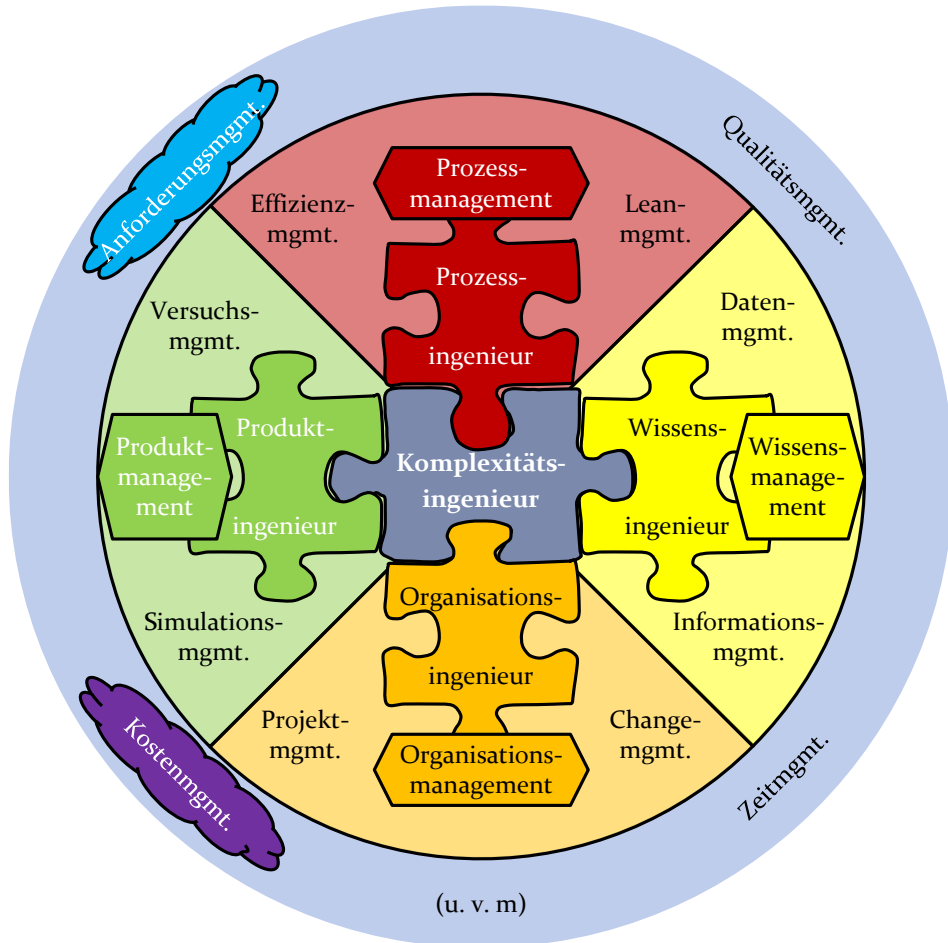


Bild 3: Die Rolle des Komplexitätsingenieurs im Entwicklungskontext

In Folge des Entwicklungskontextes (vgl. Bild 3) und der unterschiedlichen Komplexitätsarten (vgl. Bild 1) ergeben sich für den Komplexitätsingenieur **grundsätzliche Fragestellungen** (vgl. Bild 4), die prinzipiell in jeder Entwicklung zu beantworten sind. Dabei können grundsätzlich zwischen Fragen innerhalb einer Domäne bzw. bzgl. einer einzigen Domäne (sog. intra-Domänen-Fragen) und Fragen zwischen zwei oder mehr Domänen (sog. inter-Domänen-Fragen) unterschieden werden. Einige, beispielhafte intra-Domänen-Fragen sind in Bild 4 in den sechs wolkenförmigen Denkbblasen dargestellt. Die wesentlichen beiden inter-Domänen-Fragen sind: Welche Elemente interagieren zwischen den verschiedenen Domänen miteinander? Wie interagieren diese Elemente miteinander (d. h. Art der Relation)?



Bild 4: Grundsätzliche Fragestellungen des Komplexitätsingenieurs

Der bereits in Zusammenhang mit den verschiedenen Teilbereichen und Arten von Komplexität verwendete **Begriff Domäne** bezeichnet nach dem DUDEN im Allgemeinen ein Spezial- bzw. Fachgebiet i. S. v. einem Gebiet, auf dem sich jemand besonders gut auskennt und/oder betätigt [136]. Dabei besteht eine Domäne aus Elementen und Relationen zwischen diesen Elementen (vgl. [195], [153]).

Im Hinblick auf **intra-Domänen-Fragen** sind z. B. für die Produktdomäne alle Elemente – ausgehend vom Produktgesamtsystem bis hin zu den Produkteigenschaften und Produktmerkmalen einzelner Bauteile – und Relationen zwischen diesen einzubeziehen (analog für die anderen Domänen). Dadurch wird sichergestellt, dass durch die Modifikation bzw. Manipulation eines Elements die Auswirkung(en) auf alle anderen Elemente in der Domäne Produkt (analog für die anderen Domänen) erfasst werden.

In Bezug auf **inter-Domänen-Fragen** sind auch die Relationen zwischen Elementen unterschiedlicher Domänen zu berücksichtigen. Dadurch kann die Frage beantwortet werden, welche Auswirkung(en) die Modifikation bzw. Manipulation eines Elements einer gewissen Domäne auch auf bestimmte Elemente anderer Domänen haben kann. So kann bspw. erfasst werden, dass die Modifikation eines Produktelements auch eine Modifikation zweier Prozesselemente zur Folge hat.

Demzufolge ist die übergeordnete **Aufgabe** des Komplexitätsingenieurs das holistische Komplexitätsmanagement in der IPE. Dabei kann er

grundsätzlich eine der drei Strategien des Komplexitätsmanagements (d. h. Komplexitätsvermeidung, -reduzierung und -beherrschung) oder auch Kombinationen davon verfolgen (vgl. Anhang A.10). Für das zielgerichtete Lösen von Problemstellungen in komplexen Systemen ist v. a. der Einsatz von geeigneten Methoden und IT-Werkzeugen (z. B. zur Modellierung, Analyse, Visualisierung und Bewertung) hilfreich [151].

Im Sinne eines **systemischen Denkens bzw. Systemdenkens** (vgl. Anhang A.2) sind für ein ganzheitliches Komplexitätsmanagement alle relevanten Elemente und Relationen für die Problemlösung und insb. auch für die Entscheidungsfindung zu berücksichtigen. Dabei sind nicht nur die einzelnen Perspektiven der verschiedenen Domänen (intra-Domänen), wie z. B. der Perspektive von Konstrukteuren, sondern v. a. auch die integrative Sicht (inter-Domänen) wegen der Interdependenzen zwischen den Domänen einzunehmen. Komplexe Aufgaben und Entscheidungssituationen in der PE sind häufig so beschaffen, dass mehrere Ziele erreicht bzw. mehrere Anforderungen erfüllt werden müssen. Hieraus resultiert die Herausforderung, dass der Zustand nicht nur von einer Variablen, sondern von mehreren Variablen aus diversen Domänen im Hinblick auf möglichst alle Ziele zu verbessern ist (i. S. einer „Mehrzieloptimierung“). Dabei können die verschiedenen Ziele bzw. Anforderungen konfliktär sein und/oder sich im zeitlichen Verlauf des PEP ändern, was eine wiederholte Neubewertungen der Zielzustände erforderlich macht.

1.5 Beispielhaftes Anwendungsszenario

In diesem Abschnitt werden anhand eines sehr stark **vereinfachten Anwendungsszenarios** typische Problemstellungen, Ziele bzw. Aufgaben des Komplexitätsingenieurs skizziert. Zudem dient dieses Anwendungsszenario auch zur besseren Veranschaulichung verschiedener Aspekte im Stand der Forschung und Technik in Kapitel 2 sowie im Kapitel 4 zur Ableitung des Handlungsbedarfs (s. Anhang A.4 für weitere Erläuterungen).

Für das Anwendungsszenario dient ein Hohlzylinder als exemplarischen Produkt. Dieser stellt als ein **vereinfachtes Rohr** zwar geometrisch keine große Komplexität dar, dient aber sehr gut zur beispielhaften Veranschaulichung diverser Aspekte dieser Arbeit. Zudem sind Rohre in einer Vielzahl von verschiedenen Anwendungsbereichen zu finden (vgl. [189], [174]). Im Maschinenbau sind Rohre als Halbzeuge [256] und als konstruktive Elemente, wie z. B. als Achsen und Wellen (vgl. [406]) in Verwendung. Rohre bzw. Rohrleitungen sind im Allgemeinen Elemente, die zur Führung von Fluiden, wie gasförmigen, flüssigen und feinen festen Stoffen dienen [617].

In diesem Anwendungsszenario wird auf die ebenfalls sehr vereinfachten sechs, bereits in Kapitel 1.2 skizzierten, Domänen eingegangen. Dazu werden einige für das Anwendungsszenario ausgewählte **Spezifikationen** erläutert. Zu den ausgesuchten Anforderungen des Kunden an das zu entwickelnde Rohr gehören ein maximaler Preis, ein maximales Gewicht, eine erforderliche Wanddicke sowie eine Mindestquerschnittsfläche des Rohrs. Zudem ist das Rohr für eine maximal wirkende Kraft bzw. Last (i. S. v. Nutzungsbedingung) für einen Einsatz bei einer gleichbleibenden Temperatur von etwa 20 °C zu spezifizieren (i. S. v. Umgebungsbedingung).

Zu den für das Anwendungsszenario relevanten **Produktelementen** gehören das Produktverhalten sowie einige, wesentliche Produkteigenschaften und -merkmale (s. Anhang A.4 für Erläuterungen). So ist eine möglichst geringe Verformung des Rohrs für den optimalen Einsatz wichtig. Die Produkteigenschaften sind Steifigkeit, Masse, Flächenträgheitsmoment, Volumen, Querschnittsfläche und Wanddicke. Die Wanddicke ergibt sich aus der Festlegung der beiden Produktmerkmale Innen- und Außendurchmesser des Rohrs. Weitere Produktmerkmale des Rohrs sind die Länge und der Werkstoff, mit dem die Dichte und das E-Modul einhergehen. Die vier Produktmerkmale des Rohrs Werkstoff Edelstahl, Innen- und Außendurchmesser d_i bzw. d_a sowie die Länge l sind in der Produktdomäne (PD) in Bild 5 dargestellt.

Die vier gewählten **Prozesselemente** sind in Bild 5 in der Prozessdomäne (PZ) ersichtlich. Für die Rohrkonstruktion ist im Fall einer Neuentwicklung des Hohlzylinders das *3D-CAD-Modell zu modellieren* und der *Werkstoff zu wählen* (im Prozessbeispiel parallel zur CAD-Modellierung). Bei einer Anpassungskonstruktion ist dagegen i. d. R. nur das bereits vorhandene 3D-CAD-Modell zu modifizieren. Danach ist eine *Fertigungszeichnung abzuleiten*, z. B. zur Fertigung oder für den Einkauf, sowie parallel dazu ein *Festigkeitsnachweis durchzuführen* (zumindest in dem skizzierten Fall).

Dazu passende **Wissenselemente** der Wissens- und Informationsdomäne (WI) sind somit v. a. die drei Elemente *2D-Zeichnung*, *3D-CAD-Modell* und *Bericht zum Festigkeitsnachweis* (vgl. Bild 5). Dabei kann bspw. die 2D-Zeichnung als .pdf-Datei, das 3D-CAD-Modell als .step-Datei und der Bericht zum Festigkeitsnachweis als .doc-Datei vorliegen.

Mit Blick auf die Organisationsdomäne (OG) werden für das Anwendungsszenario drei **Organisationselemente** gewählt (vgl. Bild 5). So wird z. B. das Rohr von einem Konstrukteur entwickelt, von einem Berechnungsingenieur berechnet und von einem Einkäufer extern beschafft.

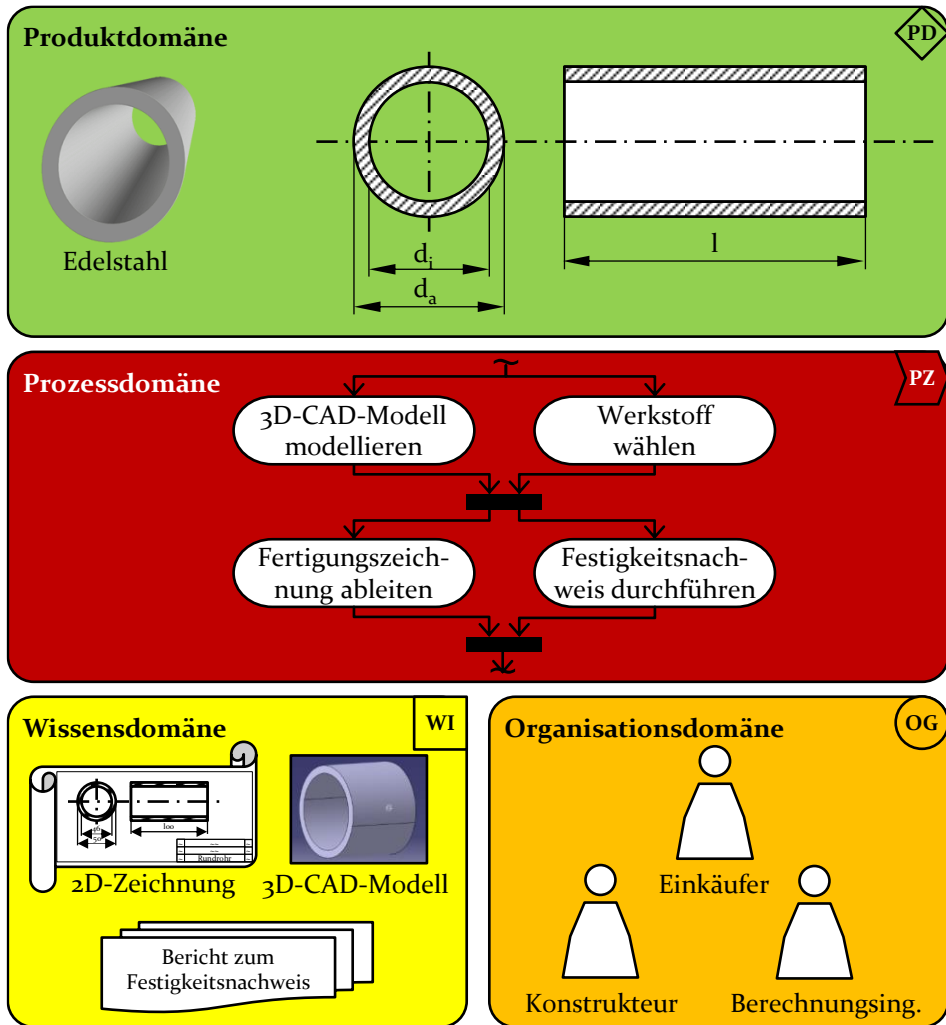


Bild 5: Stark vereinfachtes, beispielhaftes Anwendungsszenario zur Veranschaulichung

Die einzelnen Elemente der fünf bereits beschriebenen Domänen haben auch Auswirkungen auf **Kosten** (vgl. Kapitel 1.1 und Anhang A.9). So fallen bspw. Konstruktionskosten an, die sich u. a. aus dem Konstruktionsstundensatz, der Konstruktionsdauer und den anteiligen Infrastruktur- und Lizenzkosten für die CAD-Software ergeben (analog Berechnungskosten). Hinzu kommen z. B. Einkaufskosten, die sich u. a. aus dem Preis für ein Rohr und den anteiligen, durch Einkäufer verursachten, Kosten ergeben.

Nach dieser kurzen Erläuterung des vereinfachten Anwendungsszenarios mit den sechs Domänen wird im Folgenden die **Produkt Domäne mit Hilfe des vereinfachten Rohrs** etwas tiefergehender dargestellt. Hierfür

werden für das vereinfachte Rohr (vgl. Bild 5) zusätzliche Produktelemente mit einigen ausgewählten Relationen zwischen diesen Elementen eingeführt. Bei den Produktelementen wird zwischen Produktmerkmalen, die direkt vom Entwickler festgelegt werden können, Produkteigenschaften, die sich aus den Merkmalen ergeben, und Produktverhalten, die sich in Folge der Interaktion mit Umgebungs- und Nutzungsbedingungen ergeben, unterschieden (s. CPM-Ansatz in Kapitel 3.1.1). Ferner wird die auf das Rohr an der Stelle $x = l$ wirkende Kraft bzw. Last F_N als eine Nutzungsbedingung sowie die konstante Temperatur ϑ als eine Umgebungsbedingung dargestellt (vgl. jeweils blaue Farbe in Bild 6).

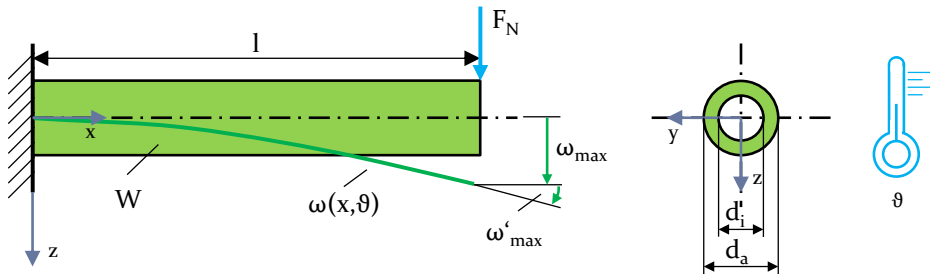


Bild 6: Einseitig eingespanntes Rohr

Im Anhang A.4 werden die wesentlichen Elemente der Spezifikations- und der Produktdomäne für das **vereinfachte Anwendungsszenario Rohr** mit ihrem Typ und Symbol sowie ihrer Einheit ausführlich dargestellt. Dort sind auch Formeln zur Berechnung sowie für verschiedene Produkteigenschaften und -verhaltensweisen entsprechende Beispielwerte zu finden.

Demzufolge lässt sich aus der Beschreibung des Anwendungsszenarios erkennen, dass es selbst bei diesem vereinfachten Rohr schon relativ viele Elemente und auch Relationen zwischen diesen gibt. Dennoch weist dieses Beispiel eine verhältnismäßig geringe „Komplexität“ auf. In der Praxis hingegen werden, z. B. in chemischen Anlagen wie Raffinerien, kilometerlange Rohrleitungen mit einer Vielzahl unterschiedlicher Rohrleitungselemente und weiterer Einzelteile eingesetzt, die eine erheblich höhere Komplexität aufweisen. Für weitere komplexe, technische Systeme sei an dieser Stelle auf moderne Automobile, die je nach Ausstattung und Größe aus deutlich über 10.000 Einzelteilen bestehen können, und auf den Airbus A380, der aus etwa vier Millionen Einzelteilen bestehen kann, verwiesen (s. Kapitel 2.2.2). Vereinfacht ausgedrückt: Die **Komplexität technischer Systeme** steigt exponentiell sowohl mit der Vielzahl als auch Vielfalt an Elementen und Relationen.

2 Grundlagen zur Modellierung komplexer Systeme

“Dem Anwenden muss das Erkennen vorausgehen.“

MAX K. E. L. PLANCK (1858-1947)

Gemäß dieser, durch ihren Namensgeber formulierten, Mission der Max-Planck-Gesellschaft schafft erst das Wissen, welches über grundlegende Gesetzmäßigkeiten in Natur und Gesellschaft, über Strukturen und Zusammenhänge gewonnen wird, die Voraussetzung für wirkliche Neuerungen. Daher wird in dieser Arbeit basierend auf der Zielsetzung zuerst eine detaillierte Betrachtung der relevanten **wissenschaftlichen Grundlagen** in diesem Kapitel vorgenommen. Das übergeordnete Ziel dieser Arbeit ist ein Konzept zur Unterstützung des Entwicklers beim holistischen Komplexitätsmanagement komplexer Systeme in der PE durch geeignete, computerunterstützte Methoden zur Modellierung dieser komplexen Systeme. Darauf aufbauende mathematische Analysen, graphbasierte Visualisierungen und multikriterielle Bewertungen sollen Produktentwicklern bei der zielgerichteten Entscheidungsfindung unterstützen (s. Kapitel 1.3).

Die nachfolgend diskutierten Grundlagen dienen zur wissenschaftlichen Fundierung dieser Arbeit sowie für ein **grundlegendes Verständnis für die Modellierung komplexer Systeme**. Dazu werden im Kapitel 2.1 die wesentlichen, wissenschaftlichen Grundlagen zu Systemen und zum Systemverständnis erläutert und dabei zentrale Begriffe definiert. Zudem wird auch mit Blick auf die PE ein Überblick über die wesentlichen Bestandteile technischer Systeme gegeben. Darauf aufbauend wird der Begriff Komplexität zusammen mit den Arten und Treibern der Komplexität in Kapitel 2.2 erläutert sowie grundlegende Strategien des Komplexitätsmanagements näher vorgestellt. Abschließend werden in Kapitel 2.3 für ein grundlegendes Verständnis von Modellen und deren Modellierung der Modellbegriff und Modellzweck sowie Modellierungsansätze beleuchtet.

2.1 System und Systemverständnis

Bereits Aristoteles hat mit dem Satz „*Das Ganze ist mehr als die Summe seiner Teile*“ zwischen der Menge (griechisch „pan“) und der Ganzheit (griechisch „holon“) differenziert (vgl. für genauen Wortlaut des obigen, verkürzten Zitats in [19]). Dabei bezeichnet er Vielheit als eine Menge, bei der die Anordnung von Teilen keinen Unterschied macht, und im Gegensatz

dazu Ganzheit als eine Menge, bei der die bestimmte Anordnung von Teilen von Bedeutung ist [458]. Daher gilt der griechische Universalgelehrte Aristoteles nicht nur als (ein) Begründer diverser Disziplinen (z. B. Wissenschaftstheorie, Naturphilosophie, Logik) sondern auch als erster großer Systematiker.

In Kapitel 2.1.1 wird ein knapper Abriss zur Systemtheorie skizziert sowie der Systembegriff eingeführt, um für das grundlegende Systemverständnis darzulegen. Darauf aufbauend wird in Kapitel 2.1.2 die Systemtechnik an sich und zusammen mit deren Bedeutung für die Entwicklung technischer Systeme erläutert. Weitere Erläuterungen zu Bestandteilen (technischer) Systeme sowie deren Beziehungen sind in Anhang A.5 und A.6 zu finden.

2.1.1 Systemtheorie und Systembegriff

Die Systemtheorie sowie die Ursprünge des Systemdenkens waren schon im Altertum Gegenstand der Wissenschaft. Ab dem 20. Jahrhundert gewinnt das Denken in Zusammenhängen bzw. die ganzheitliche Betrachtungsweise für prinzipiell alle Wissenschaftsdisziplinen sowohl national als auch international zunehmend an Bedeutung (vgl. u. a. [425], [180]). So wurde die sog. „**Allgemeine Systemtheorie**“ in den 1930er Jahren v. a. von dem Biologen BERTALANFFY entwickelt (vgl. [48]), der als einer der bedeutendsten Systemtheoretiker des 20. Jahrhunderts gilt. Dieser gelangt im Rahmen seiner Arbeiten zu Grundsatzfragen der Biologie zur Erkenntnis, dass sich die Eigenschaften und/oder Verhaltensweisen von Systemen übergeordneter (System-)Ebenen nicht durch die Addition der Eigenschaften bzw. Verhaltensweisen ihrer Bestandteile bei deren isolierter Betrachtung erklären lassen. Zudem erkannte er, dass erst durch die Betrachtung der Relationen zwischen Eigenschaften und/oder Verhaltensweisen von Systemen die übergeordneten (System-)Ebenen von deren (untergeordneten) Bestandteilen erklärbar werden [48] (vgl. weitere Erläuterungen in [50], [49] und detaillierter in [458], [425]). Demgemäß ist das Ganze die Summe seiner Elemente plus die Summe der Relationen zwischen den Elementen, wodurch sich das aristotelische Begriffsverständnis der Ganzheit etwas konkretisieren lässt [458].

Sowohl für die Systemtheorie als auch für den Begriff System gibt es in der Literatur keine allgemein anerkannte Definition [425]. In der Allgemeinen Systemtheorie enthält der **Systembegriff** nach Auffassung von ROPOHL grundsätzlich die folgenden drei Aspekte bzw. Systemkonzepte [458] (s. Bild 7).

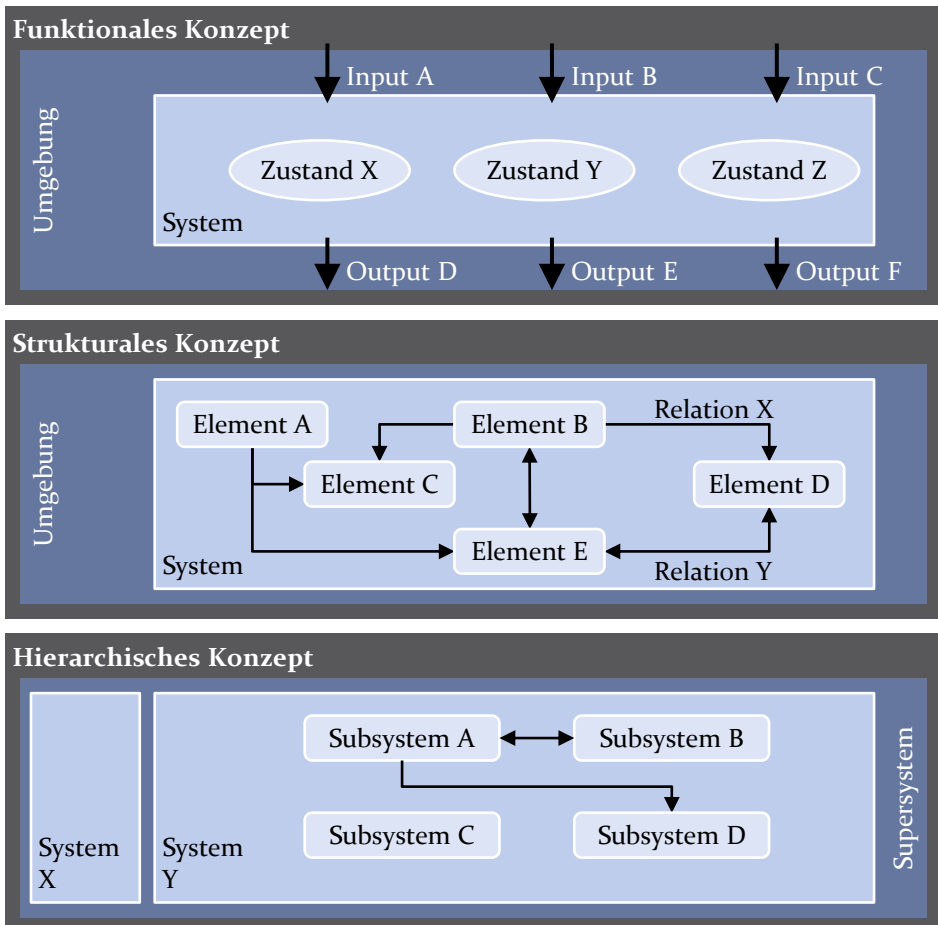


Bild 7: Das funktionale, strukturele und hierarchische Konzept eines Systems (in Anlehnung an [458])

Das strukturele Konzept beschreibt ein System als eine Ganzheit in Relation stehender Elemente (vgl. aristotelisches Begriffsverständnis). Bei diesem Ganzheitskonzept ergeben sich verschiedene Systemeigenschaften aus der Art der einzelnen Elemente als auch aus der Vielfalt der möglichen Relationen zwischen diesen. Somit fokussiert das strukturele Systemdenken auf die Wechselwirkungen zwischen den Elementen eines Systems (vgl. auch strukturorientierte Systembetrachtung in [237]).

Das funktionale Konzept beschreibt ein System als Black-Box, d. h. ohne Kenntnis des inneren Aufbaus und der inneren Zusammenhänge des Systems. Dabei werden die Systemeigenschaften bzw. -verfassung durch die Inputs (Eingangsgrößen), die Outputs (Ausgangsgrößen) sowie die Zustände des Systems beschrieben. Somit fokussiert das funktionale

Systemdenken auf das Verhalten des Systems in der Umwelt ohne Kenntnis des inneren Aufbaus des Systems (vgl. auch wirkungsorientierte Systembetrachtung in [237])

Das hierarchische Konzept beschreibt ein System als ein System von Systemen (engl. systems of systems), d. h., dass ein System mit Elementen wiederum ein Teil eines (hierarchisch) übergeordneten Systems mit mehreren Teilen ist. Daher sind bei diesem Konzept einige Systemebenen zu unterscheiden, wobei ein System einer Betrachtungsebene immer ein Element auf nächst höherer Ebene (von Ganzheit) ist. Somit fokussiert das hierarchische Systemdenken auf den Erkenntnisgewinn zum einen durch die Bewegung abwärts (engl. top-down), die zu einer detaillierten Analyse von Einzelheiten des Systems führt, und zum anderen durch die Bewegung aufwärts (engl. bottom-up), die zu einer stets weiter umfassenden Synthese von Zusammenhängen des Systems führt (vgl. auch umfeldorientierte Systembetrachtung in [237]).

Diese drei Systemkonzepte lassen sich sehr leicht miteinander verweben und in der folgenden **Definition für den Begriff System nach ROPOHL, die auch dieser Arbeit zugrunde liegt**, zusammenbringen [458]: Ein System ist das Modell einer Ganzheit, die erstens – gemäß dem funktionalem Konzept – Beziehungen zwischen Elementen (z. B. Inputs, Outputs, Zustände) aufweist, die zweitens – gemäß dem strukturalen Konzept – aus miteinander verknüpften Elementen (bzw. Subsystemen) besteht, und die drittens – gemäß dem hierarchischen Konzept – von ihrer Umgebung (bzw. vom Supersystem) abgegrenzt wird. Nach dieser Definition des Begriffs System handelt es sich um ein vollständig beschriebenes Systemmodell, wenn alle drei Systemkonzepte modelliert sind [458].

2.1.2 Systemtechnik und Theorie technischer Systeme

Die **Systemtechnik**, der die Allgemeinen Systemtheorie zugrunde liegt, beschäftigt sich mit dem ganzheitlichen Management von komplexen, technischen Systemen. Daher hat die Systemtechnik nach EHRENSPIEL & MEERKAMM sehr enge Beziehungen zur Konstruktionswissenschaft [151]. Hierfür sind besonders die Publikationen *Systemtechnik in der Konstruktion* von BEITZ [43], die *Theorie technischer Systeme* von HUBKA [268] sowie die VDI-Richtlinie *Entwicklung technischer Produkte und Systeme* [580] stellvertretend zu nennen. Dabei umfasst die Systemtechnik zum einen den Teilbereich des (technischen) Systembegriffs zur Beschreibung technischer Systeme und zum anderen auch den Teilbereich der Methodik zur Systemanalyse und -synthese [151].

Nach EHRENSPIEL & MEERKAMM werden die zwei Teilbereiche der Systemtechnik analog auch in der Konstruktionswissenschaft verwendet: Die **Theorie technischer Systeme** dient als abstrakte Beschreibungsmöglichkeit von zu entwickelnden, technischen Objekten (d. h. was festgelegt wird) und die Theorie der Konstruktionsprozesse dient als Darstellung der Denkvorgänge und Tätigkeiten im PEP (d. h. wie festgelegt wird) [151]. Diese beiden Theorien zusammen bilden die Grundlage der Konstruktionsmethodik [151].

HUBKA erklärt ein **technisches System** als ein aus einer gewissen Menge von Elementen bestehendes und mit Relationen zwischen diesen Elementen nach definierten Regeln geordnetes Ganzes [268] (s. Anhang A.6). In ähnlicher Weise definieren EHRENSPIEL & MEERKAMM technische Systeme als künstlich geschaffene, geometrisch-stoffliche Gebilde, die einen gewissen Zweck (Funktion) erfüllen, d. h. Operationen (physikalische, chemische, biologische Prozesse) bewirken (sollen) [151], [173]. Dadurch wird die Allgemeine Systemtheorie und der allgemeine Systembegriff für den Kontext der PE präzisiert (vgl. auch [421]).

Die Gründe für das Entstehen sowie auch für die steigende **Bedeutung der Systemtechnik** sind vielfältig. Nach [425], [458], [237], [239], [64] und v. a. an [151] spielen die zunehmende **Wissens- und Informationsflut** (z. B. können Generalisten das System in seiner Ganzheit kaum begreifen), die unüberschaubare **Vielfalt von Produkten** (z. B. aufgrund verschiedener Varianten und Versionen) und die immer **komplexer werdenden Systeme** (z. B. weil deren Planung, Analyse, Bewertung und Optimierung schwieriger werden) hier eine wesentliche Rolle (s. für Bedeutung und Klassifikation technischer Systeme Anhang A.6).

Der **Nutzen der Systemtechnik** liegt insb. darin, komplexe, technische Systeme (z. B. Produkte, Prozesse) zu modellieren, um mit Hilfe eines Modells das System besser verstehen und handhaben zu können [151]. Die Systemtechnik unterstützt durch die verständliche Strukturierung von Systemen in Teilsysteme mit entsprechendem In- und Output besonders auch die mathematische Modellierung von Systemen [587]. Außerdem begreift die Systemtechnik (nach ihrer Entwicklung, vgl. Kapitel 2.1.1 und Anhang A.5) auch den Menschen (z. B. auch i. S. v. Betrachter, Modellierer) als Teil des technischen Systems. Darüber hinaus berücksichtigt die Systemtechnik nicht nur das zu entwickelnde Produkt, sondern auch die mit diesem in Beziehung stehenden Nutzungs- und Umgebungsbedingungen (vgl. auch [151], [421]).

Zwar unterscheiden sich die Systemdefinitionen in der Literatur, hinsichtlich der wesentlichen **Bestandteile des Modells eines (technischen) Systems** besteht jedoch weitestgehend Übereinstimmung (vgl. u. a. [268], [151], [350], [437], [176], [173], [498], [64], [237], [580]). Im Hinblick auf die Definitionen der wesentlichen Systembegriffe liegt dieser Arbeit insb. die Auffassung von HUBKA zugrunde [268], die anderen Beschreibungen sehr ähnlich ist. Danach umfassen technische Systeme (s. Bild 8) die folgenden Bestandteile. Diese und weitere Systembegriffe sind im Anhang A.5 und A.6 näher erläutert.

- Systemgrenze als Abgrenzung zwischen einem System und der Umwelt (in Bild 8 als Strich-Punkt-Linie dargestellt)
- Umwelt bzw. -feld als Umgebung des Systems außerhalb dessen Grenze (d. h. alles was nicht zum betrachteten System gehört), wie z. B. (übergeordnete) Systeme („Supersysteme“)
- Element als unterschiedliche „Bausteine“ eines Systems, die auch Sub- bzw. Teilsysteme sein können (z. B. „Subsystem E“)
- Relationen (Beziehungen) als i. d. R. verschiedene Verbindungen zwischen den einzelnen Elementen und Systemen, wie z. B. Flussrelationen, hierarchische Relationen (bei technischen Systemen v. a. auch Funktions-, Lage-, Bewegungs- und Kräftebeziehungen [151] und [55])
- In- und Output (Ein- und Ausgangsgrößen) als Relationen des Systems zu seiner Umwelt, welche die Systemgrenze überschreiten

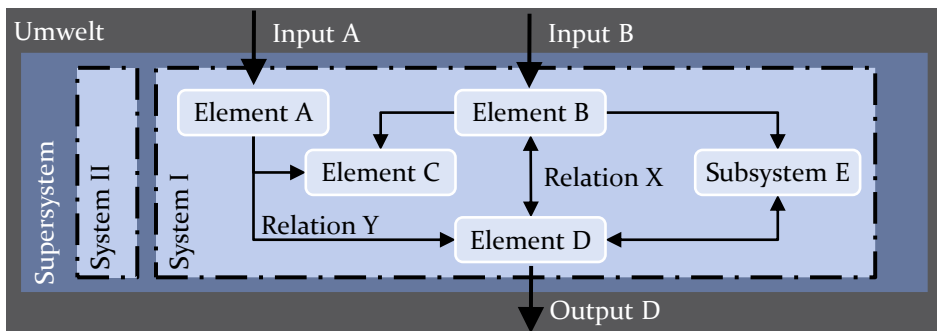


Bild 8: Wesentliche Bestandteile des Modells eines (technischen) Systems

Sofern nicht der Prozess oder das Verfahren, das das System durchführt, sondern ein geometrisch-stofflicher Gegenstand mit System gemeint ist, liegt ein **technisches Produkt** vor [151]. Dann ist ein technisches System auch nach PONN ein in sich geschlossenes, aus einer Menge von Bauteilen und Baugruppen (Modulen) bestehendes, funktionsfähiges Produkt, wie z. B. ein Apparat, eine Maschine oder ein Gerät [437].

2.2 Komplexität und Komplexitätsmanagement

Das vereinfachte Zitat „*For every complex problem there is a simple solution. And it is always wrong*“ von MENCKEN [480], einem bedeutenden Journalisten und Schriftsteller des 20. Jahrhunderts, verdeutlicht, dass **eine einfache Lösung für ein komplexes Problem oft falsch ist**. Da dieses Zitat häufig und unterschiedlich umformuliert und meist gekürzt wiedergegeben wird (z. B. obiges Zitat nach bzw. in [480]), ist das Zitat von MENCKEN selbst ein gutes Beispiel dafür, dass Vereinfachungen oft nicht angemessen sind – sei es im Hinblick auf Zitate oder auch im Umgang mit komplexen Problemen oder Systemen. Der originale Wortlaut des Zitats von MENCKEN lautet nämlich: „Explanations exist; they have existed for all time; there is always a well-known solution to every human problem – neat, plausible, and wrong“ [385].

Nachdem in Kapitel 1 die zunehmende Bedeutung von Komplexität herausgestellt wurde, wird in Kapitel 2.2.1 der Begriff **Komplexität für ein gutes Grundverständnis** dieser Arbeit erläutert. Darauf aufbauend werden Grade bzw. Stufen von sowie Treiber der Komplexität in Kapitel 2.2.2 vorgestellt. Abschließend werden Strategien des Komplexitätsmanagements im Kapitel 2.2.3 aufgezeigt.

2.2.1 Begriffserklärung von Komplexität

Ähnlich wie für den Systembegriff (vgl. Kapitel 2.1.1) lassen sich auch viele **verschiedene Begriffsdefinitionen für Komplexität** finden. Die Vielfalt verschiedener Begriffserklärungen und -interpretationen von Komplexität lässt sich u. a. damit begründen, dass Komplexität in verschiedenen Zusammenhängen, wie z. B. Wissenschaftstheorie und industrieller Praxis, verwendet wird (vgl. u. a. [591], [15]) sowie in verschiedenen Wissenschaften, wie z.B. Ingenieur-, Natur-, Wirtschafts- und Sozialwissenschaften, weit verbreitet ist (vgl. u. a. [368], [91], [479]). Zudem sind die Begriffsdefinitionen der Komplexität auch vom verfolgten Forschungsziel und von der angewendeten Methodik abhängig [480]. Daher stellt sich grundsätzlich die Frage „Was ist Komplexität?“, welche viele Wissenschaftler – auch WEBER – als Titel von Forschungsarbeiten wählen (vgl. [591], [209]). Die Begriffsdefinition und damit auch das Verständnis von Komplexität kann u. a. aus einer etymologischen Sicht, systemtheoretischen Sicht und auch aus einer PE-Sicht erfolgen, wobei es selbst aus Sicht der PE eine Vielzahl von verschiedenen Definitionen gibt (s. Anhang A.7).

EHRENSPIEL & MEERKMANN differenzieren grundsätzlich zwischen einer ausdrücklich sehr **vereinfachten technischen Komplexität** technischer Systeme (vgl. auch Komplexität bzgl. der Merkmale der Systemstruktur in [425]) und einer weitergefassten Objektkomplexität [151]. Dabei ist die bewusst sehr abstrakt vereinfachte technische Komplexität eines Systems von der Anzahl und der Vielfalt der Elemente (Varietät) sowie der Anzahl und Vielfalt deren Relationen (Konnektivität) abhängig und infolgedessen i. d. R. auch ein objektiv feststellbares Maß [151]. Zur Veranschaulichung der vereinfachten (technischen) Komplexitätsdefinition dient Bild 9. So werden in Bild 9 Vielzahl und Vielfalt sowohl von Elementen als auch von Relationen dargestellt (vgl. auch korrespondierende Inhalte aus Kapitel 1).

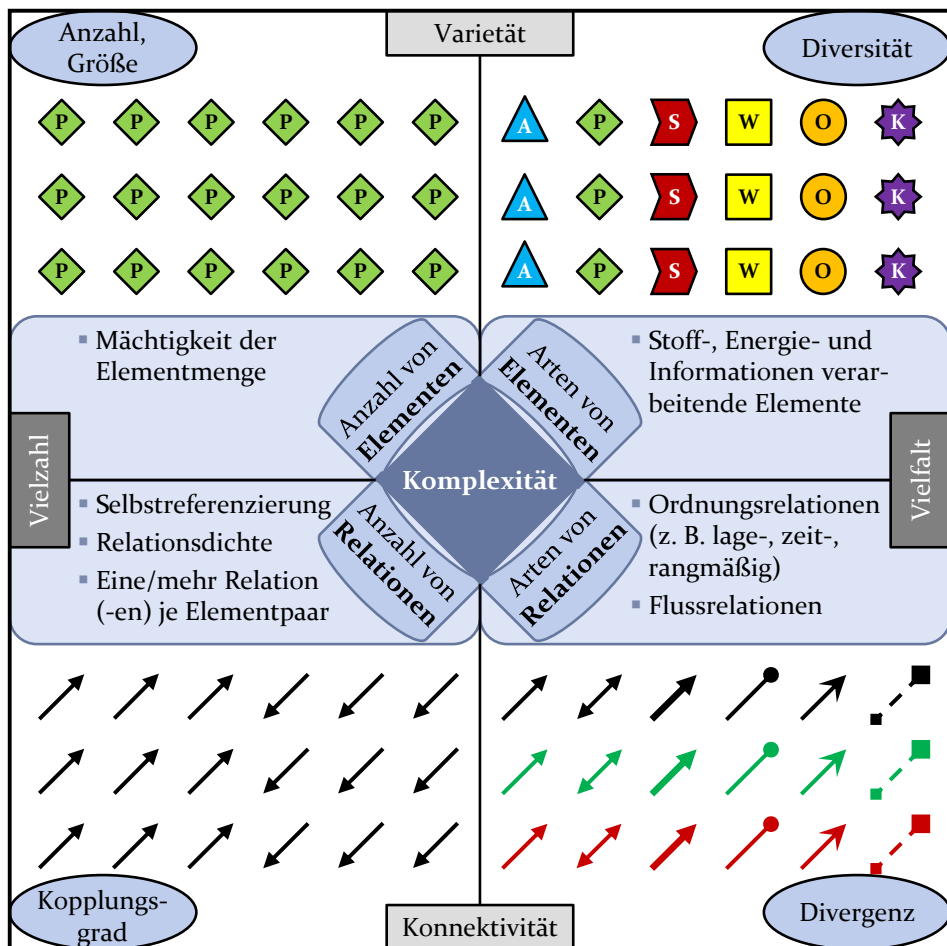


Bild 9: Veranschaulichung der vereinfachten Komplexitätsdefinition (mit inhaltlichen Ergänzungen in Anlehnung an [425])

Im Gegensatz zur vereinfachten Definition der technischen Komplexität umfasst die sog. **Objektkomplexität** neben den beiden Dimensionen Varietät und Konnektivität zwei weitere Komplexitätsdimensionen. Neben der Zahl der Variablen, d. h. v. a. Elemente (Vielzahl), deren Vernetztheit, d. h. v. a. verschiedene, wechselseitige Relationen (Vielfalt) gehören auch die folgenden zwei Aspekte zur Objektkomplexität [151]:

- Unklarheit bzw. Intransparenz (Vieldeutigkeit), d. h., dass die Systemeigenschaften (z. B. Produktelemente und deren Relationen) unklar oder unsicher sind
- (Eigen-)Dynamik (Veränderlichkeit), d. h., dass sich ein System ohne Einfluss bzw. eine Wirkung oder Ähnliches von außerhalb der Systemgrenzen dynamisch verändert.

Ähnlich dazu kann auch nach LINDEMANN die Komplexität eines Systems mittels folgender Merkmale charakterisiert werden: Anzahl, Art, Verschiedenartigkeit und Ungleichmäßigkeit der Aufteilung der Elemente; Anzahl, Art, Verschiedenartigkeit und Ungleichmäßigkeit der Aufteilung der Relationen; (Eigen-)Dynamik des Systems; Art und Anzahl der möglichen Zustände [355], [378]. Somit ist die Definition der Objektkomplexität nach EHRENSPIEL & MEERKAMM [151] sowohl sehr ähnlich zur Definition der Komplexität eines Systems nach LINDEMANN [355] als auch zu den zuvor beschriebenen vier Dimensionen der Komplexität (Vielzahl, Vielfalt, Vieldeutigkeit und Veränderlichkeit) nach REIß [451]. **Dieses Verständnis von Komplexität liegt dieser Arbeit zugrunde** (s. Bild 10).

Komplexitätsdimensionen				
	Vielzahl	Vielfalt	Veränderlichkeit	Vieldeutigkeit
System- elemente Element- komplexität	Anzahl, Größe	Diversität	Dynamik	Freiheits- grade
	Varietät			
System- relationen Relationen- komplexität	Kopplungs- grad	Divergenz	Chaos	Unschärfe
	Konnektivität			
	Überwiegend strukturelle Komplexität			Überwiegend funktionale Komplexität

Bild 10: Vier Dimensionen der Komplexität (in Anlehnung an [451] und ergänzt)

In Bild 10 lassen sich die zuvor beschriebenen Erläuterungen des Komplexitätsbegriffs einordnen, wie z. B. die vereinfachte technische Komplexität nach [151], die sich nur auf die beiden Komplexitätsdimensionen Vielzahl und Vielfalt bezieht (vgl. Bild 9 sowie die in Bild 10 ergänzten Begriffe Varietät und Konnektivität). Die in Bild 10 in Bezug auf Systemelemente und -relationen auf der einen und auf den vier Komplexitätsdimensionen auf der anderen Seite verorteten Begriffe, welche z. T. bereits in Bild 9 angeführt sind (Anzahl, Diversität, Kopplungsgrad, Divergenz), dienen der Präzisierung des Komplexitätsverständnisses (s. für Details Anhang A.7).

In Bild 10 wird die Differenzierung zwischen **struktureller und funktionaler Komplexität** nach KIRCHHOF dargestellt (vgl. [451] und im Gegensatz dazu [66]), welche bei der Charakterisierung und beim Umgang mit komplexen Systemen hilfreich ist [300]. Mit der strukturellen Komplexität ist v. a. die Vielzahl und Vielfalt der Systemelemente und -relationen (i. S. v. Element-, Relationen- und Strukturkomplexität) gemeint (vgl. Definition der vereinfachten technischen Komplexität [151] sowie Komplexität als „Fülle“ in [451]) [300]. Die strukturelle Komplexität lässt sich weiter in eine statisch-strukturelle Komplexität (d. h. einer Funktion aus Anzahl von Elementen und deren Relationen sowie jeweils der unterscheidbaren Vielfalt) und eine dynamisch-strukturelle Komplexität (d. h. Veränderungen von Zuständen und Strukturen der Elemente und Relation mit einer Wahrscheinlichkeit oder Ungewissheit) eines Systems gliedern [300]. Im Unterschied zur strukturellen Komplexität entsteht die funktionale Komplexität durch aktives Tun i. S. v. Handlungen, Entscheidungen und damit Funktionen [300] (vgl. Komplexität als „Lücke“ in [451]). Daher beschreibt die funktionale Komplexität den Umgang mit Komplexität, weshalb deren Komplexitätsverständnis subjektiv geprägt ist [300].

2.2.2 Grade und Treiber der Komplexität

In der Alltagssprache wird Komplexität häufig mit undurchschaubar, schwierig, unerklärlich, unverständlich und auch mit kompliziert assoziiert [300], [91]. Jedoch sind diese Adjektive im Rahmen des Komplexitätsmanagement keinesfalls als Synonyme zu verwenden. Aus diesem Grund ist es sinnvoll, für das zu managende, komplexe System im Vorfeld eine Klassifikation vorzunehmen [480] (vgl. Anhang A.5) und Rolle des Komplexitätsingenieurs in Kapitel 1.4). Zu diesem Zweck wird die Einordnung komplexer Systeme nach ULRICH & PROBST [565] (s. Bild 11), die in der Literatur oft referenziert ist (vgl. [491], [480], [91]), auch zur **Verortung komplexer Systeme in dieser Arbeit** herangezogen.

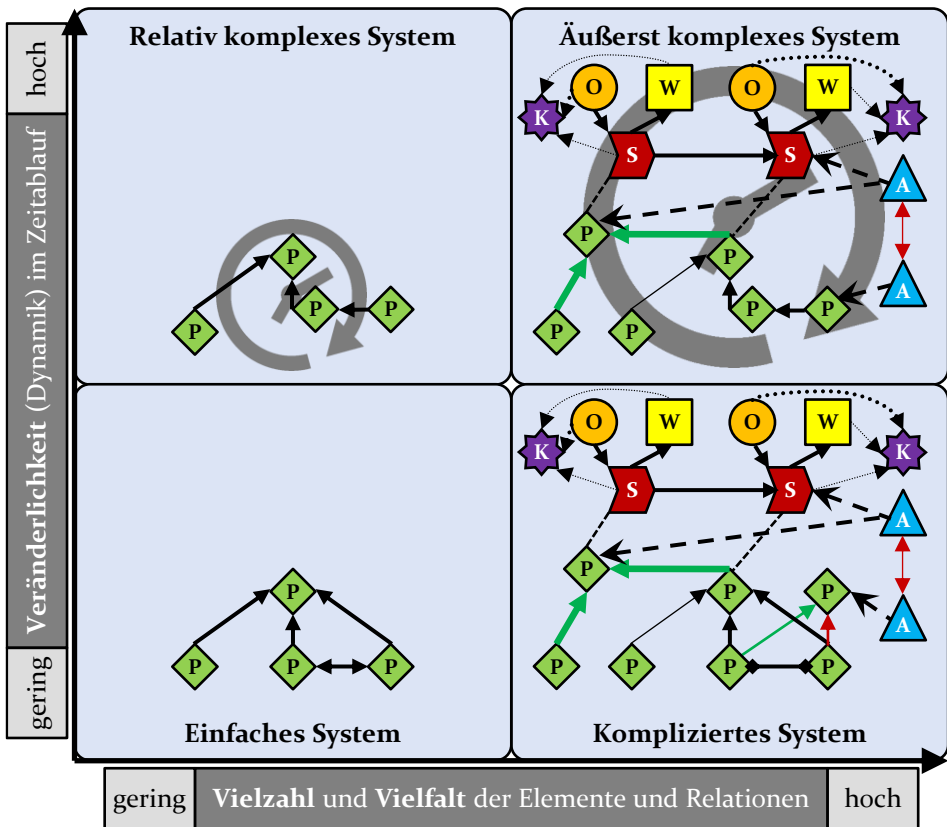


Bild 11: Verortung komplexer Systeme (in Anlehnung an [480] und ergänzt)

Mit dieser Einordnung werden nach SCHOENEBERG die vier Abstufungen **einfache, komplizierte, relativ komplexe und äußerst komplexe Systeme** aus wissenschaftlicher Sicht voneinander abgegrenzt [480], womit ein zunehmender Grad der Komplexität einher geht. Wie in Bild 11 dargestellt, sind mit der Abszissenachse die zwei Komplexitätsdimensionen Vielzahl und Vielfalt und mit der Ordinatenachse die Komplexitätsdimension Veränderlichkeit (Dynamik) berücksichtigt (s. Bild 10), wobei die Vieldeutigkeit in dieser Klassifikation weitgehend nicht einbezogen wird. Im Hinblick auf die horizontale Achse steht die Art der Zusammensetzung und damit die Kompliziertheit des Systems im Fokus, d. h. zum einen Anzahl und Verschiedenheit der Elemente und zum anderen Anzahl und Verschiedenheit der Relationen zwischen den Elementen (vgl. [565] und Bild 9). In Bezug auf die vertikale Achse steht die Veränderlichkeit im Zeitverlauf (in Bild 10 durch die Uhr symbolisiert) und damit auch die Komplexität im Fokus, d. h. einerseits die Vielfalt der Verhaltensmöglichkeiten der Elemente und andererseits die Veränderlichkeit der Wirkungsflächen zwischen den

Elementen (vgl. [565]). Dabei sind die Übergänge zwischen den vier Komplexitätsgraden fließend [480]. Diese Komplexitätsgrade sind nachfolgend sowie in Anhang A.8 anhand des didaktisch gewählten Rohrbeispiels (s. Anhang A.4) erläutert, welches an sich ein vereinfachtes System darstellt.

Einfache Systeme lassen sich durch eine geringe Anzahl und Vielfalt an Relationen charakterisieren, wie z. B. durch vier Produktelemente und mit vier Relationen (vgl. Bild 11). Für ein einfaches System kann die Berechnung des Volumens des Rohrs (vgl. Kapitel 1.5) als Beispiel angeführt werden. So lässt sich mit den drei Produktmerkmalen Außen- und Innendurchmesser sowie Länge das Rohrvolumen berechnen (s. Anhang A.4).

Komplizierte Systeme sind im Gegensatz zu einfachen Systemen durch eine hohe Vielzahl und Vielfalt an Elementen und Relationen sowie ähnlich zu einfachen Systemen durch eine geringe Veränderlichkeit dieser im Zeitverlauf charakterisierbar. In Bild 11 sind beispielhaft das Produktmerkmal Dichte sowie die beiden Produkteigenschaften Masse und Wandstärke aus der Produktdomäne ergänzt. Darüber hinaus ist das System – gemäß dem Anwendungsszenario in Kapitel 1.5 – um je zwei ausgewählte Prozesselemente (3D-CAD-Modell modellieren, Werkstoff auswählen), Wissensselemente (Fertigungszeichnung, 3D-CAD-Modell), Organisationselemente (Konstrukteur, Einkäufer) sowie Anforderungen (maximales Gewicht, erforderliche Wandstärke) und Kostenelemente (Konstruktionskosten, Einkaufskosten) erweitert worden. Zudem lässt sich in Bild 11 erkennen, dass sowohl die Anzahl als auch die Vielfalt an Relationen zugenommen hat.

Relativ komplexe Systeme unterscheiden sich von einfachen Systemen durch die Veränderlichkeit im Zeitverlauf, womit einerseits die Vielfalt der Verhaltensmöglichkeiten der Elemente und andererseits die Veränderlichkeit der Wirkungsverläufe zwischen den Elementen (d. h. der Relationen) gemeint sind. Dabei wird die Veränderlichkeit (Dynamik) im Zeitverlauf in Bild 11 durch die schematische Uhrzeit symbolisiert. Im Unterschied zum Beispiel für das einfache System können sich bei einem relativ komplexen System die Ausprägungen der drei Produktmerkmale (i. S. v. Veränderlichkeit im Zeitverlauf) und damit auch das Produktverhalten ändern. Zudem kann z. B. das Produktmerkmal Außendurchmesser „verschwinden“, da dies nicht mehr vom Konstrukteur zu modellieren ist, und das Produktmerkmal Wandstärke „dazu kommen“ (vgl. Bild 11).

Äußerst komplexe Systeme sind wie relativ komplexe Systeme durch eine hohe zeitliche Dynamik als auch wie komplizierte Systeme durch eine hohe Vielzahl und Vielfalt von Elementen und Relationen gekennzeichnet (s. Bild 11). So sind in Bild 11 die hohe Veränderlichkeit der Elemente und

Relationen im Zeitverlauf durch die Uhr sowie die Vielzahl und Vielfalt an Elementen und Relationen zum einen durch sechs Produktelemente und jeweils zwei Elemente der fünf weiteren Domänen und zum anderen durch viele, verschiedene Relationen zwischen Elementen dargestellt.

Grundsätzlich steigt der **Grad an Komplexität** eines Systems mit einer steigenden Anzahl an variablen Merkmalen (Elementen) sowie auch deren steigender Vernetztheit und zunehmender wechselseitiger Beeinflussung (Relationen) [133]. Laut DOERNER kann es eine einzige (bestimmbare, quantifizierbare) Komplexität eines Systems nicht geben und somit ist Komplexität keine objektive, sondern lediglich eine subjektive Eigenschaft eines Systems [133]. Dennoch lassen sich einige Ansätze zur grundsätzlichen Messung von Kompliziertheit (vgl. u. a. [551]), der Komplexität von Systemen aus Sicht des Managements (vgl. u. a. [368]) sowie zur Komplexitätsmessung von Produkten (s. [491]) und von Produktmodellen (s. [27]) in der Literatur finden. Insgesamt 18 unterschiedliche Komplexitätsmaße, die für die PE relevant sind, werden von AUSTING klassifiziert und erläutert [27].

Komplexität lässt sich nicht nur in diverse Grade differenzieren, sondern kann vielfältige Ursachen haben, die sogenannten Komplexitätstreiber. In Anlehnung an [432] wird in dieser Arbeit unter dem Begriff Komplexitätstreiber ein Phänomen verstanden, das durch einen bestimmten Beitrag die Komplexität in einem (Teil-)System beeinflusst. Generell wird zwischen externen und internen Komplexitätstreibern differenziert, wobei zwischen diesen beiden Arten jeweils enge dynamische Wechselwirkungen und Interaktionen bestehen. In Bild 12 ist ein **Überblick über zentrale externe und interne Komplexitätstreiber** zu finden, wobei auch zwischen den korrelierten und autonomen Komplexitätstreibern unterschieden wird. Die im Vergleich zu den korrelierenden Komplexitätstreiber weniger von den externen Komplexitätstreibern abhängigen autonomen Komplexitätstreiber sind die Prozess-, Wissens- und Organisationskomplexität und weitere Komplexitäten (vgl. Bild 12). In diesem Bild 12 sind die im Fokus dieser Arbeit stehenden Domänen mit den entsprechenden Farben eingefärbt.

Die Wechselwirkungen zwischen externen Komplexitätstreibern, wie der Gesellschafts- und Marktkomplexität, einerseits und der Produktkomplexität andererseits werden u. a. von MARTIN & ISHII in einigen Publikationen unter dem Begriff Design for Variety ausführlich aufgezeigt (vgl. [369], [370], [371]). Demnach gehören zu den relevanten externen **Treibern von Produktkomplexität** u. a. wechselnde Performancebedürfnisse (wie z. B. Gewicht, Größe, Design), neue Umweltauflagen, verändernde Regularien und fallende Preise [371]. Die daher steigende Produktvarianz führt zu

höheren Lagerbeständen und -kosten [369], [370]. Weitere Erläuterungen zu Komplexitätstreibern und deren Wechselwirkungen sind in [91], [480], [66], [216], [300], [614], [84] zu finden.

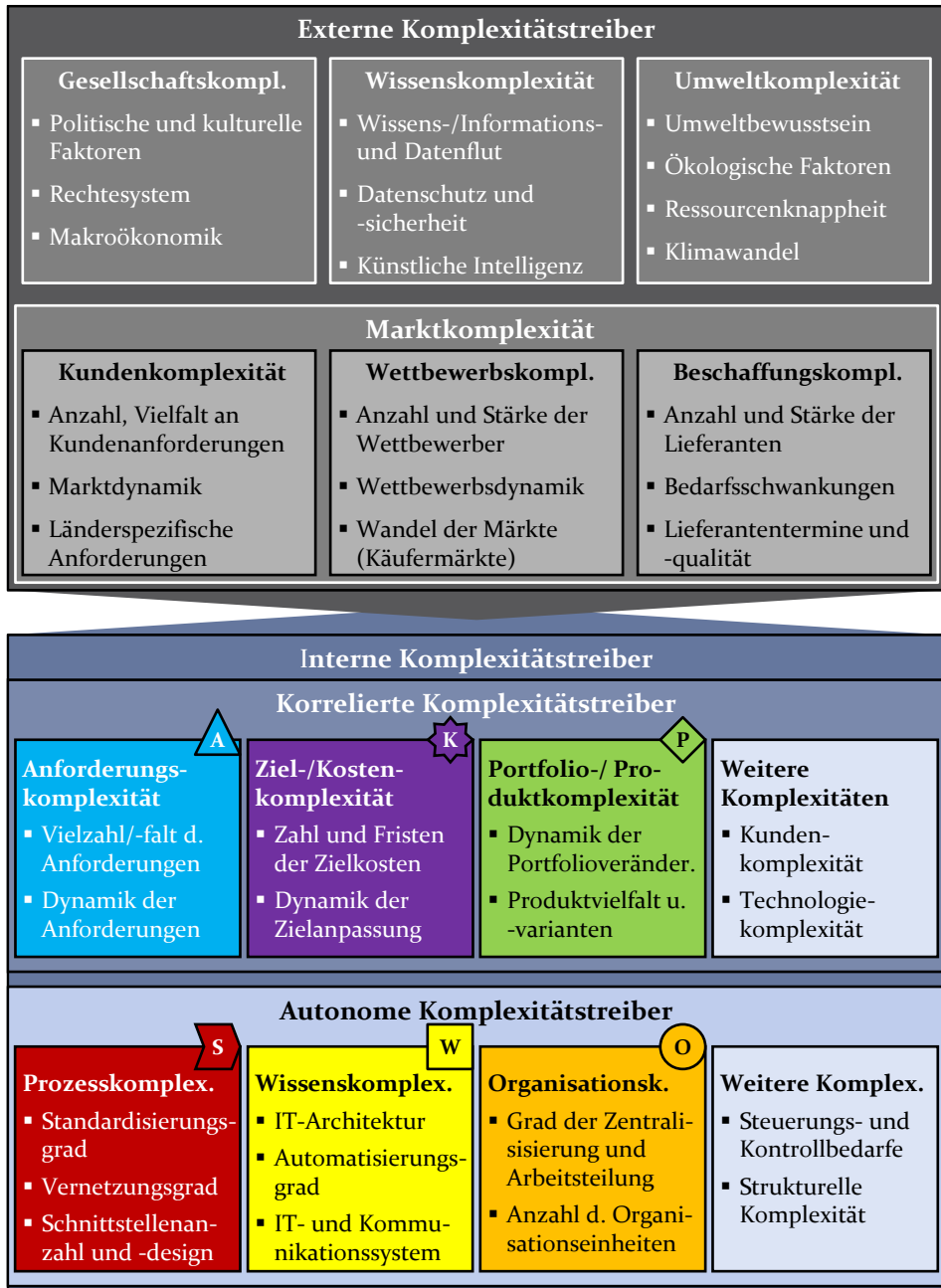


Bild 12: Komplexitätstreiber (in Anlehnung an v. a. [91], [480] und [66], [216], [300], [614])

2.2.3 Strategien des Komplexitätsmanagements

Ähnlich wie für den Begriff der Komplexität (vgl. Kapitel 2.2.1) lassen sich auch verschiedene **Definitionen von Komplexitätsmanagement** finden (vgl. [491]). In dieser Arbeit umfasst Komplexitätsmanagement – in Anlehnung an und Modifikation von [491] – die Planung, Entwicklung und Steuerung der unternehmensinternen Komplexität (bzgl. Anforderungen, Produkten, Prozessen, Wissen, Organisation und Kosten). Dabei wird durch das Komplexitätsmanagement in der PE das Ziel verfolgt, zum einen den Kundennutzen und zum anderen gleichzeitig den Unternehmenserfolg zu maximieren (vgl. [491]). Dazu sind alle relevanten externen und v. a. internen Komplexitätsarten sowie alle Auswirkungen von Komplexität auf unterschiedliche Unternehmensbereiche (vgl. u. a. Kapitel 1 und Anhang A.8) zu beachten.

Eine zentrale Herausforderung ist dabei der **Umgang mit Komplexität**, der sich in verschiedenen Komplexitätsstrategien ausdrücken lässt. In dieser Arbeit wird zwischen den drei grundlegenden **Strategien der Komplexitätsreduktion, -beherrschung und -vermeidung** differenziert (s. Bild 13). Diese drei Strategien des Komplexitätsmanagements sind auch in der Literatur stark verbreitet (vgl. [3], [610], [480], [206]). Dies sind in Anhang A.10 ausführlich beschrieben.

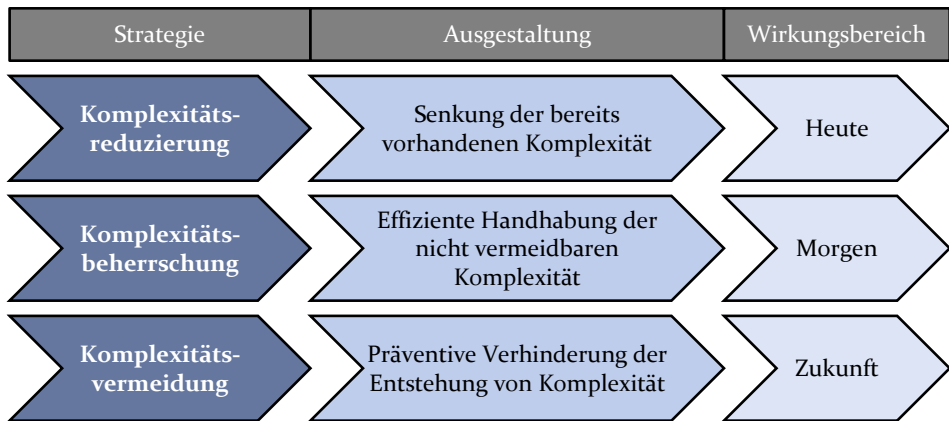


Bild 13: Strategien der Komplexitätsreduktion, -beherrschung und -vermeidung (in Anlehnung an [610], [611])

Die **Komplexitätsreduktion** verfolgt als Ziel bereits vorhandene Komplexität zu verringern. Komplexitätsreduzierende Maßnahmen auf systemstruktureller Ebene erfolgen generell durch Reduktion der Vielzahl und der Vielfalt der Systemelemente und deren Relationen [480]. So kann bspw. durch Modularisierung eine Variantenreduktion erfolgen [480] und

dadurch auch generell die internen Komplexitätsarten reduziert werden. Bei Ansätzen der Komplexitätsreduzierung (vgl. Anhang A.10) ist zu beachten, dass diese in einigen Fällen mit einem negativen Kosten/Nutzen-Verhältnis verbunden sein können. Oft ist eine bestimmte Komplexität erforderlich, um Kundenanforderungen ausreichend zu erfüllen [353].

Die **Komplexitätsbeherrschung** hat das Ziel, die entweder „optimale“ und erforderliche oder nicht reduzierbare und vermeidbare Komplexität möglichst effektiv und effizient zu managen. Nach [611] und [480] ist das Hauptziel der Komplexitätsbeherrschung mit der nicht vermeidbaren internen Komplexität als Folge der externen Komplexität möglichst effizient umzugehen. Daher dienen Ansätze der Komplexitätsbeherrschung (vgl. Anhang A.10) v. a. zur Effizienzverbesserung [480]. Zudem kann eine erfolgreiche Komplexitätsbeherrschung zu einem Wettbewerbsvorteil führen, da allein anhand des Produkts keine Rückschlüsse auf das Know-how der effizienten Komplexitätsbeherrschung gezogen werden können und somit die Imitierbarkeit durch Wettbewerb schwierig ist [615]. Daher dient eine gewisse Komplexität auch als Schutz vor Produktplagiaten [615].

Die **Komplexitätsvermeidung** hat die Zielsetzung, die Entstehung von Komplexität präventiv zu verhindern. Zur Komplexitätsvermeidung sind somit einerseits mögliche Komplexitätsbedarfe vor deren Entstehung im Hinblick auf die diversen Domänen zur Steigerung der Effektivität zu antizipieren [480] und andererseits mögliche Komplexitätstreiber bereits im Produktplanungs- und spätestens im PEP zur Steigerung der Effizienz zu verhindern [611]. Demzufolge hat diese Strategie den längsten Umsetzungs- und Wirkhorizont [611]. Dabei steht eine fortwährende Komplexitätsvermeidung aus systemtheoretischer Perspektive im Widerspruch zu ASHBY's Postulat einer notwendigen Varietät, gemäß der die Ausprägung der externen mit der internen Systemkomplexität in enger Wechselwirkung steht [20]. Daher kann Komplexität nur bis zu einem gewissen Grad vermieden werden, um den Unternehmenserfolg zu realisieren [480]. Grundsätzliche methodische Ansätze zur Komplexitätsvermeidung sind im Anhang A.10 zu finden.

2.3 Modellverständnis und Modellierung

LEONARDO DA VINCI, der als einer der bedeutendsten Universalgelehrten gilt, wird das Zitat „*Der Mensch ist das Modell der Welt*“ zugeschrieben, welches die Bedeutung von Modellen ausdrückt. Er hat sich u. a. als Anatom, Architekt, Bildhauer, Biologe, Ingenieur, Maler, Mathematiker, Naturforscher, Philosoph und Zeichner engagiert. Dabei hat er eine Vielzahl verschiedener und bedeutender Illustrationen bzw. Modelle erstellt, wozu z. B. die detaillierten anatomischen Zeichnungen (vgl. Vitruvianischer Mensch) im Rahmen seiner Anatomiestudien zählen. Die große **Bedeutung von Modellen und des Modellierens** in den diversen Disziplinen war Leonardo da Vinci bewusst. So dienten ihm menschliche Modelle beim Erstellen von Gemälden, wie z. B. dem Porträt der Mona Lisa, Tonmodelle beim Gießen von Bronzestatuen, wie z. B. dem Sforza-Monument, und eine Vielzahl weiterer Modelle, wie bspw. für die Kanalisierung von Flüssen.

Aufgrund der meist sehr hohen Komplexität noch zu planender oder schon realer technischer Systeme (vgl. Kapitel 2.2.2), werden insb. auch in der PE mit Hilfe der Systembetrachtung bzw. Systemtechnik (vgl. Kapitel 2.1.2) mannigfache Modelle entwickelt und eingesetzt (s. [350]). Dabei unterstützt die **Modellierung komplexer, technischer Systeme** auf abstraktem Niveau bei deren Umgang, wie der Analyse, Visualisierung und Bewertung sowie der darauf aufbauenden, abgeleiteten Handlungsmöglichkeiten (vgl. [151]). Hierbei dienen die Systemtechnik sowie das Denken in Systemen zur Strukturierung von Systemen in Subsystemen mit jeweils entsprechenden In- und Outputs die mathematische Modellierung von Systemen und somit deren Eigenschaftsberechnung [587].

Daher werden im Kapitel 2.3.1 zunächst der Modellbegriff diskutiert und Zwecke von Modellen erläutert. Modellräume mit entsprechenden Navigationsmöglichkeiten sind Gegenstand von Kapitel 2.3.2. Außerdem werden einige relevante Grundlagen zur Modellierung im Allgemeinen sowie zu Modellierungsmethoden und mit diesen einhergehenden Modellierungssprachen und -verfahren in Kapitel 2.3.3 erläutert. Dadurch soll ein für diese Arbeit wichtiges, **grundlegendes Verständnis von Modellen und für deren Modellierung** erreicht werden. Ergänzende Grundlagen zum Modellverständnis sowie zur Modellierung und dabei auftretende Herausforderungen sind in Anhang A.11, A.12, A.13, A.14 und A.15 zu finden.

2.3.1 Modellbegriff und Modellarten

Der **Begriff Modell** lässt sich auf das lateinische Wort *modulus*, der Verkleinerungsform von *modus* für Maß, Maßstab und Einheit, zurückführen [296]. Dabei wird Modellbegriff in verschiedenen Anwendungen mit oft unterschiedlichen Bedeutungen verwendet [296]. Nach VAJNA ET AL. werden in der Fachliteratur einige Begriffe im Zusammenhang mit der Modellbildung mit z. T. unterschiedlicher und vereinzelt sogar widersprüchlicher Bedeutung verwendet [569] (s. Anhang A.12). Nach KASTENS & KLEINE BÜNING kann unter einem Modell das Folgende verstanden werden [296]:

- das Abbild eines vorhandenen Originals, z. B. ein Rohrmodell
- das Vorbild für ein herzustellendes Original, z. B. ein neuartiges Rohrleitungsmodell, oder das Vorbild für Maler (vgl. da Vinci)

Dabei kann einerseits das Modell konkret (z. B. ein physisches Rohrmodell) oder abstrakt (z. B. ein Simulationsmodell zur Festigkeitsberechnung) sein und andererseits kann auch das Modellierte konkret (z. B. das Rohr) oder abstrakt (z. B. potenzielle Durchbiegung des Rohrs unter einer Belastung) sein (s. [296]). Auch wenn während des Modellierungsprozesses vom Original zum Modell meist abstrahiert, vereinfacht, formalisiert und Unwesentliches weggelassen wird, muss ein enger Bezug zwischen Original und Modell bestehen bleiben – auch wenn das Modell nur ausgewählte Aspekte des Originals beschreibt [296]. In Anlehnung an STACHOWIAK ist zu beantworten, wovon, für wen, wann und wozu ein Modell erstellt wird und demzufolge kann ein **Modell wie folgt definiert** werden: M ist das Modell des Systems S (i. S. v. des Originals) für den Nutzer N in der Zeitspanne Z bezüglich der Intention I [525]. Diese Begriffsbestimmung von Modell liegt dieser Arbeit zugrunde und wird u. a. in [525] sowie auch im Anhang A.12 präzisiert und ergänzt.

Nach HABERFELLNER ET AL. dienen Modelle von Systemen als Grundlage des Systemdenkens, da ein zentrales Prinzip des Systemdenkens darin besteht, komplexe Systeme mit Hilfe von modellhaften Abbildungen zu veranschaulichen [237]. So werden **komplexe Systeme in einem Modell** strukturiert und abstrahiert, so dass diese besser verstanden (s. [151]) sowie auch analysiert, bewertet und schließlich manipuliert werden können. Unter Manipulation von Systemmodellen wird v. a. das neu erstellen, Auswählen, Hinzufügen, Löschen und Ändern von Elementen sowie Relationen und deren Ausprägungen verstanden (vgl. auch Datenmanipulation durch SQL-Befehle in [68]).

Modelle entstehen also durch Abstraktion und sind zweckorientiert sowie problem- bzw. aufgabenspezifisch. Nach LINDEMANN werden Modelle v. a. für die folgende **Zwecke** eingesetzt [350]:

- Schaffung eines besseren System-, Aufgaben- oder Problemverständnisses bzgl. existierender oder zu entwickelnder komplexer, technischer Systeme u. a. zur Analyse und Bewertung von Schwachstellen und Optimierungspotenzialen,
- Erfassung und Analyse wichtiger Parameter eines Systems, die für eine bestimmte, oft subjektive Perspektive relevant sind, wie z. B. Volumen, Gewicht und Kosten des Rohrs,
- Spezifikation der Struktur oder des Verhaltens eines Systems, wie z. B. Neigung, Durchbiegung, Dehnung des Rohrs,
- Dokumentation von relevanten Systemmerkmalen, wie z. B. Anforderungen, Lösungsprinzipien des Rohrs.

In der PE gibt es eine **Vielzahl verschiedener Modellarten**. Dies liegt v. a. daran, dass es vielfältige abzubildende Systeme (i. S. v. Originale), viele Subjekte (v. a. Modellnutzer und -ersteller), zahlreiche Zeitspannen (z. B. zur Modellerstellung und -nutzung) sowie vielfältige Modellzwecke gibt.

In der **disziplinneutralen Klassifikation** nach STACHOWIAK werden Modelle in vier verschiedene Klassen eingegliedert: Graphische, Technische, Semantische und Semantisch-szientifische Modelle [526] (s. Anhang A.13 für Details). Die Klassifizierungskriterien für die einzelnen Modellklassen werden von STACHOWIAK zwar nicht explizit erläutert, aber zwei zentrale Dimensionen sind grob erkennbar: Einerseits dient der Zweck bzw. Herkunftsbereich der Modelle und andererseits die Art und Weise der formalen Darstellung (z. B. natürliche Sprache, graphische Elemente, mathematische Formeln) als Differenzierungskriterium [280]. Die in den Ingenieurwissenschaften sowie insb. in der PE weit verbreiteten Modelle finden sich prinzipiell in allen vier Klassen wieder.

So werden zwar in der PE **disziplinspezifische Modelle** für die Modellierung einzelner Domänen eingesetzt, aber oft nur in der jeweiligen Fachdisziplin ausreichend gut verstanden. So wird bspw. unter einem Produktmodell je nach Zeitpunkt im PEP ein Funktions-, Prinzip- oder ein Gestaltmodell (vgl. Anhang A.12) und je nach Fachdisziplin z. B. ein CAD- oder Simulationsmodell (s. Kapitel 3.1.1) verstanden. In dieser Arbeit werden auf Basis der sechs Komplexitätsarten (s. Kapitel 1.1) auch die entsprechenden Modellarten voneinander differenziert: Anforderungs-, Produkt-, Prozess-, Wissens-, Organisations- und Kostenmodelle. Ein Überblick verschiedener Modellklassifikationen und -arten ist dem Anhang A.11 zu entnehmen.

Die im **Fokus der Arbeit stehenden Modellarten** sind typische, in der PE verwendete Modelle. Daher werden diese nur kurz wie folgt skizziert:

- **Anforderungsmodelle** (z. B. Lasten- und Pflichtenhefte) bzw. Spezifikationen sind der Ausgangspunkt jeder Entwicklung,
- **Produktmodelle** (z. B. Funktions- und CAD-Modelle) sind diverse immaterielle und/oder materielle Ergebnisse jeder Entwicklung,
- **Prozessmodelle** (z. B. Vorgehensmodelle, Mikrozyklen) dienen der Strukturierung der Entwicklung in Phasen und Teilschritte,
- **Wissens- und Informationsmodelle** (z. B. Ontologien, Wissensgraphen) unterstützen zum einen die Entwicklung von Produkten und zum anderen werden die zunächst durch Daten/Informationen beschriebenen Produkte materiell realisiert (vgl. [151]),
- **Organisationsmodelle** (z. B. traditionelle vs. agile Entwicklung) dienen der Organisation der interdisziplinären Entwicklungsarbeit,
- **Kostenmodelle** (z. B. direkte Kosten, Komplexitätskosten) sind in der Entwicklung einerseits zu berücksichtigen und andererseits auch weitgehend festzulegen (vgl. Anhang A.8 und A.9)

2.3.2 Modellraum und Navigationsmodell der Entwicklung

Modelle werden auch als Mittel zum Komplexitätsmanagement (insb. zur Beherrschung von Komplexität, vgl. Kapitel 2.2.3) mit der Hilfe der Systemtechnik und des Systemdenkens (vgl. Kapitel 2.1.2) entwickelt sowie auch verwendet [437]. In der PE sind eine Vielzahl verschiedener Modellarten in Verwendung (vgl. Anhang A.8). Diese werden in den Modellraum des Entwickelns in Anlehnung an das **Münchener Produktkonkretisierungsmodell** (MKM) verortet, welches im Folgenden erläutert wird [437].

Das **Münchener Produktkonkretisierungsmodell** (MKM) wurde auf Grundlage einiger Entwicklungsmodelle (v. a. dem Modellraum des Konstruierens nach RUDE) entwickelt. Als Beschreibungsmodell für die Entwicklung orientiert sich das MKM an den Eigenschaften der für den PEP relevanten Produktmodelle [437]. Die zwei Hauptkomponenten des MKMs sind der Anforderungsraum und der Lösungsraum, wobei sich letzterer in die Funktions-, Wirk- und Bauebene gliedert [437]. Die zwei Modellräume und drei Ebenen des Lösungsraums sind in [437] wie folgt erläutert:

- **Anforderungsraum:** Anforderungen stellen die Entwicklungsziele bzw. geforderte Produkteigenschaften dar. Dabei wird das Anforderungsmodell im Verlauf des PEPs i. d. R. nach und nach ergänzt, konkretisiert und modifiziert.

- **Lösungsraum:** Der Lösungsraum umfasst die gesamte Menge aller Lösungsmöglichkeiten zur Realisierung der Anforderungen in ein Produkt. Im Lösungsraum werden folgende drei Konkretisierungsebenen unterschieden: die Funktions-, Wirk- und Bauebene.
- **Funktionsebene:** Mittels Funktionen wird das technische System mit seinen Bestandteilen in abstrakter Art und Weise zweckorientiert modelliert. Mit Hilfe des Denkens in Funktionen erfolgt eine Lösung von konkreten (bekannten) Lösungsoptionen und dadurch wird das Finden neuer, innovativer Lösungsmöglichkeiten unterstützt. In Funktionsmodellen sind die Funktionen des Produkts und deren Relationen repräsentiert (z. B. Funktionsstruktur).
- **Wirkebene:** Auf der Wirkebene werden prinzipielle Lösungsideen und -konzepte der technischen Problemstellung dargestellt. Die für die jeweilige Funktion relevanten Lösungsaspekte sind im Wirkmodell abgebildet. Darin werden die Wirkprinzipien zur Realisierung der verschiedenen Teilfunktionen zur Wirkstruktur verbunden.
- **Bauebene:** Auf der Bauebene erfolgt die Festlegung der konkreten Gestalt des Produkts. Das Baumodell umfasst alle Produktbestandteile und deren Relationen in der Baustruktur (vgl. Produktarchitektur). Im Rahmen der Konkretisierung des Baumodells sind viele Anforderungen und DfX-Kriterien, wie Fertig- und Montierbarkeit, zu berücksichtigen. Da diese i. d. R. Wechselwirkungen haben, steigt die Komplexität bzgl. der Produktgestaltung.

Im MKM wird im Vergleich zum Modellraum des Konstruierens nach RUDE die **Bedeutung von Anforderungen** für die PE betont. Während gewisse Anforderungen am Anfang der PE bereits feststehen (z. B. Anforderungen von Kunden und aus Gesetzen), werden zusätzliche Anforderungen erst im Laufe der Konkretisierung von Lösungen im Lösungsraum ergänzt, konkretisiert und modifiziert [437]. Daher erstreckt sich der Anforderungsraum über die gesamte Dimension der Konkretisierung (s. Bild 14) und stellt somit einen eigenständigen Anforderungsraum neben dem Lösungsraum dar. Während des PEPs ist ein andauernder Abgleich zwischen Anforderungs- und Lösungsraum i. S. v. Anforderungsanalyse und Lösungsbewertung notwendig [437]. Dabei sind die Übergänge zwischen den drei Ebenen fließend [437]. Da der PEP i. d. R. durch viele Iterationen und Rücksprünge charakterisiert ist, wird im MKM auf eine prinzipielle Arbeitsrichtung durch Pfeile, wie bei Vorgehensmodellen üblich (vgl. z. B. [323]), verzichtet und stattdessen werden lediglich die drei Dimensionen (d. h. Konkretisierungs-, Zerlegungs- und Variationsgrad) verwendet.

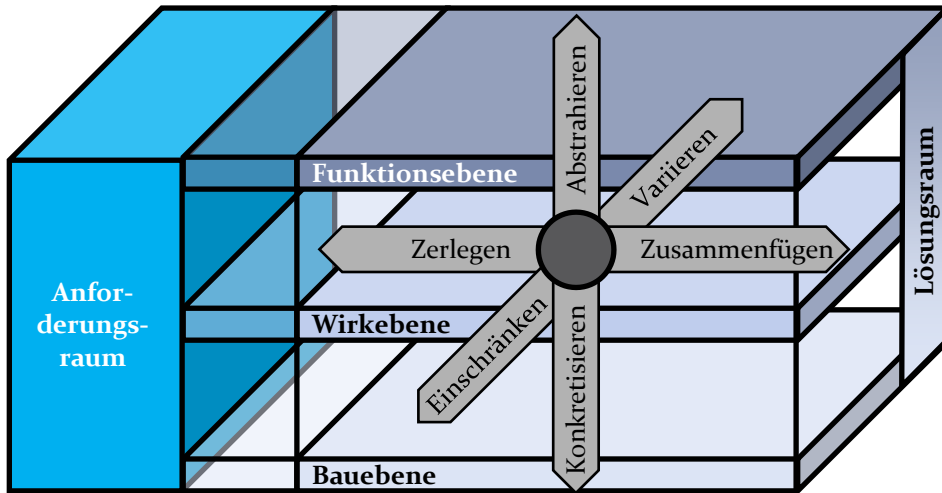


Bild 14: Darstellung von Entwicklungstätigkeiten im MKM (nach [437])

Ähnlich zum Systemdenken, das einem holistischen Denken in Zusammenhängen und auch dem Strukturieren und Gliedern eines Systems in jeweils voneinander abgegrenzte Teilsysteme, die zueinander in Beziehung stehen, dient (vgl. [238], [421] und Anhang A.12), kann i. S. eines **Modelldenkens** der Modellraum im Hinblick auf die drei Dimensionen in Anlehnung an das MKM durchnavigiert werden. Dabei kann der Modellraum aus einem oder auch einer Vielzahl von Modellen bestehen (vgl. Kapitel 2.1). Für das Navigieren durch den Modellraum stehen entlang der drei Dimensionen die sechs grundsätzlichen Entwicklungstätigkeiten zur Verfügung (s. Bild 14), die nachfolgend mit Beispielen [437] erläutert werden:

- **Konkretisierungsgrad:** Abstrahieren, z. B. zur Ermittlung der Funktionen konkreter Bauteile, und Konkretisieren, z. B. zur Ermittlung von Wirkprinzipien zu Teilfunktionen
- **Zerlegungsgrad:** Zusammenfügen, z. B. zum Kombinieren von Teilkonzepten bzw. -lösungen zu einem Gesamtkonzept bzw. einer Gesamtlösung, und Zerlegen, z. B. zur Aufteilung der Gesamtfunktion in Teilfunktionen
- **Variationsgrad:** Einschränken, z. B. zur Auswahl eines Gesamtkonzepts aus mehreren Alternativkonzepten, und Variieren, z. B. zur Generierung von alternativen Lösungsideen für eine Funktion.

2.3.3 Modellierungsmethoden und -sprachen

Nachfolgend werden **die für diese Arbeit relevanten Begriffsdefinitionen skizziert und zueinander in Beziehung gesetzt**, um ein generelles Verständnis für Modellierungsmethoden und insb. Modellierungssprachen zu bekommen. Dieses Verständnis ist v. a. aus zwei Gründen wichtig:

- Die Entwicklung beruht auf vielen verschiedenen Modellen aus verschiedenen Disziplinen (vgl. Tabelle 15 in Anhang A.11), die der Erfassung, Analyse, Visualisierung und Bewertung komplexer, technischer Systeme als auch der Kommunikation der an der Entwicklung beteiligten Personen dienen. Daher wird auch häufig von einer modellbasierten bzw. -getriebenen Entwicklung gesprochen;
- Die Kommunikation zwischen verschiedenen Entwicklern als auch die Entwicklung und Implementierung von IT-Systemen findet mittels geeigneter (Fach-)Sprachen statt (s. [292]). Da auch in dieser Arbeit ein IT-System zum Komplexitätsmanagement konzeptuell erarbeitet und prototypisch implementiert werden soll, ist ein Verständnis der konzeptuellen Modellierung wichtig.

Modellierungsmethoden unterstützen v. a. bei der systematischen, zielgerichteten Vorgehensweise der Modellierung (vgl. Anhang A.14). Unter einer Methode wird im Allgemeinen die Art und Weise eines Vorgehens verstanden. Für den **Begriff der Modellierungsmethoden** lassen sich in der Literatur viele verschiedene Definitionen finden (s. u. a. [218] vs. [291], [294] vs. [185], [187] und vgl. Anhang A.14). Dieser Arbeit liegt grundsätzlich das Begriffsverständnis gemäß FRANK (vgl. u. a. [185], [187]) zugrunde.

Aufgrund der Komplexität von Modellierungsaufgaben ist i. d. R. ein professionelles Vorgehen bzw. eine geeignete Methode erforderlich. Dabei dient eine Modellierungsmethode der Lösung einer Klasse von Problemen [185]. Eine **Modellierungsmethode** besteht nach FRANK aus mindestens einer Modellierungssprache sowie mindestens aus einem entsprechenden Modellierungsverfahren, das bei der Konstruktion und Analyse von Modellen unterstützt, und enthält Annahmen über erfolgreiche Handlungsmuster [187] (s. Anhang A.15 für Details).

Modellierungssprachen dienen zur Erstellung eines externen Modells (aus einem internen Modell) bzw. eines konzeptuellen Modells (vgl. Anhang A.12) [187] und damit zur Abbildung eines Realitätsausschnitts in ein Modell. Zur Dokumentation der Modellierungsergebnisse werden Modellierungssprachen verwendet, die textueller und/oder visueller Art sein können, wobei in der Unternehmensmodellierung (s. Anhang A.12) – somit

auch in der PE – vorwiegend graphische Modellierungssprachen zur Anwendung kommen [467]. Visuelle Modellierungssprachen setzen zur Ergebnisvisualisierung Diagramme ein, welche i. d. R. aus graphischen Symbolen in Form von geometrischen Formen (z. B. Kreise, Rechtecke, Ovale) sowie aus Relationen zwischen diesen Elementen (z. B. Linien, Pfeile) bestehen (s. [467]). Demnach ist eine Modellierungssprache im Allgemeinen eine Menge an symbolischen Ausdrücken mit bestimmten syntaktischen und semantischen Regeln über deren Kombinationsmöglichkeiten [170]. Im Hinblick auf Symbole und v. a. Syntax und Semantik von Modellierungssprachen lassen sich in der Literatur eine Vielzahl unterschiedlicher Definitionen finden (vgl. Anhang A.14 und Anhang A.15), so dass zum einen viele verschiedene Verständnisse von Modellierungssprachen und zum anderen auch eine Vielzahl von Modellierungssprachen gibt.

Dieser Arbeit liegt das grundlegende **Begriffsverständnis von Modellierungssprachen** nach FRANK zugrunde [185], [187] (s. Anhang A.14 für Details). Demzufolge wird eine konzeptuelle Modellierungssprache durch ihre Syntax und Semantik definiert, wobei die abstrakte Syntax Regeln für die Konstruktion syntaktisch korrekter Modelle vorgibt [187]. Die konkrete Syntax definiert die verwendeten Symbole bzw. die graphische Notation zur Repräsentation der abstrakten Syntax [187]. Die Semantik einer Modellierungssprache legt die Interpretation der Modellierungskonzepte fest [187] und erweitert die Syntax, so dass syntaktisch einwandfreie Konstruktionen auch semantisch zulässig sind [188] (s. Anhang A.14 und Anhang A.15 für Details).

Modellierungssprachen sind häufig mit bestimmten **Modellierungswerkzeugen** bzw. Modellierungsmittel verknüpft [37]. Diese Werkzeuge unterstützen bei der Erstellung, Analyse und Manipulation von Modellen und können in konventionelle und IT-unterstützte Werkzeuge klassifiziert werden [467]. IT-unterstützte Modellierungssoftware in der PE sind für die Modellierung von 3D-CAD-Modellen z.B. Catia von Dassault Systèmes, Creo von PTC und NX von Siemens.

Mit zunehmender Bedeutung von IT bzw. Digitalisierung im Rahmen der digitalen Entwicklung wird auch die **computerunterstützte Erstellung und Repräsentation von Modellen** immer wichtiger (vgl. u. a. [313]). In Bild 15 wird der Zusammenhang zwischen dem Modell, dessen digitaler (i. S. v. computerinterner) Repräsentation und der visuellen Darstellung zur Nutzung durch Anwender vereinfacht veranschaulicht. Dabei ist die Visualisierung der rechnerinternen Daten und Informationen in einer für die Nutzer angemessenen und für die jeweilige Anwendung geeigneten

Form essenziell (s. [499]). So hat die visuelle Darstellung des Modells auch nach der VDI 3633 eine hohe Bedeutung, da diese erstens zur Aufbereitung von Daten und Informationen im Rahmen der Modellerstellung, zweitens zur Darstellung der Ergebnisse der Modellanalyse sowie drittens zur Interaktion mit dem rechnerinternen Modell (z. B. Modellmanipulation) durch Menschen insb. bei komplexen Modellen unerlässlich ist [578]. Dabei stellt die Modellierungssoftware mit ihrer graphischen Benutzeroberfläche (engl. Graphical User Interface (GUI)) die notwendigen Schnittstellen sowohl zur Interaktion mit als auch zur Visualisierung der rechnerinternen Repräsentation des Modells bereit (vgl. [313]). Aufgrund vieler Arten und Anwendungen von Modellen (s. u. a. Tabelle 15 in Anhang A.11) gibt es auch eine Vielzahl an Modellierungssoftware (vgl. [467]).

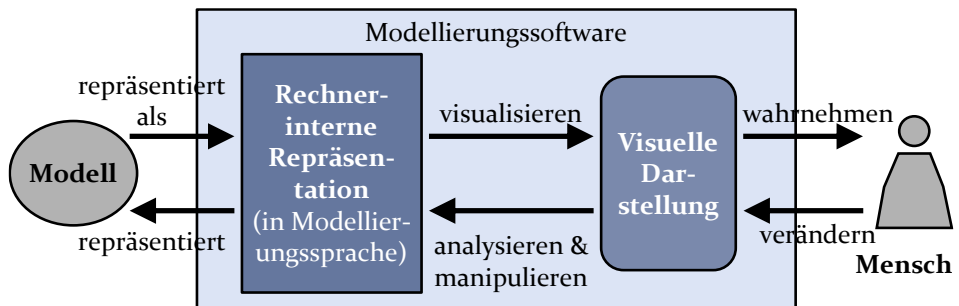


Bild 15: Computerunterstützte Erstellung und Repräsentation von Modellen (in Anlehnung an [313])

3 Stand der Forschung und Technik

*„Wenn du ein wirklicher Wissenschaftler werden willst,
dann denke wenigstens eine halbe Stunde am Tag das Gegenteil von dem,
was deine Kollegen denken.“*

ALBERT EINSTEIN (1879 – 1955)

Aufbauend auf den Grundlagen zur Modellierung komplexer Systeme im Kapitel zuvor werden nachfolgend der Stand der Forschung und Technik zum Komplexitätsmanagement in der PE beleuchtet. Dazu werden relevante, **bestehende nationale und internationale Forschungsarbeiten** vorgestellt, die zum einen für diese Arbeit eine relevante Basis sind sowie zum anderen für die Verortung und Abgrenzung dieser Arbeit dienen.

Die Grundlagen zu den domänenspezifischen Modellen und Modellierungsansätze für die sechs im Fokus dieser Arbeit stehenden Domänen sind in Kapitel 3.1 zu finden. Da die Zusammenhänge und Wechselwirkungen für das Komplexitätsmanagement in der PE immer wichtiger werden, sind die Grundlagen zu domänenübergreifenden Ansätzen zur Modellierung komplexer Systeme in der Entwicklung in Kapitel 3.2 dargestellt. Daraus aufbauend werden wesentliche Ansätze für das matrixbasierte Komplexitätsmanagement, welches im Fokus dieser Arbeit steht, in Kapitel 3.3 ausführlicher beleuchtet. Dadurch wird i. S. des Zitats von EINSTEIN der Stand der Forschung und Technik in den nachfolgenden Kapiteln differenziert abgebildet. Im Vergleich dazu wird der **Beitrag der vorliegenden Arbeit deutlich**, welcher insb. in Kapitel 5 und Kapitel 6 erläutert wird. Zur Veranschaulichung von Ansätzen, Methoden und Werkzeugen wird auf das in Kapitel 1.5 beschriebene Anwendungsszenario Bezug genommen.

Angrenzende Themenbereiche, wie z. B. Grundlagen zum Anforderungs-, Produkt-, Prozess-, Wissens-, Organisations- und Kostenmanagement stehen nicht im Fokus des Kapitels und werden daher nur soweit erforderlich berücksichtigt. Auf diese und weitere relevante Themenbereiche wird an den entsprechenden Stellen lediglich verwiesen.

3.1 Grundlagen domänenspezifischer Modelle

In der PE werden v. a. **domänenspezifische Modelle** für die Modellierung einzelner Domänen eingesetzt. Dabei handelt es sich bspw. um Produktmodelle, die das Produkt u. a. mit allen Merkmalen und Eigenschaften beschreiben. Der PEP ist Gegenstand von Prozessmodellen, welche

einzelne Prozesse und deren Relationen u. a. bzgl. deren zeitlicher Abfolge darstellen. Zudem existieren Wissens- und Informationsmodelle, die u. a. zur Unterstützung des Wissensmanagements und zur Gestaltung von Informationssystemen verwendet werden. Geeignete Koordinationsformen für Aufbau- und Ablauforganisation bieten Organisationsmodelle. Außerdem sind eine Vielzahl von Anforderungs- und Kostenmodelle in der Literatur zu finden (vgl. Anhang A.11 und Anhang A.16).

Neben domänenübergreifenden Modellen und Modellierungsansätzen (vgl. Kapitel 3.2) lassen sich eine **Vielzahl domänenspezifischer Modelle und Modellierungsansätze in der Literatur** finden, die sich grundsätzlich zum Komplexitätsmanagement in der PE eignen. Auch wenn sich die meisten dieser Modellierungsansätze bzw. grundlegenden Modelle nicht nur auf eine, sondern auf mehrere Domänen beziehen, werden diese Ansätze entsprechend der in dieser Arbeit im Fokus stehenden Domänen kategorisiert. Somit werden die Ansätze, worunter i. w. S. auch Theorien, Methoden und Vorgehensweisen gezählt werden (vgl. [557]), bzgl. ihres jeweiligen Schwerpunkts in eine der sechs Domänen verortet:

- Anforderungsorientierte Modellierungsansätze v. a. von Anforderungen (z. B. Kundenforderungen, DfX-Kriterien)
- Produktorientierte Modellierungsansätze v. a. von Produkten (z. B. eigenschaftsbasierte PE)
- Prozessorientierte Modellierungsansätze v. a. von Prozessen (z. B. Vorgehensmodelle)
- Wissensorientierte Modellierungsansätze v. a. von Wissen und Informationen (z. B. wissensbasierte Modelle)
- Organisationsorientierte Modellierungsansätze v. a. von Organisationen (z. B. projektmanagementorientierte PE)
- Kostenorientierte Modellierungsansätze v. a. von Kosten (z. B. kostengerechte PE)

Da die **Produkt- und Prozessdomäne** sowohl bei der Vorstellung des in dieser Arbeit entwickelten Konzepts für das Komplexitätsmanagement in Kapitel 5 als auch bei der Anwendung des implementierten Assistenzsystems in Kapitel 6 im Fokus steht, werden diese in den folgenden beiden Kapiteln beleuchtet. Detailliertere Erläuterungen für diese zwei sowie für die weiteren vier Domänen sind im Anhang A.16 zu finden. Dabei wird neben der Bezeichnung des grundlegenden Ansatzes nur eine kurze Erläuterung mit Hinweis auf entsprechende Vertiefungen in der Literatur angeführt.

3.1.1 Produktorientierte Modellierungsansätze

In der Literatur lassen sich **viele Produktmodellierungsansätze** finden, wie z. B. in VAJNA ET AL. [568] zum Themenbereich CAx. In diesem Kapitel werden zuerst „konkrete“ i. S. v. für bestimmte Disziplinen (z. B. Detailkonstruktion, Festigkeitsanalyse, Komplexitätsmanagement) spezifische Modelle (z. B. CAD-Modelle, FEM-Modelle, Systemmodelle) dargestellt. Weitere ausgewählte, allgemeine Ansätze zur Modellierung von Produktmodellen für die PE sind im Anhang A.16 zusammengefasst.

In der PE werden im Anschluss an die Konzeptphase **3D-CAD-Modelle** des Produkts erstellt (s. Rohrbeispiel in Bild 16), um Geometrieinformationen des Produkts zu repräsentieren. Dabei enthalten CAD-Modelle oft auch das Wissen des Konstrukteurs in Form von Beziehungen zwischen Elementen des Modells (vgl. für Details insb. zur wissensbasierten Konstruktion [568] und [579]). Die Darstellung des Produkts, wie z. B. eines Rohrs, als 3D-CAD-Modell bietet eine gute visuelle Darstellung in Form der Idealgeometrie und ist dem Original optisch sehr ähnlich (vgl. Bild 16). Zudem dienen CAD-Modelle oft als Grundlage für die – vereinfacht ausgedrückt – Ableitung weiterer Produktmodelle [576]. Hierfür sind die CAD/CAM- und die CAD/FEM-Schnittstelle für den rechnerunterstützten Fertigungs- bzw. den Simulationsprozess zwei von vielen Beispielen i. S. des CAx (vgl. [568]).

Häufig schließt sich an die CAD-Modellierung die Berechnung und Simulation des Produktmodells an (sog. CAD/FEM-Schnittstelle). Für die Festigkeitsanalyse benötigen die **FEM-Modelle** neben den geometrischen Informationen i. d. R. auch Informationen zum Material und zu externen Kräften und Momenten in Folge von Belastungen. Im Rahmen der Berechnung und Simulationen ist zu beachten, dass je nach Simulationmethode unterschiedliche Anforderungen vom Modell zu erfüllen sind.

Ein drittes in Bild 16 dargestelltes Produktmodell ist die **technische Zeichnung**, die i. d. R. aus dem 3D-CAD-Modell abgeleitet wird. In technischen Zeichnungen sind u. a. die geometrischen Maße, Toleranzen und oft auch fertigungsrelevante Informationen dargestellt.

Eine weiteres Produktmodell ist die **Design Structure Matrix (DSM)** in Form einer Matrix, in der die Elemente und Relationen eines technischen Systems modelliert werden (vgl. Bild 16 mit den drei Produktmerkmalen Innen- und Außendurchmesser sowie Länge des Rohrs). Derartige Systemmodelle werden v. a. zur Modellierung von komplexen Systemen eingesetzt, um die vielfältigen Elemente und deren Relationen zielgerichtet analysieren und übersichtlich visualisieren zu können (vgl. Kapitel 3.2.3).

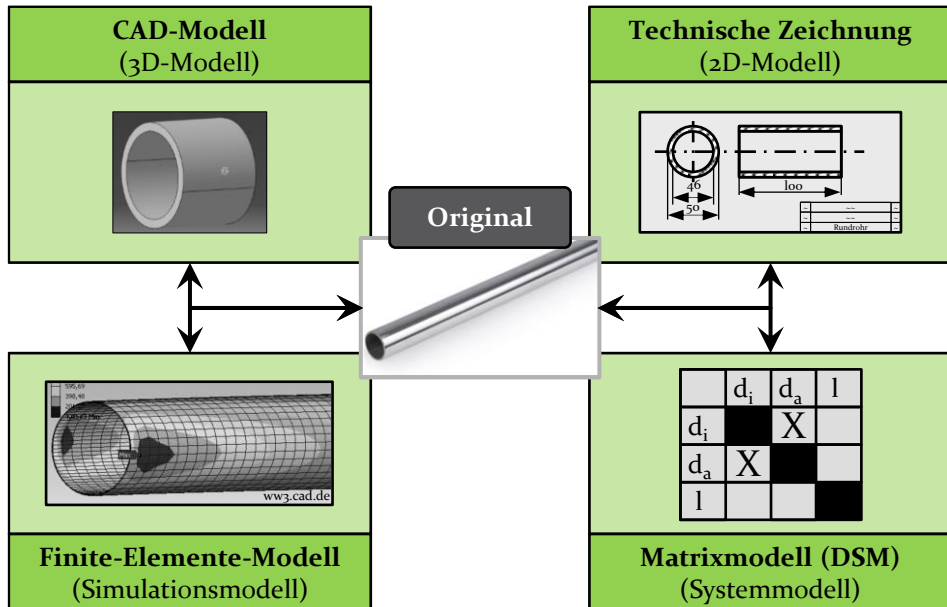


Bild 16: Verschiedene Modelle desselben Originals

3.1.2 Prozessorientierte Modellierungsansätze

Einige **übergeordnete Ansätze zur prozessorientierten Modellierung** der PE bzw. allgemeine Prozessmodelle stehen im Zentrum dieses Kapitels. Da sich in der Literatur eine Vielzahl von Prozessmodellen mit zahlreichen Zielsetzungen und Zwecken finden lassen (vgl. [383]), werden diese in Anlehnung an die Gliederung in [323] in die drei Kategorien handlungsorientierte Ansätze, phasen- und zustandsorientierte Ansätze sowie Mikrozyklen gegliedert und jeweils mit drei in der Literatur verbreiteten Vertretern kurz skizziert (s. Anhang A.16). Im Anhang A.16 sind Ansätze zu finden, die für diese Arbeit relevant sind. Zudem sind u. a. in [323], [156] sowie in der VDI-Richtlinie 2221 [580] weitere Ansätze mit ausführlichen Erläuterungen zu Prozessmodellen zu finden. In Bild 17 sind ausgewählte Prozessmodelle dargestellt und bzgl. deren Detaillierungsgrad und Formalisierungsgrad verortet. Dabei sind in Bild 17 sowohl Prozessmodelle mit nicht standardisierten Notationen zur Visualisierung prinzipieller Abläufe als auch sehr formalisierte und detaillierte Prozessmodelle, die auch von IT-Werkzeugen verarbeitet und somit zur Simulation oder sogar Steuerung von Prozessen (i. S. v. Workflows) genutzt werden können (s. [580]), zu finden.

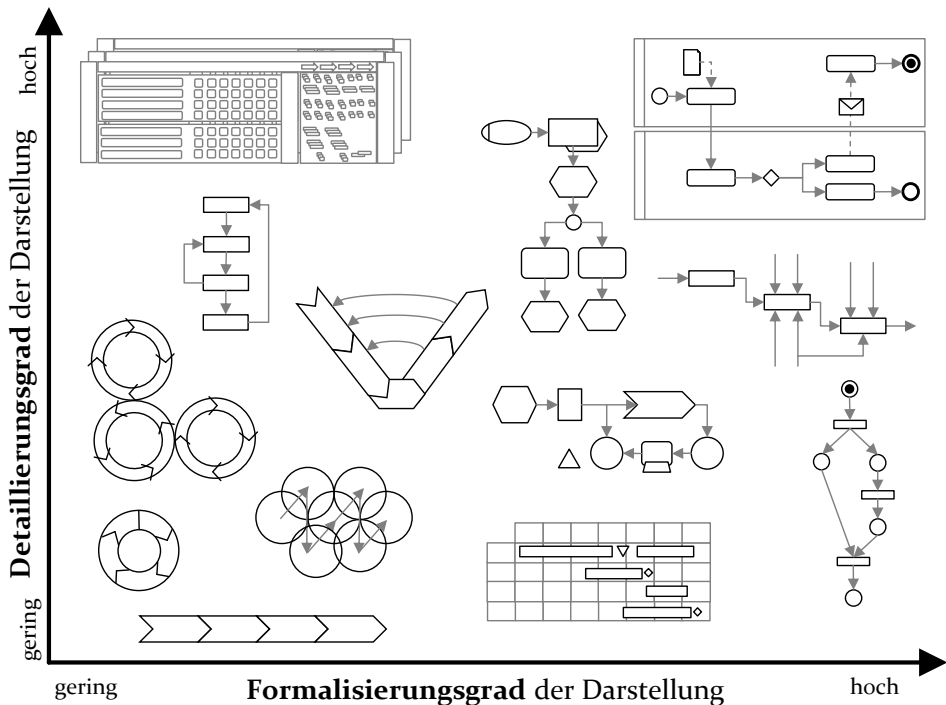


Bild 17: Verschiedene Arten der Darstellung von Prozessen (nach [580])

3.2 Grundlagen domänenübergreifender Modelle

Im Sinne der IPE nach EHRENSPIEL & MEERKAMM sind zur Überwindung domänenspezifischer Spezialisierung sowie Komplexität integrierende Ansätze, Methoden und Softwarewerkzeuge zu verwenden, um durch gemeinsames, kooperatives und integratives Handeln ein optimales Produkt termingerecht und zu günstigen Kosten zu entwickeln [151]. Daher werden in diesem Kapitel ausgewählte **Grundlagen domänenübergreifender Modelle** zum ganzheitlichen Management von komplexen Systemen mit dem Fokus auf modellbasierter und virtueller bzw. digitaler Entwicklung vorgestellt. Dabei versteht EIGNER unter der modellbasierten virtuellen PE die umfassende computerunterstützte Erstellung digitaler Modelle (inkl. aller Dokumentationen) mit einer formalen Sprache entlang der relevanten Produktlebenszyklusphasen mit dem Ziel der Weitergabe und Nutzung der digitalen Modelle in den einzelnen Entwicklungsphasen (inkl. für Simulation, Verifikation und Validierung), um frühzeitig das zu entwickelnde Produkt im Sinne einer holistischen Optimierung entlang des gesamten Produktlebenszyklus zu realisieren [154].

Daher werden im nachfolgend die Grundlagen für drei ausgewählte domänenübergreifende Modellierungsansätze für die virtuelle bzw. digitale Entwicklung vorgestellt. Zuerst wird in Kapitel 3.2.1 Systems Engineering mit dem Fokus auf das modellbasierte Systems Engineering erläutert. Darauf folgt ein Überblick über Ontologien in der Entwicklung in Kapitel 3.2.2. In Kapitel 3.2.3 steht schließlich die Erläuterung der matrixbasierten Modelle mit der Design Structure Matrix Methodik im Mittelpunkt. Im Anhang sind zudem noch weitere, wesentliche Grundlagen überblicksartig zusammengefasst. So finden sich in Anhang A.17 Grundlagen zu ontologiebasierten Modellen und in Anhang A.18 Grundlagen zu matrixbasierten Modellen.

3.2.1 Modellbasiertes Systems Engineering

Systems Engineering (SE) ist gemäß der Definition des INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE) ein transdisziplinärer sowie integrativer Ansatz, um die erfolgreiche Erstellung, Nutzung und Einstellung von technischen Systemen zu ermöglichen, wobei Systemprinzipien und -konzepte (z. B. die drei Systemkonzepte gemäß ROPOHL, vgl. Kapitel 2.1.1) sowie wissenschaftliche, technologische und Management-Methoden verwendet werden [274]. Dabei weist INCOSE darauf hin, dass der Begriff „technisch“ im weitesten Sinne verwendet wird, was zur Folge hat, dass technische Systeme aus grundsätzlich allen möglichen Elementen, wie bspw. Produkte, Prozesse, Wissen/Informationen, Organisationen und Dienstleistungen, bestehen können [274]. Nach HABERFELLNER ET AL. sind die beiden wichtigen Prinzipien des SE das Systemdenken (vgl. Kapitel 2.1) und das Vorgehensmodell des SE, welches auf den vier folgenden Grundprinzipien basiert [239]:

- Modelliere vom Abstrakten zum Konkreten und nicht umgekehrt
- Denke in Varianten, gebe dich nicht mit nur einer Variante (i. d. R. die erste verfügbare) zufrieden, sondern suche Alternativen
- Strukturiere den Prozess der Systementwicklung und -implementierung entsprechend zeitlichen Aspekten (stufenweiser Ansatz)
- Verwende eine Art Arbeitslogik als formale Handlungsanleitung bei der Problemlösung, unabhängig von deren Art und der Phase, in der diese auftreten

Für tiefergehende **Erläuterungen zu SE** sei auf die einschlägige Literatur verwiesen. So wird SE als methodischer Ansatz bei der Problemlösung bzgl. der SE-Prinzipien, der Systemgestaltung (insb. Architektur- und Konzeptgestaltung) und des Projektmanagements in [238], [239] ausführlich erläutert. Das Systems Engineering Handbuch von INCOSE [273], welches von

der Gesellschaft für Systems Engineering (GfSE) in [215] ins Deutsche übersetzt wurde, dient v. a. als Leitfaden für die wesentlichen Prozesse und Aktivitäten, die von (System-)Ingenieuren über den Lebenszyklus eines Systems durchzuführen sind.

Das **modellbasierte Systems Engineering** (engl. model-based SE, kurz: MBSE) ist eine Teildisziplin des SE, die sich anstatt mit einer heterogenen dokumentbasierten Entwicklung mit einer homogenen und vernetzten modellbasierten Entwicklung beschäftigt und im SE an Bedeutung gewinnt (s. u. a. [193] und Abgrenzung zu Model-Based Engineering in [421]). MBSE ist nach INCOSE die formalisierte Anwendung der Modellierung zur Unterstützung von Systemanforderungen, Entwurfs-, Analyse-, Verifikations- und Validierungsaktivitäten über den gesamten Lebenszyklus von der Entwicklung über die Herstellung bis zum Betrieb und Support [271]. Zudem ist MBSE ein Teil eines langfristigen Trends hin zu modellzentrierten Ansätzen, die u. a. von Ingenieursdisziplinen, wie z. B. Mechanik, Elektrik und Software, übernommen werden [271]. Diese formale Systemmodellierung soll gemäß INCOSE der Standard für die Spezifizierung, die Analyse, den Entwurf und die Verifizierung von Systemen werden und vollständig mit anderen technischen Modellen integriert sein [272]. Dabei sind Systemmodelle an die jeweilige Anwendungsdomäne angepasst und umfassen ein breites Spektrum an Modellen zur Darstellung aller Aspekte von Systemen [272]. Nach BURSAC ist ein Ziel des MBSE Modelle der einzelnen Anwendungsdomänen in einem übergreifenden Meta-Modell (vgl. Anhang A.15) zu vereinigen [99]. Ein mathematischer Formalismus für das MBSE wurde bereits 1993 von WYMORE in [620] vorgestellt.

Das modellbasierte SE hat zwar an Akzeptanz und Verbreitung gewonnen, befindet sich aber (selbst) noch in einem frühen Entwicklungsstadium. Zentrale (zukünftige) **Ziele von MBSE** sind nach INCOSE u. a. [272], [271]:

- Domänenspezifische Modellierungssprachen (vgl. Anhang A.12 und A.14) und Visualisierung, die es Systemingenieuren ermöglichen, sich auf die Modellierung der Benutzerdomäne zu konzentrieren
- Modellierungsstandards, die auf einer soliden mathematischen Grundlage beruhen und die sog. High-Fidelity-Simulation sowie Repräsentation der realen Welt unterstützen
- Modellbasierte Visualisierungen, die eine nahtlose Navigation in holistischen Modellen zwischen verschiedenen Perspektiven und Systemhierarchien ermöglichen
- Standards, die das Management und die Integration von Modellen über ein dezentrales Modell-Repository hinweg unterstützen

- Umfangreiche Wiederverwendung von Modellbibliotheken, Taxonomien und Entwurfsmustern
- Zuverlässiger, sicherer Datenaustausch über offene Schnittstellen.

Mit Bezug zu MBSE sind der Literatur weitere Erläuterungen (s. [132], [193], [167], [271], [272]), Nutzenbetrachtungen (s. [364]) und Lehrkonzepte (s. [373]) zu entnehmen. **MBSE-Literatur** mit Bezug zur PE ist u. a. in [12], [42], [320], [472], [556], [616] sowie auch mit Bezug zu RLFP in [304], [138] zu finden.

Demzufolge ist das Ziel von MBSE die Aktivitäten von SE mit Hilfe eines holistischen Modells zu unterstützen. Das Systemmodell setzt sich aus den miteinander in Beziehung stehenden Modellelementen zusammen, welche wesentliche Aspekte des Systems darstellen (v. a. Anforderungen, Parameter, Struktur, Verhalten) [193]. Eine Übersicht verschiedener Methoden zu MBSE ist in [167] zusammengestellt. Modellierungsmethoden, wie z. B. Systems Modeling Toolbox (SysMod) [597] oder die Object-Oriented Systems-Engineering Method (OOSE) [285], unterstützen Entwickler bei der korrekten Anwendung der Modellierungssprache, um konsistente und regelkonforme Modelle zu erstellen (s. auch Anhang A.14). Die Modellerstellung erfolgt mit einer Modellierungssprache, wie z. B. der **Systems Modeling Language** (SysML) oder Object-Process Methodology (OPM) [132]. Während mit der SysML-Sprache bis zu neun Diagrammartentypen verwendet werden, die unabhängig voneinander abgeleitet werden und möglicherweise nicht vollständig konsistent sind, kommt bei dem OPM-Ansatz nur ein einziges Diagramm zur Anwendung [132]. Aufgrund der weiteren Verbreitung von SysML wird folgend diese Modellierungssprache vorgestellt.

Die insb. aus der Softwareentwicklung bekannte Modellierungssprache **Unified Modeling Language** (UML) stellt die Grundlage der SysML dar. Anfang 2007 entstand diese mit dem Ziel eine durch IT-Werkzeuge unterstützte Standardsprache für SE darzustellen [285] und wurde als offizieller Standard 2007 in der Version 1.0 durch die Object Management Group (OMG) beschlossen und veröffentlicht [14]. Die SysML verwendet viele Konzepte der UML, lässt einige auch explizit weg (z. B. aufgrund deren hohen Spezifität für die Softwareentwicklung) und fügt einige zudem auch hinzu (z. B. aufgrund deren hoher Eignung zur Entwicklung technischer Systeme) [14]. Demzufolge wird das UML-Metamodell so angepasst, dass dadurch die SysML-Sprache entsteht (vgl. [14] für Details und für Grundlagen zu Metamodellen Anhang A.15).

Die aus der UML bekannten **Grundkonzepte der Objektorientierung** werden auch in der SysML angewendet (s. [14] für Details). So wird der

Begriff der Klasse verwendet, um z. B. Dinge, Personen, Daten mit gleichen Eigenschaften zu klassifizieren. Eine Klasse hat Eigenschaften (sog. Attribute) und besteht i. d. R. aus Objekten (d. h. Objekt als Instanz einer Klasse). Mit dem Konzept der Vererbung werden allgemeinere Eigenschaften in übergeordnete Klassen ausgelagert und – i. S. v. Eltern-Kind-Beziehung – an untergeordnete Klassen vererbt. Weitere wesentliche Begriffe sind u. a. Parts, Blöcke und Properties, die in [14] erklärt werden.

Das Prinzip der **Trennung von Modell und Sicht** liegt auch der SysML zugrunde [14] (s. Bild 18). Dies bedeutet zum einen, dass ein sog. Repository das gesamte Modell enthält (z. B. Datenbank als Speicher für das Modellrepository), und zum anderen, dass es mehrere Sichten auf das Modell gibt (i. S. v. Ausschnitte bzw. Projektionen des Modells), die in SysML als Diagramme bezeichnet werden [14]. Der Modellanwender verwendet die für eine bestimmte Fragestellung geeignete Sicht, welche eine Projektion der relevanten Modellelemente aus dem Repository darstellt (vgl. [14]). Wie im Bild 18 gezeigt, kann dasselbe Element mehrfach in unterschiedliche Sichten projiziert werden, für welche auch SysML-Diagramme mit entsprechenden Notationen verwendet werden können. Dies ist z. B. hilfreich, um spezifische Sichten (z. B. Produkt-, Prozesssicht) für unterschiedliche Entwickler (z. B. Produkt-, Prozessingenieure) zu visualisieren oder verschiedene Fragestellungen mit Hilfe unterschiedlicher Sichten (Diagramme) zu beantworten [14]. Aufgrund eines konsistenten Gesamtmodells im Repository haben Änderungen in einer Sicht eine Änderung im gesamten Modell zur Folge. Dies hat zwar zum einen den Vorteil, dass das Gesamtmodell konsistent bleibt, aber zum anderen auch den Nachteil, dass die Erstellung von Modellen komplexer wird, weil lediglich konfliktfreie Modifikationen vorgenommen werden können [11].

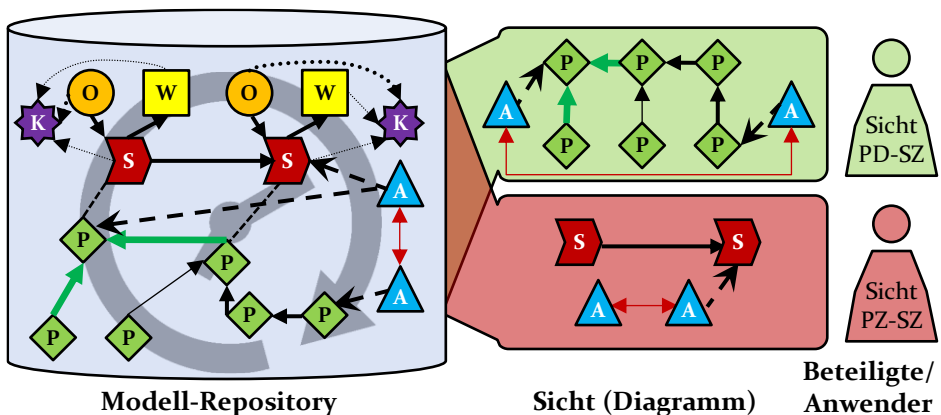


Bild 18: Trennung von Sicht und Modell (in Anlehnung an [14] und [99])

Die SysML differenziert zwischen neun **Arten von Diagrammen**. Diese legen die Sicht auf das Modell fest und bilden daher die Schnittstelle zwischen dem Modell und dem Modellanwender. In Bild 19 werden die SysML-Diagramme überblicksartig dargestellt. Dabei wird deutlich, dass den SysML-Diagrammen die UML-Diagramme zugrunde liegen und diese – analog zu den bereits angesprochenen Modellelementen – zum Teil identisch wie bei UML, modifiziert, neu hinzugefügt oder ausgeschlossen werden (in der UML stehen 14 Arten von Diagrammen zur Verfügung [465]). Die neun Diagrammarten der SysML lassen sich in die folgenden drei Kategorien einteilen [14]:

- Strukturelle Diagramme dienen der Darstellung von statischen Aspekten des Systems oder des Systemmodells und werden daher v. a. für die Modellierung der Systemarchitektur verwendet,
- Verhaltensdiagramme dienen der Darstellung von dynamischen Aspekten und werden daher v. a. für die Modellierung des Systemverhaltens eingesetzt,
- Das neu hinzugefügte Anforderungsdiagramm dient zur Darstellung aller Aspekte in Bezug zu den Systemanforderungen.

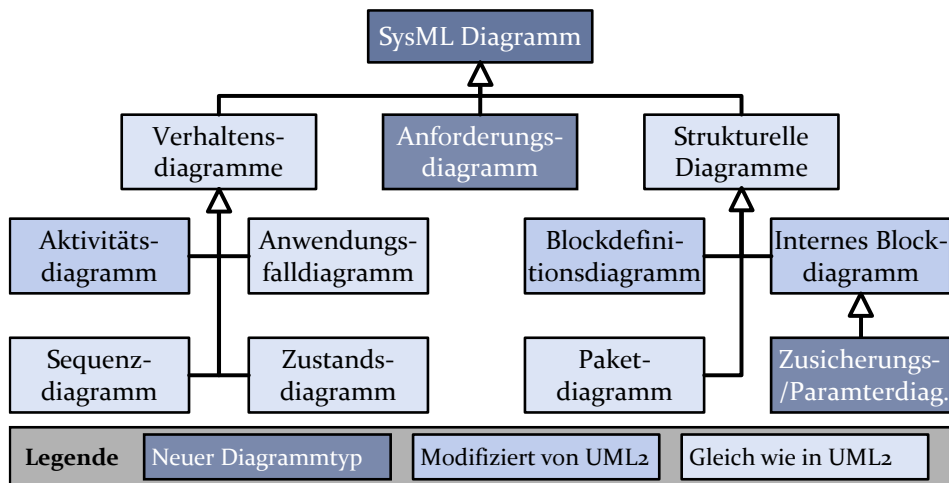


Bild 19: Überblick der Diagrammarten in SysML (nach [418])

Diese Diagrammarten werden mit weiteren Erläuterungen zu Grundlagen der SysML sowie zur praktischen Anwendung in [14] beschrieben. Eine sehr umfassende Beschreibung von SysML ist in der nationalen Literatur von WEILKIENS u. a. in [596], [597] und bspw. [319] sowie in der internationalen Literatur insb. im SysML-Standard selbst (aktuell in [418]) sowie in [193] und [132] zu finden. Für mit SysML modellierte Beispiele aus der PE sei auch auf die betreute studentische Arbeit [S33] verwiesen.

Im Rahmen der Systementwicklung kann SysML sowohl zur Dokumentation für das Re-Engineering bereits bestehender Systeme als auch zur entwicklungsbegleitenden Unterstützung für das Entwickeln neuer Systeme eingesetzt werden [14]. Dabei ist der **Einsatz von SysML** während der Entwicklung vorteilhafter, da dadurch Modelle iterativ weiterentwickelt und dabei die erforderliche Entwicklungsdokumentation erstellt wird, welche zudem als einheitliche Diskussionsgrundlage dient [14].

3.2.2 Ontologiebasierte Modelle

Ein zweiter interdisziplinärer und domänenübergreifender Modellierungsansatz im Rahmen der virtuellen, digitalen Entwicklung sind Ontologien. Der Begriff **Ontologie** setzt sich aus den griechischen Worten *on* (seiend) und *logos* (Lehre) zusammen, stammt aus der Philosophie und dient dort zur Beschreibung der Existenz des Seins [199]. Zudem wird der Begriff auch in der Informatik und künstlichen Intelligenz seit Beginn der 1990er Jahre verwendet, wobei Ontologien v. a. in den Themenfeldern Wissensmodellierung, -repräsentation und Computerlinguistik erforscht werden [235].

Ontologien dienen v. a. zur formalen Modellierung eines Systems mit Elementen bzw. Objekten, deren Relationen und dessen Struktur, welche für eine bestimmte Betrachtung von Bedeutung sind [233]. Somit werden Ontologien insb. zur Formalisierung von Wissen spezifischer Wissensbereiche eingesetzt, um dieses Wissen für Computer verarbeitbar und analysierbar zu machen. Diese Formalisierung erfolgt mittels Domänen, welche die Begriffe, Konzepte, Regeln, und Beziehungen für eine Formalisierung in maschinenlesbarer Form festlegen [199]. Daher ist nach GRUBER eine Ontologie als eine formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung definiert [229]. Dabei stellt eine Konzeptualisierung eine Menge informeller Regeln dar, welche die Struktur eines Realitätsausschnitts einschränken und von einer Person verwendet werden, um die relevanten Elemente (Objekte) und Relationen des Realitätsausschnitts abzugrenzen und zu abstrahieren [231]. Demnach ist eine Konzeptualisierung eine abstrakte, vereinfachte Sicht auf einen abzubildenden Ausschnitt der realen Welt [228]. Neben dieser oft zitierten und verbreiteten **Begriffsdefinition von Ontologie** nach GRUBER [229] (z. B. in [235], [101], [604]) lassen sich in der Literatur zahlreiche weitere Begriffsdefinitionen finden (s. Übersicht in [536]). Eine in [199] als nützlich empfundene und an einem Beispiel dargestellte Definition von Ontologie in der Informatik gemäß SOWA definiert Ontologie als eine formale Beschreibung des Wissens in einer Domäne (i. S. v. Realitätsausschnitt, vgl. Anhang A.12) in der Form von Konzepten

der Domäne, deren Relationen untereinander und der Eigenschaften dieser Konzepte und Beziehungen, sowie der in der Domäne gültigen Prinzipien und Axiome [516] (übersetzt nach [199]). Nicht nur innerhalb der Informatik gibt es eine Vielzahl von Erläuterungen des Begriffs, sondern auch in weiteren Disziplinen, wie der Neuroinformatik und -biologie, Psychologie sowie im Knowledge Engineering und den Ingenieurwissenschaften [101].

Der **Zweck des Einsatzes von Ontologien** ist v. a. die Verbesserung der Kommunikation innerhalb einer bestimmten (Anwendungs-)Domäne. Hinsichtlich der Kommunikationsbeziehung zwischen maschinellen und menschlichen Akteuren wird prinzipiell zwischen Computeranwendungen untereinander, zwischen Computeranwendungen und Individuen sowie zwischen Individuen untereinander differenziert.

Diese Akteure befinden sich nach STAAB in einer Kommunikationssituation, deren Eigenschaften über das **semiotische Dreieck** (s. Bild 20, vgl. semiotisches Rechteck in Anhang A.12) veranschaulicht werden können, welches die Beziehung zwischen den Symbolen, den Begriffen und den Dingen eines Realitätsausschnitts abbildet [520]. Grundlegende Definitionen zur Semiotik sind bereits in der Publikation [413] von OGDEN & RICHARDS zu finden und werden von verschiedenen Autoren (s. [482], [520], [552], [200]) sowohl weiterentwickelt als auch mit unterschiedlichen Begrifflichkeiten erläutern. Daher werden in Bild 20 zu den drei nach STAAB bezeichneten drei Ecken des semiotischen Dreiecks jeweils Alternativen (alt.) genannt.

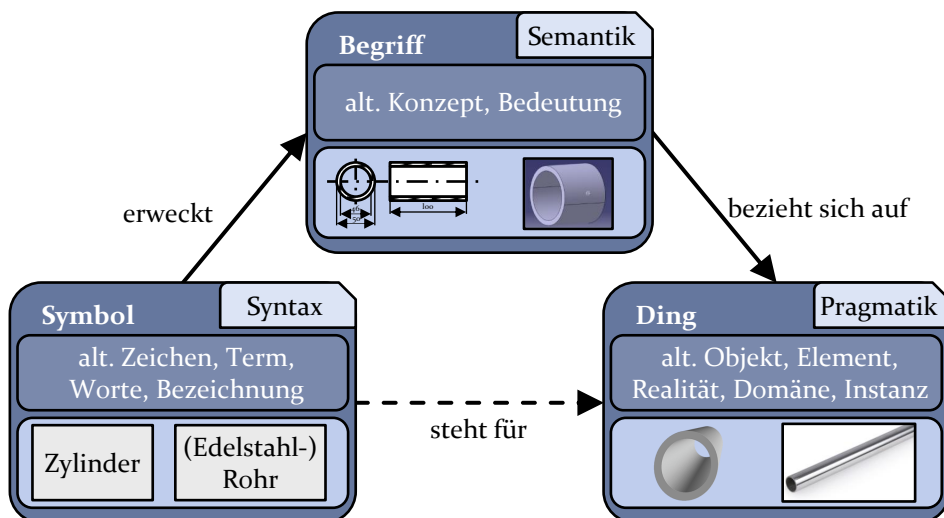


Bild 20: Semiotisches Dreieck mit Alternativbegriffen und Beispielen

Zudem sind die **Begriffe Syntax, Semantik und Pragmatik** (s. Anhang A.14 und A.15) nach [524], [482] und [200] in Bild 20 dargestellt. Während bei der Syntax der Zusammenhang der Symbole bzw. Zeichen (formal und strukturell) zwischen den Abbildungen im Zentrum steht, ist bei der Semantik die Bedeutung eines Begriffs zum Realitätsausschnitt im Fokus. Aufbauend auf der Syntax und Semantik beschreibt die Pragmatik die Verbindung zwischen dem Symbol und dem Ding (d. h. dem Element bzw. Objekt des Realitätsausschnitts) und den daraus abzuleitenden Aktionen für den Akteur infolgedessen Interpretation für seine spezifische Situation oder Fragestellung. Somit wird mit der Pragmatik die Semantik eines Begriffs eng mit dem spezifischen Kontext des Interpretierenden assoziiert, wodurch die Subjektivität von der Semantik zur Pragmatik steigt (s. [524]).

Zwischen Symbol, Begriff und Ding im semiotischen Dreieck (vgl. Bild 20) bestehen Interaktionen. Dabei werden Symbole zur Informationsübertragung verwendet, wobei die die Begriffsbedeutung sowie das Ding (i. S. v. Objekt bzw. Element) nicht umfassend erfasst werden können. Dies liegt insb. daran, dass eine Korrespondenz zwischen Symbol, Begriff und Ding aus einer **Vielzahl möglicher Korrespondenzen** durch den Empfänger der Nachricht festgelegt wird [363]. Die Auswahl einer Korrespondenz durch den Empfänger erfolgt aufgrund dessen Begriffsinterpretation und Erfahrung, so dass in dem in Bild 20 dargestellten Beispiel ein im nicht ingenieurwissenschaftlichen Kontext stehender Empfänger sich unter einem Zylinder bzw. einem (Edelstahl-)Rohr auch einen Zylinderhut bzw. – eher unwahrscheinlich – die Gemeinde Rohr bei Schwabach in der Nähe von Nürnberg vorstellen könnte.

Das zentrale **Ziel einer Ontologie** ist das Festlegen möglicher Korrespondenzen zwischen Symbolen, Zeichen und Worten auf der einen Seite und Dingen, Objekten und Elementen auf der anderen Seite, welche ein Empfänger einer Nachricht als gültig interpretieren kann [145]. Dazu wird eine Ontologie durch zusammengesetzte Vokabeln und einer Menge von logischen Aussagen zur relevanten Anwendungsdomäne ausgedrückt. Mittels der logischen Theorie der Ontologie werden die Beziehungen zwischen den Symbolen spezifiziert und somit die Menge möglicher Interpretationen für Symbole und den entsprechenden Relationen begrenzt. Sofern sich Sender und Empfänger einer Korrespondenz auf die Anwendung einer bestimmten Ontologie verständigt haben, besteht im Idealfall für jedes Symbol bzw. Wort im Vokabular (bestenfalls) genau eine Korrespondenz mit einem Begriff bzw. Ding des Realitätsausschnitts [363]. Dadurch wird die Anzahl möglicher Korrespondenzen reduziert und die Wahrscheinlichkeit erhöht,

dass Empfänger und Sender dasselbe Ding des Realitätsausschnitts bezeichnen [363].

Die **Nutzung von Ontologien in der PE** kann zur Externalisierung von Wissen über Elemente und deren Relationen eines Produktmodells als auch zur Beschreibung von z. B. Prozess- und Wissensdomänen genutzt werden. Dadurch lassen sich auch verteilte Produktmodelle miteinander integrieren und formal beschreiben (vgl. [241]). Somit kann das semiotische Dreieck in Bild 20 um die Ontologie, in der die Bedeutung (insb. der Begriffe) formalisiert wird, das Produkt (oder auch Prozess usw.) sowie das Produkt repräsentierende Produktmodell (oder auch Prozessmodell usw.) erweitert werden (s. Bild 21, vgl. auch [552]).

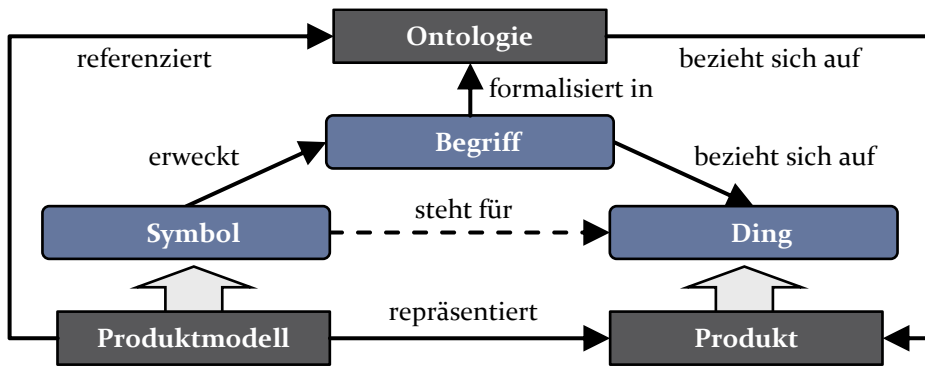


Bild 21: Erweitertes Semiotisches Dreieck (in Anlehnung an [552])

Zudem kann eine **Ontologie auch als Meta-Modell** (s. Anhang A.15) einer Wissensbasis dienen und so nach KOHN z. B. die für die Arbeit mit Produktmodellen relevanten Konzepte und deren Relationen definieren [313]. Diese in [313] entwickelte Wissensbasis unterstützt bei der Erfassung, Strukturierung und Bereitstellung von Wissen über die Arbeit mit Produktmodellen.

Eine Ontologie dient außerdem als Grundlage für ein in [604] erarbeitetes Konzept zur Planung des **Absicherungsprozesses von Produkteigenschaften** unter Berücksichtigung der verfügbaren Produkt- und Prozessinformationen sowie Methoden. Für die Modellierung des Konzepts und die prototypische Implementierung wird eine geeignete Architektur und ein Datenmodell der Ontologie vorgestellt [604].

Überdies bilden **Ontologien die Grundlage für die zielgerichtete Wiederverwendung** von Produkt- und Prozessmodellen im PEP [P5]. Dabei müssen die wiederzuverwendenden Produkt- und Prozessmodelle bei der Entwicklung neuer Produkte in deren digitalen Modellen (i. S. v. Wissens-

basen) modifiziert werden. Ein Konzept für eine ontologiebasierte Wissensbasis zur Unterstützung dieser Modifikationen ist in [P5] zu finden.

Ergänzende **Erläuterungen zu Ontologien** und insb. deren Eignung und Einsatz zur digitalen, modellbasierten Entwicklung sind in der zuvor zitierten Literatur sowie im Anhang A.17 zu finden. Zudem sei v. a. auf die drei betreuten studentischen Arbeiten [S2], [S13], [S15] verwiesen.

3.2.3 Matrixbasierte Modelle

Den dritten, domänenübergreifenden Ansatz zur Modellierung sowohl von Elementen aus diversen Domänen als auch deren Relationen in einem holistischen Modell stellen matrixbasierte Modelle dar (vgl. u. a. [355], [378], [161], [237]). **Da die matrixbasierten Modelle zum Komplexitätsmanagement in der PE im Fokus dieser Arbeit** stehen, werden diese Ansätze nachfolgend ausführlich erläutert.

Die ersten Wissenschaftler haben bereits in den 1960er Jahre matrixbasierte Ansätze zur Systemmodellierung vorgeschlagen (s. [355]). So hat STEWARD 1962 erste matrixbasierte Ansätze zur Analyse der Struktur großer Systeme von Gleichungen vorgestellt [530] und 1981 in seiner Publikation „Design Structure System: A Method for Managing the Design of Complex Systems“ die Grundlagen für die Design Structure Matrix (DSM) gelegt [531]. Seitdem haben die darauf aufbauenden und auch weitere **matrixbasierte Ansätze zunehmende Aufmerksamkeit und internationale Bedeutung in Wissenschaft und Industrie gewonnen** (vgl. z. B. [355], [326], [161]), um die Komplexität von u. a. technischen Systemen zu managen [355].

Matrixbasierte Modellierungsansätze, wie neben der im Fokus stehenden DSM-Methodik bspw. auch das House of Quality im Zusammenhang mit dem Quality Function Deployment, dienen v. a. der Modellierung, Analyse, Visualisierung und Bewertung von komplexen Systemen (vgl. Kapitel 2). So werden in unterschiedlichen Anwendungsfeldern (z. B. Ingenieurs- und Wirtschaftswissenschaften) komplexe Sachverhalte dargestellt, indem die Systemelemente und deren Relationen in Beziehung zueinander gesetzt werden (vgl. [161] für Beispiele). Grundsätzlich erfolgt die **Modellierung von Matrizen in einer matrixartigen Form**, in der die Elemente (Knoten) in Spalten und Zeilen angeordnet werden. Die Relationen (Kanten) zwischen den Elementen werden prinzipiell durch paarweisen Vergleich durch definierte Symbole (z. B. X wie in Bild 22 oder „1“ und „-1“) modelliert [355]. Da mittels einer binären Notation nur (Nicht-)Existenz einer Relation modelliert werden kann, werden häufig auch farbliche Symbole oder

Ziffern zur Darstellung der Stärke und Art der Relation verwendet (z. B. mit „+2“ kann ausgedrückt werden, dass mit steigendem Außendurchmesser des Rohrs dessen Volumen steigt, vgl. Anhang A.4). Zudem kann der Informationsgehalt durch Verwenden von Vektoren in den Matrixzellen erhöht werden (s. [531], [626]).

Im Folgenden werden die **drei elementaren Matrixarten**, die sich bzgl. ihres Zwecks differenzieren lassen, kurz vorgestellt (s. Bild 22). Die Design Structure Matrix (DSM) dient der Modellierung der Relationen zwischen Elementen derselben Domäne. Dagegen werden mit der Domain Mapping Matrix (DMM) die Relationen zwischen Elementen von zwei verschiedenen Domänen modelliert. Die Multi Domain Matrix (MDM) setzt sich aus der quadratischen Anordnung von mind. zwei DSMs und DMMs zusammen. Dabei ist zwischen den zwei Konventionen der Matrixleserichtung bzw. -beeinflussungsrichtung (i. S. v. Ursache-Wirkungs-Matrix) „inputs in rows“ (IR) und „inputs in columns“ (IC) zu unterscheiden [161]. In dieser Arbeit wird i. d. R. die erste und damit die europäische Konvention IR verwendet (d. h. das Zeilenelement beeinflusst das Spaltenelement). Den dargestellten Matrizen in Bild 22 liegt das Anwendungsbeispiel aus Kapitel 1.5 und exakt die in Bild 1 verwendeten Elemente zugrunde.

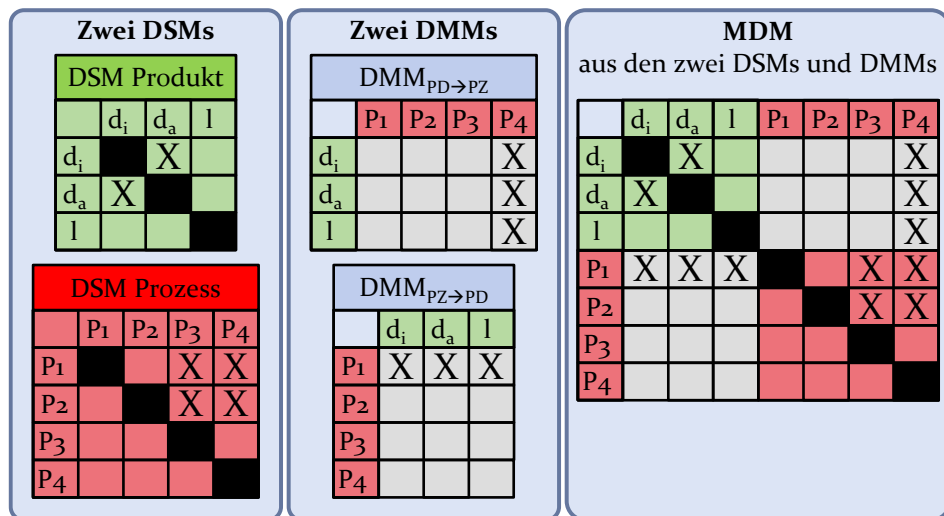


Bild 22: Vereinfachte MDM am Beispiel Rohr

Design Structure Matrix

In der DSM, die auch als Dependency Structure Matrix oder Intra-Domain Matrix bezeichnet wird, werden die **Relationen zwischen Elementen derselben Domäne** modelliert (z. B. in der Produktdomäne die Relation

zwischen dem Innen- und Außendurchmesser eines Rohrs, vgl. Bild 22) und somit ist eine DSM immer eine $n \times n$ Matrix. Da ein Element i. d. R. keine Relationen zu sich selbst hat, weist die Diagonale der DSM nach BROWNING keine Einträge auf [86] (in Bild 22 ausgeschwärzte Zellen).

Die **Kategorisierung von DSMs** erfolgt nach BROWNING mittels der zwei Hauptkategorien „Statische DSM“ und „Zeitbasierte DSM“, die wiederum in jeweils zwei Unterkategorien gegliedert werden [86] (s. Bild 23). Während statische DSMs Systemelemente umfassen, die sich im betrachteten Zeitraum nicht ändern, liegt der Fokus von zeitbasierten DSMs auf der zeitlich-logischen Reihenfolge von Elementen im betrachteten Zeitraum [86]. Innerhalb der statischen DSMs wird zwischen komponentenbasierten und organisationsbasierten DSMs differenziert. Dabei werden mit komponentenbasierten DSMs insb. Produktelemente mit deren Relationen modelliert. Die Organisationselemente und deren Struktur werden in der organisationsbasierten DSM dargestellt. Im Hinblick auf die zeitbasierten DSMs wird zwischen den aktivitätenbasierten DSMs und parameterbasierten DSMs unterschieden. Aktivitätenbasierte DSMs dienen zur Modellierung insb. von Prozessen und auch Wissens- und Informationsflüssen. Die parameterbasierten DSM werden u. a. zu Modellierung von Entscheidungen auf Grundlage relevanter Parameter verwendet. Aufgrund der Relevanz für diese Arbeit werden die vier DSM-Kategorien nachfolgend mit entsprechenden, relevanten Forschungsarbeiten skizziert. Zur Vertiefung der matrixbasierten Modellierung wird auf die einschlägige internationale Literatur (u. a. [161], [355], [326]) sowie auf die Publikation [88] von BROWNING aus dem Jahr 2016 mit einem umfassenden Überblick über DSM-Publikationen und für aktuelle Arbeiten auf die Publikationen der DSM-Konferenz (s. dsmweb.org, z. B. die Publikation u. a. des Autors in [P34]) verwiesen.

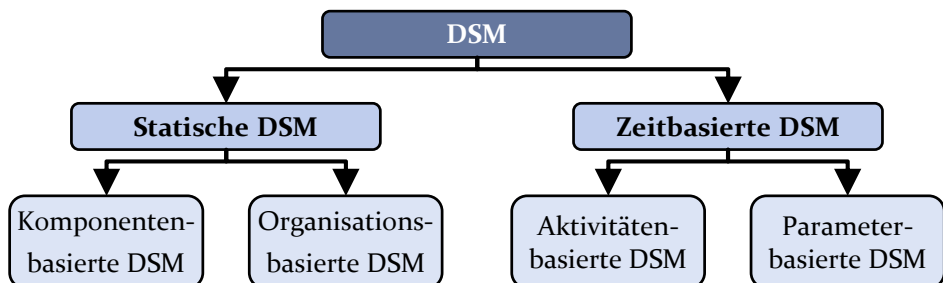


Bild 23: Kategorisierung von DSMs (nach [86])

Komponentenbasierte DSM

Die **Zielsetzung** der Verwendung komponentenbasierter DSMs ist es einen systematischen Überblick über den strukturellen Aufbau komplexer, technischer Systeme zu erhalten (s. [161]). Das sich daraus ergebende, tiefe Verständnis von Zusammenhängen über Hierarchieebenen hinweg unterstützt beim Komplexitätsmanagement (z. B. bei der Modularisierung). Zudem lassen sich die Auswirkungen in Folge von bspw. Modifikationen von Produktmerkmalen oder von Anforderungsänderungen abschätzen und handhaben.

Die **Modellierung** von komponentenbasierten DSMs erfolgt v. a. von Produktstrukturen (z. B. Relationen zwischen Produktelementen auf gleichen oder verschiedenen Produkthierarchieebenen, vgl. Anhang A.6). Dabei ist die Modellierung eines technischen Systems von dessen Komplexität (vgl. Komplexitätsgrad in Kapitel 2.2.2) sowie vom gewählten Abstraktionsgrad (s. Anhang A.12 und Bild 14) abhängig. Im Rahmen der DSM-Modellierung sind nach TANG ET AL. v. a. folgende Fragen zu beantworten [547]: Was sind die Elemente des Produktes? Wo (und wie) sind die Relationen zwischen den Elementen repräsentiert? Was ist die Quelle der Relationen? Warum stehen diese Elemente in dieser Art und Weise zueinander in Beziehung? Dabei sollen nach EPPINGER & BROWNING die Anzahl der modellierten Elemente zwischen 20 und 50 sein, so dass einerseits der erforderliche Modellumfang bzw. Detaillierungsgrad realisiert und andererseits der Aufwand der Matrixanalyse nicht übermäßig groß wird (s. für Details [161]).

Organisationsbasierte DSM

Das **Ziel** der organisationsbasierter DSMs ist die Darstellung des strukturellen Aufbaus und der Zusammenhänge von Organisationselementen. Je nach gewähltem Abstraktionsgrad (vgl. Bild 14 in Kapitel 2.3.2) stehen die Relationen zwischen bspw. Rollen, Stellen, Teams oder weitere Konstrukte einer Organisation im Mittelpunkt (s. [P34]).

Sofern z. B. mit einem X-Symbol die (disziplinarische) Weisungsbefugnis zwischen Organisationselementen abgebildet wird, kann aus einer organisationsbasierten DSM eine hierarchische Darstellung mittels planarer Graphen (s. Kapitel 3.3.2) „per Knopfdruck“ abgeleitet werden (s. für Details [161]). Dafür ist während der **Modellierung** von organisationsbasierten DSMs die Abfolge der Organisationselemente in den Zeilen und Spalten essenziell und daher zu beachten, da der sog. Wurzelknoten (vgl. Baumgraphen) das Element in der ersten Zeile darstellt. Ausgehend vom Wurzelknoten sind die Relationen zu (hierarchischen) Nachfolgeknoten usw.

zu modellieren (s. [555]). Neben dieser Weisungsbefugnis-Relation lassen sich auch viele weitere Relationen abbilden. So verwendet z. B. KASPAREK die folgenden Ausdrücke zur Modellierung verschiedener Relationstypen: besteht aus, gehört zu, kommuniziert mit, leitet, verantwortlich für [295].

Aktivitätenbasierte DSM

Das **Ziel** der aktivitätsbasierten DSMs ist die Abbildung von zeitabhängigen Prozesselementen in ihrer zeitlich-logischen Abfolge. Damit basieren diese DSM bspw. auf den Aktivitäten und Relationen des Material- oder Informationsflusses mit entsprechenden In- und Outputs (vgl. drei Hauptsatzarten in Anhang A.6). Hierfür können je nach Abstraktionsgrad (s. Kapitel 2.3.2) verschiedene Prozesshierarchieebenen Gegenstand der Modellierung sein, wie z. B. kleine Aufgaben und Mikrozyklen über Teilprozesse bis hin zu funktionsübergreifenden Unternehmensprozesse und Lebenszyklusprozesse (s. Kapitel 3.1.2 für Beispiele).

Bei der **Modellierung** der aktivitätsbasierten DSMs ist es hilfreich darauf zu achten, dass die Aufeinanderfolge der Zeilen bzw. Spalten den prinzipiellen, sequenziellen Prozessablauf darstellen (sollten) [87]. Bei der DSM-Erstellung ist nach TANG ET AL. insb. die folgende Frage zu beantworten [547]: Wie können die Produktelemente mit Hilfe der in Relation stehenden Prozesselementen entwickelt und hergestellt werden? Zur Modellierung komplexer Prozesse stehen nach BROWNING die fünf folgenden und in Bild 24 dargestellten Arten von Relationen zur Verfügung [87] (vgl. [161]): Sequenzielle, parallele, gekoppelte (d. h. sich gegenseitig Input gebend), bedingte (d. h. Verwendung der Konnektoren „AND“, „OR“ oder „XOR“) und iterative (i. S. v. Feedback-Schleife) Relationen.

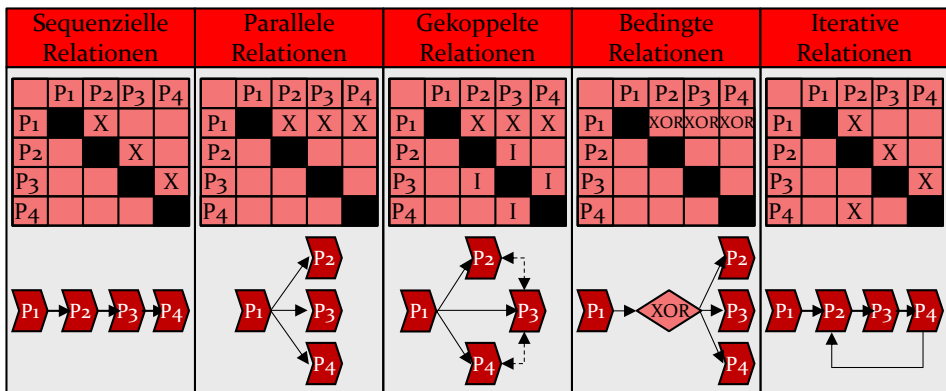


Bild 24: Veranschaulichung der fünf Arten von Relationen der aktivitätenbasierten DSMs (in Anlehnung an [87])

Parameterbasierte DSM

Die **Zielsetzung** der parameterbasierten DSMs ist die Modellierung des Netzwerks von Entwurfparameter-Entscheidungen (i. S. v. Festlegung von Produktmerkmalen, um Produkteigenschaften zu realisieren) als ein Netzwerk von PEP-Aktivitäten, von denen jede einen oder z. T. auch mehrere Entwurfparameter bestimmt [161]. Daher dienen parameterbasierte DSMs zur Modellierung und Analyse von Wechselwirkungen von Parameterentscheidungen im Rahmen der PE.

Bei der **Modellierung** von parameterbasierten DSMs stehen die Struktur und die Relationen von Parameterentscheidungen auf der untersten PEP-Ebene im Fokus. Dabei wird unter Parameterentscheidungen v. a. die Festlegung von (physikalischen) Merkmalen zur Realisierung von Produkteigenschaften und -verhalten eines technischen Systems gemeint. Unterschiede zwischen aktivitätenbasierter und parameterbasierter DSMs sind zum einen, dass die parameterbasierten (i. S. v. bottom-up) im Vergleich zu den aktivitätsbasierten (i. S. v. top-down) DSMs auf niedrigerer Prozess- und Produkthierarchieebene verwendet werden, und zum anderen, dass die Abbildung in parameterbasierten DSMs die Relationen zwischen Produktverhalten und -eigenschaften bestimmenden Parametern bzw. Merkmalen definieren und aktivitätsbasierte DSMs v. a. Prozessabfolgen und Vorgehensweisen festlegen (s. für sehr ausführliche und auch beispielhafte Erläuterungen [62] sowie [426] und [427]).

Domain Mapping Matrix

In der DMM, die auch als Inter-Domain Matrix bezeichnet wird, werden die **Relationen zwischen Elementen aus unterschiedlichen Domänen** modelliert. Als ein Beispiel dient die Relation zwischen einem Element der Produktdomäne und der Prozessdomäne, wie bspw. zwischen dem Rohrinwendurchmesser d_i und dem Schritt „Festigkeitsnachweis durchführen“ P_4 für ein Rohr (s. Bild 22 obere DMM). Ein zweites Beispiel ist die Relation zwischen dem Innen- und Außendurchmesser eines Rohrs aus der Produktdomäne mit dem Schritt „3D-CAD-Modell modellieren“ P_1 aus der Prozessdomäne (s. Bild 22 untere DMM). Somit ist eine DMM meistens eine $n \times m$ Matrix, in welche Einträge in der Diagonalen sinnvoll sind.

Die Analyse der Relationen zwischen Elementen verschiedener Domänen auf Grundlage von DMMs, wie bspw. der Produkt- und Prozessdomäne, hat viele **Nutzenpotentiale** (s. [88] für Details). So werden bspw. durch die Verwendung von DMMs die Auswirkungen von Änderungen eines bestimmten Elements in einer Domäne auf die Elemente der anderen

Domäne(n) transparent dargestellt (z. B. hat die Produktmodularisierung auch immer Änderungen von Prozessen zur Folge). So wird bspw. in der Publikation des Autors [P14] gezeigt, dass mit Hilfe der DMMs zwischen den drei Domänen Prozess, Dokument (i. S. v. Wissen, Informationen) und Organisation auch der Informationsfluss im PLM verbessert werden kann.

Multiple Domain Matrix

Die MDM ist eine $n \times n$ Matrix und setzt sich aus mind. zwei DSMs und den entsprechenden DMMs zusammen (vgl. Bild 22). Dabei sind die unterschiedlichen Domänen und auch deren Elemente in derselben Reihenfolge in den Zeilen als auch in den Spalten angeordnet, so dass die DSMs die Diagonale einer MDM darstellen. Sofern die DMMs ober- und unterhalb der Diagonalen identisch (i. S. v. transponiert) sind, ist nur eine der beiden Dreiecksmatrizen von Interesse [376].

Die Analyse der MDM bietet viele **Nutzenpotentiale**, wie v. a. die Basis für eine ganzheitliche, zielgerichtete und domänenübergreifende Verbesserung im Rahmen der PE (vgl. [153], [316] und v. a. [88]). Daher werden auch nach BROWNING MDM für vielfältige Aufgaben- und Problemstellungen in der PE eingesetzt, wie z. B. als Bewertungsgrundlage für Entscheidungen, zum Identifizieren von Produkt- und Projektrisiken sowie auch zum Variantenmanagement [88].

3.3 Ansätze für das matrixbasierte Komplexitätsmanagement in der Produktentwicklung

Da sich matrixbasierte Modelle (s. Kapitel 3.2.3) gut für das Komplexitätsmanagement in der PE eignen und daher im Mittelpunkt dieser Arbeit stehen, werden aufgrund ihrer Relevanz für diese Arbeit **ausgewählte Ansätze für das matrixbasierte Komplexitätsmanagement** in den folgenden Kapiteln kurz vorgestellt. So erfolgt in Kapitel 3.3.1 ein Überblick über verschiedene matrixbasierte Modellierungsansätze. Relevante Ansätze zur Visualisierung von Modellen komplexer Systeme, insb. auf Grundlage der Graphen- und Netzwerktheorie, werden in Kapitel 3.3.2. Eine Übersicht wesentlicher Ansätze zur mathematischen Analyse dieser Modelle wird in Kapitel 3.3.3 vorgestellt. Aufbauend auf den Modellanalysen und -visualisierungen können ausgewählte Ansätze zur multikriteriellen Bewertung und Entscheidungsfindung eingesetzt werden, die z. B. in LUFT ET AL. beschrieben sind [P25] (insb. multiattributive Wertfunktionen). Abschließend werden in Kapitel 3.3.4 einige IT-Werkzeuge zur Modellierung, Analyse und Visualisierung komplexer Systeme in der PE beleuchtet.

3.3.1 Matrixbasierte Modellierungsansätze

In diesem Kapitel werden **Ansätze zur matrixbasierten Modellierung** komplexer Systeme in der PE vorgestellt. Hierbei wird auf die Ansätze, die für diese Arbeit besonders wichtig sind, eingegangen. In den vom Autor dieser Arbeit betreuten studentischen Arbeiten sowie auch in der einschlägigen Literatur sind weitere Ansätze zu finden und ausführlicher erläutert.

Modell zur Bewertung von Risiken und Auswirkungen von Anforderungsänderungen (s. [197]). Ähnlich zur mechanischen und mechatronischen PE ist es auch im Rahmen reiner Softwareentwicklungsprojekte essenziell, Anforderungsänderungen und deren potenziellen Risiken und Auswirkungen zu beherrschen. Dafür schlagen FU ET AL. ein auf der DSM-Methodik basierende probabilistisches **Modell zur Bewertung von Risiken und Auswirkungen von Anforderungsänderungen** in der Softwareentwicklung vor (s. Bild 25). Darin werden sowohl zur Modellierung der Änderungsfortpflanzung als auch zur Risikoberechnung DSMs eingesetzt. Dabei werden die gewünschten Anforderungen in einem Anforderungsraum und die zur Erfüllung dieser benötigten Elemente der Softwarearchitektur in einem Lösungsraum modelliert (vgl. Bild 14). In Folge von Anforderungsmodifikationen durch einen Stakeholder sind auch die damit in Relation stehenden Softwareelemente zu ändern (s. Bild 25).

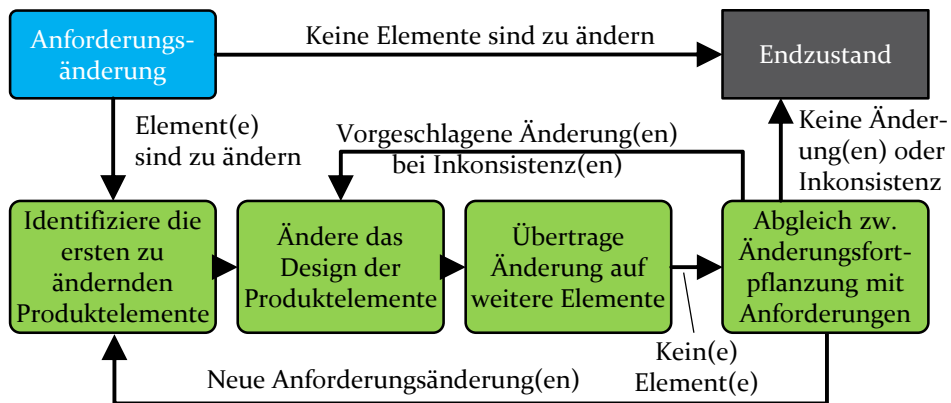


Bild 25: Schematische Darstellung des Änderungsfortpflanzungsprozesses (in Anlehnung an [197])

Da auch zwischen den Architekturelementen i. d. R. viele Relationen bestehen und die Softwarearchitektur konsistent sein muss, sind die in Folge der Änderungsfortpflanzung erforderlichen Modifikationen der entsprechenden Softwarearchitekturelemente in einer sinnvollen Reihenfolge anzupassen (s. Bild 25). Für eine bessere Abschätzung des Risikos von

Anforderungsänderungen werden die Relationen zwischen den Softwareelementen, die Fortpflanzungswahrscheinlichkeit und -stärke in DSMs abgebildet. Die erforderlichen Daten stammen sowohl aus expliziten Wissensquellen (z. B. Daten und Dokumenten) als auch aus impliziten Wissensquellen (z. B. Erfahrungen von Beteiligten). Dafür ist das implizite Wissen zu externalisieren.

Methode zur Modellierung von Parameterabhängigkeitsnetzwerken technischer Systeme (s. [259]). Auch das methodische Vorgehen von HIRAO ET AL. basiert auf der Modellierung von Elementen und deren Relationen mittels DSMs. Für diese Methode ist die Differenzierung zwischen folgenden fünf Elementarten wichtig. Dabei haben die ersten drei Elementarten Parameter, welche durch Constraints (i. S. v. z. B. Bedingungen) miteinander verknüpft sind und zusammen ein Netzwerk bilden (vgl. Bild 26 ohne Constraints). So sind ausgehend von den Kundenanforderungen die Systemverhaltensweisen (z. B. Beschleunigen) zu bestimmen und danach in Funktionen zu überführen und durch Strukturelemente (i. S. v. Produkt-elementen) zu realisieren. Diese drei Elemente, welche in [259] nicht präzise definiert sind, spezifizieren mindestens einen Parameter (s. Bild 26, z. B. das Systemverhalten „Beschleunigen“ den Parameter „Maximalgeschwindigkeit“). Die in Bild 26 nicht dargestellten Constraints resultieren aus der Bündelung von Relationen zwischen Parametern. Am Beispiel der (vereinfachten) Schraubenauslegung mit den drei Parameter „Lochtiefe“, „Lochradius“ und „Standardschraube“ ergibt sich das Constraint zwischen diesen drei Parametern i. S. v. „Schrauben in die Schraubenlöcher“.

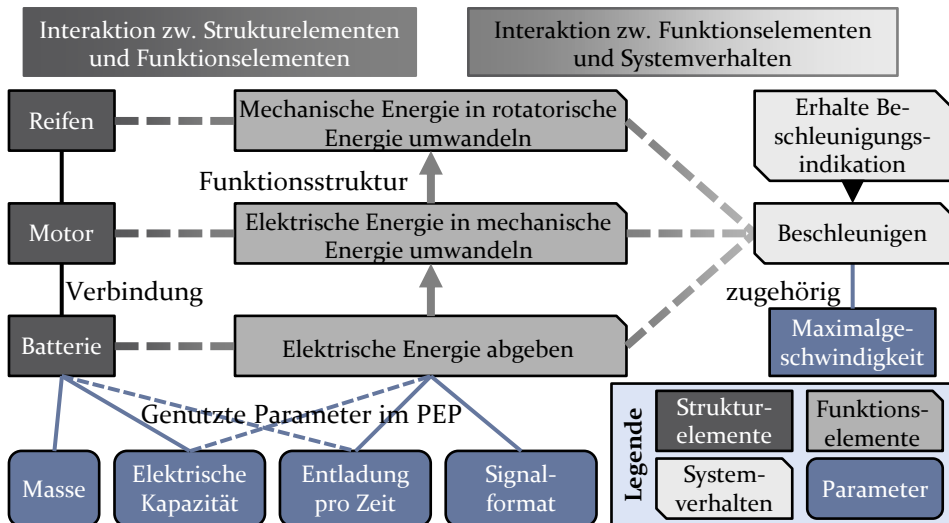


Bild 26: Darstellung der Elemente als Netzwerk (in Anlehnung an [259])

Auf Grundlage dieses Verständnisses haben HIRAO ET AL. das methodische **Vorgehen zur Modellierung des Parameterabhängigkeitsnetzwerks** erarbeitet (s. [259] für Details). Diese Vorgehensweise gliedert sich in die folgenden drei grundsätzlichen Schritte (s. Bild 27):

- Ermitteln der Relationen zwischen Parametern: Parameter werden in folgende 5 Kategorien gegliedert: Input, Output, Prozess, Verbindung und Attribut. Eine Software ermittelt eine Vorauswahl an Relationen, aus welcher Entwickler die passenden Relationen wählen;
- Ermitteln der Constraints: Constraints i. S. des Ansatzes werden aus den Abhängigkeitsbeziehungen zwischen Parametern abgeleitet. Die Software gruppiert Parameter automatisch zu Constraints, aus welchen Entwickler die geeigneten Constraints auswählen;
- Ermitteln der PEP-Schritte (s. [258] für Details): Auf Grundlage des nicht-gerichteten Netzwerks aus Parametern und Constraints werden mögliche PEP-Abfolge gesucht und evaluiert (v. a. bzgl. der drei Aspekte „Risiko von Korrekturarbeiten“, „Effizienz der Entscheidungsfindung“ und „Entscheidungsphase für wichtige Parameter“). Aus diesen Vorschlägen wählt der Entwickler eine PEP-Abfolge aus. Das Ergebnis von Schritt 3 ist somit ein gerichtetes Parameter-Constraint-Netzwerk (s. Bild 27).

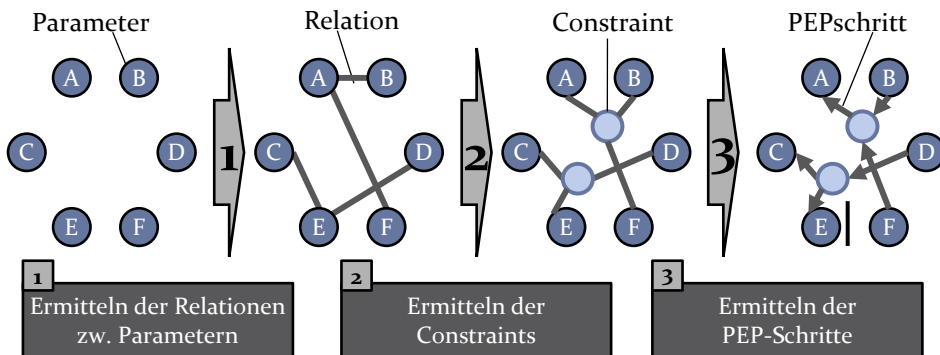


Bild 27: Vorgehen zur Modellierung von Parameterabhängigkeitsnetzwerken (in Anlehnung an [259])

Modellierungsansatz zur Multi-Solution eXtended Conceptual Design Semantic Matrix (MSX-CDSM) (s. [124]). Aufbauend auf teils eigenen, teils von Co-Autoren verfassten Publikationen (v. a. [112], [113], [114], [123]) integrieren DIAGNE ET AL. den objektorientierten Ansatz (insb. SysML-Diagramme, s. Bild 19) in den DSM-Modellierungsansatz zur konzeptionellen, semantischen Modellierung bspw. komplexer Produktfamilien von mechatronischen Produkten. Dabei ist die MSX-CDSM eine

Weiterentwicklung (i. S. v. Generalisierung) der X-CDSM und ermöglicht die Modellierung mehrerer Familien komplexer Produkte durch Verwendung mehrerer X-CDSMs. Dazu bildet jede X-CDSM eine Produktfamilie (d. h. eine Lösung) ab (s. Bild 28). Falls für jede X-CDSM die Werte eines jeden Elements und einer jeden Relation festgelegt sind, ist ein konkretes, komplexes Produkt definiert. Somit können aus einer MSX-CDSM mehrere Lösungen (d. h. komplexe Produktfamilien) und mehrere Instanzen jeder Lösung (Instanzen einer komplexen Produktfamilie) abgeleitet werden (vgl. Instanziierung, s. Bild 28).

Klasse der Komponenten	d_i	d_a	l	Kritikalität	Zuverlässigkeit
d_i	Klasse 1	r_{12}		k_1	z_1
d_a		Klasse 2	r_{13}	k_2	z_2
l			Klasse 3	k_3	z_3
Anzahl a der Relationen r	a_1	a_2	a_3	k_o (Schwellwert)	z_o (Schwellwert)

Bild 28: Schematische Darstellung der MSX-CDSM der Lösung 1 (in Anlehnung an [124])

In Bild 29 wird einerseits die schrittweise Erweiterung der komponentenbasierten DSM über die X-DSM und X-CDSM bis zur MSX-CDSM erläutert und andererseits auch die wesentlichen Unterschiede zwischen diesen DSMs deutlich. Dabei bietet MSX-CDSM v. a. den Vorteil, mehrere Familien komplexer Produkte und mehrere Varianten von jedem dieser Produkte in einem Modell bereits in der Konzeptphase darzustellen. In dieser Phase sind die Produktmerkmale und -eigenschaften noch nicht genau definiert und verschiedene Ausprägungen dieser können die Anforderungen und Randbedingungen in der Spezifikation erfüllen. Daher liegt der Fokus der MSX-CDSM auf der Konzeptphase und umfasst v. a. Klassen von Elementen, Klassen von Relationen, semantische Daten, Varianten eines Produkts (Instanzen) und Produktfamilien (Lösungen). Dabei wird das Ziel verfolgt, die Definition der Ausprägungen der Elemente und Relationen möglichst spät im PEP vorzunehmen. Somit unterstützt die generische **MSX-CDSM insb. die konzeptionelle, semantische Entwicklung** [124].

Zudem dient diese MSX-CDSM nach DIAGNE ET AL. als **Input für die geometrische 3D-Modellierung** mittels 3D-CAD-Systemen [124]. Da CAD-Systeme nur eingeschränkt die Möglichkeit bieten Daten zu erfassen, die Relationen zwischen Elementen (semantisch) zu beschreiben und die Modellierung von Produktfamilien zu unterstützen, stellen MSX-CDSM eine hilfreiche Ergänzung zu den 3D-CAD-Modellen in CAD-Systemen dar.

Komp.bas. DSM	X-DSM	X-CDSM	MSX-CDSM
▪ Ausprägungen der Elemente werden nicht abgebildet ▪ Zwischen 2 Elementen wird nur die Existenz einer Relation repräsentiert	▪ Ausprägungen der Elemente werden dargestellt, aber von jedem Element wird nur eine Instanz dargestellt ▪ Zwischen 2 Elementen wird nur der Relationstyp dargestellt	▪ Die Elemente werden als Objektklassen abgebildet ▪ Die Relationen werden als Objektklassen abgebildet ▪ Die Richtung der Relation R_{xy} ist von der K_x zur K_y	▪ Abbildung mehrerer X-CDSMs (i. S. v. verschiedene Produktfamilien) ▪ Generalisierung der X-CDSMs

Bild 29: Von der komponentenbasierten DSM zur MSX-CDSM (Inhalte aus [124])

Ansatz für eine „hoch aufgelöste“ (engl. high-definition) DSM (HDDSM) für die quantitative Bewertung der Produktarchitektur (s. [554]): Im Mittelpunkt des HDDSM-Ansatzes steht ein Produktrepräsentationsmodell auf Grundlage von DSMs mit dem Ziel, möglichst viele und sehr detaillierte Relationen zwischen Komponenten eines Produkts zu erfassen. Dadurch sollen möglichst alle Elemente der Produktarchitektur auf verschiedenen Hierarchieebenen bewertet und verglichen werden können. Dafür umfasst die HDDSM eine Beziehungsbasis zur Erfassung einer Vielzahl standardisierter Arten von Relationen und eine hierarchische Modellierungsmethode zur Unterstützung der modularen, effizienten Modellierung von DSMs mit einem hohem Detaillierungsgrad.

Analog zu TILSTRA ET AL. erfolgt die **Erläuterung der Modellierung der HDDSM** mittels eines abstrakten Systems aus fünf Elementen E_1 bis E_5 und einem Umgebungs- bzw. Umweltelement U , die miteinander interagieren. (s. Bild 30). Dabei wird jede Interaktion mit einer „1“ in der DSM dargestellt, in welcher das Spaltenelement die Quelle der Interaktion ist (vgl. IC-Konvention in Kapitel 3.2.3 und Pfeilrichtung im Systemgraph in Bild 30).

Wie in Bild 30 gezeigt ist, kann das Element 5 des Systemmodells (S) durch Herunterbrechen bzw. Konkretisieren (s. MKM in Kapitel 2.3.2) auf eine niedrigere Hierarchieebene (i. S. v. „Zoom-in“) in die drei Elemente 5a, 5b und 5c des Teilsystemmodells (T) detailliert werden. Dabei können **Relationen zwischen Elementen unterschiedlicher Hierarchieebenen** modelliert werden (z. B. zwischen Element 3 und E_{5b} , s. Bild 30 und detailliert in Bild 31).

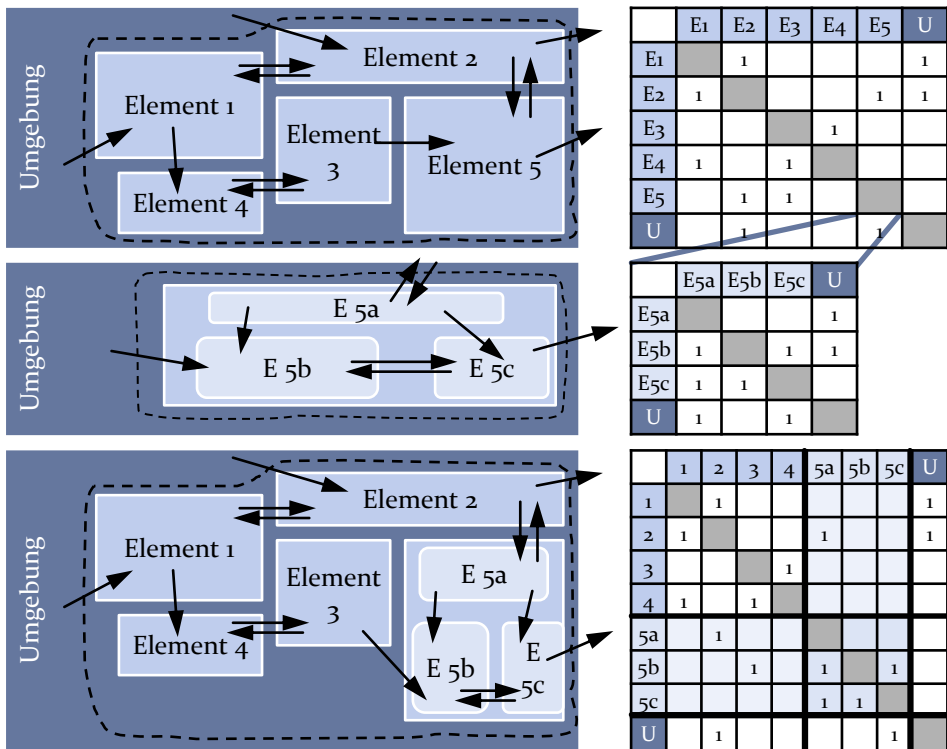


Bild 30: Veranschaulichung der Systemhierarchiemodellierung des HDDSM-Ansatzes (in Anlehnung an [554])

Mögliche **Relationen zwischen Elementen des erweiterten Systemmodells H** lassen sich durch die Kombination von Informationen aus den beiden Systemmodellen S und T ermitteln (s. Bild 31 für detaillierte Vorgehensweise). Für ein besseres Verständnis sind die zwei DSMs S und T in unabhängigen Systemen (i. S. v. Teilmatrizen zerlegt bzw. aufgespalten). Diese Teilmatrizen werden anschließend in das erweiterte Systemmodell H – wie in Bild 31 dargestellt – zusammengesetzt. So umfasst z. B. die Teilmatrix H_{ab} die Relationen zwischen den Elementen 5a, 5b und 5c und den Elementen 1, 2, 3 und 4, wobei die Spaltenelemente 5a, 5b und 5c die Quelle der Relationen sind. Die Teilmatrix H_{ab} wird durch Multiplikation von S_{ab} , welche Relationen zwischen Element 5 und den Elementen 1, 2, 3 und 4 repräsentiert mit T_{ba} , welche Relationen zwischen den Elementen 5a, 5b und 5c und Elementen außerhalb von Element 5 repräsentiert, ermittelt. Das Ergebnis dieser Multiplikation gemäß Gleichung (1) ist in Bild 31 dargestellt. Durch ein „R“ für Review (Überprüfung) wird in den Zellen angezeigt, dass diese Relationen noch manuell zu überprüfen sind. Dabei ist von Vorteil, dass die übrigen Zellen ignoriert werden können, da es dort keine

möglichen Relationen gibt. So ist es bspw. nicht notwendig, das Systemmodell zu überprüfen, ob Element 5b die Quelle einer Relation für Elemente des Systemmodells S ist, da aus dem Teilsystemmodell T bekannt ist, dass Element 5b nicht die Quelle von Relationen außerhalb von T ist.

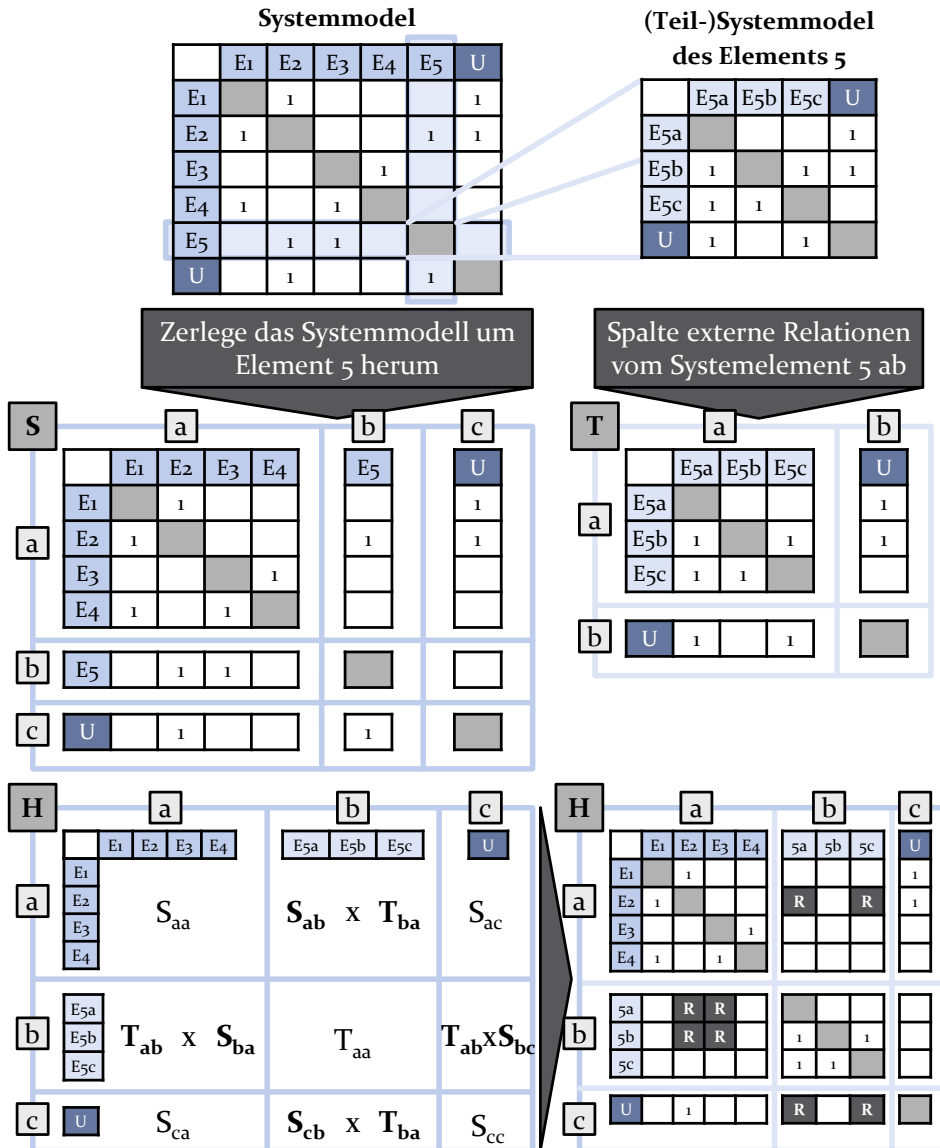


Bild 31: Vorgehensweise zur systematischen Zerlegung von DSMs in Submatrizen und zur Ermittlung übergreifender Relationen (in Anlehnung an [554])

$$H_{ab} = S_{ab} \times T_{ba} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

Die sog. **Interaktionsbasis** für die HDDSM ist in Bild 32 abgebildet und stellt nach TILSTRA ET AL. im Vergleich zu anderen DSM-bezogenen Forschungsarbeiten (u. a. in [433], [260], [254]) eine signifikante Erweiterung der Interaktionen dar [554]. Dabei reichen die drei Hauptumsatzarten (Stoff-, Energie- und Informationsumsatz, s. Tabelle 9 in Anhang A.6) jedoch nicht aus, um alle möglichen Interaktionen zwischen physikalischen Produkten zu modellieren, was v. a. daran liegt, dass Funktionsmodelle bewusst unabhängig von der Produktgestalt erstellt werden (vgl. Bild 119). Daher wurde die Interaktionsbasis von TILSTRA ET AL. um zusätzliche Interaktionen erweitert, wie z. B. um die von PIMMLER & EPPINGER [433] definierten zwei räumlichen Interaktionstypen zur Modellierung von räumlicher Nähe sowie Ausrichtung bzw. Orientierung zwischen zwei Elementen (s. Bild 32).

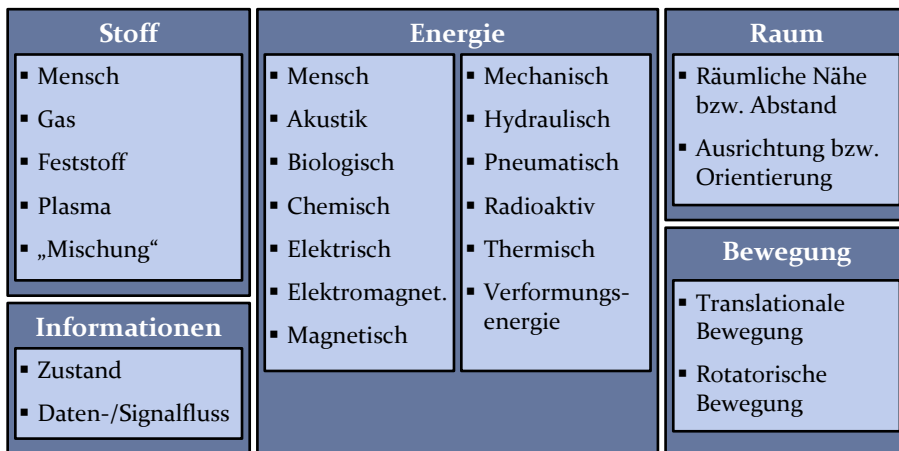


Bild 32: Interaktionsbasis für die HDDSM (in grober Anlehnung an [554])

Ansatz zur Verwendung von DSMs für wissensbasierte Engineering-Softwaresysteme (s. [51], [52], [53], [54]): Im Fokus der Arbeiten von BHASKARA, einem Mitarbeiter des Unternehmens Boeing, steht die Nutzung von in der Lattix Plattform (s. Kapitel 3.3.4) modellierten **DSMs im Rahmen der wissensbasierten Entwicklung** mittels Designsystemen, wobei darunter eine Kombination aus dem technischen Entwurfsprozess und dem Softwareentwicklungsprozess verstanden wird. Dabei verfolgt der Ansatz u. a. das Ziel, bestehende Modelle (v. a. CAD- und Software-Modelle in wissensbasierten Systemen) mittels DSMs besser verstehen, analysieren

und darauf aufbauend umgestalten zu können (vgl. Bild 33). Hierfür ist es hilfreich, dass aufgrund der Relationen zwischen stets zwei Elementen (d. h. 1:1-Relationen) in der DSM-Produkt bzw. der DSM-Software die Auswirkungsanalyse einer Design-Änderung auf Anforderungs- oder Design-parameterebene leicht verfolgt werden kann. Zudem werden mit Blick auf die CAD-Modelle, welche u. a. aufgrund der Repräsentation von Konstruktionsparameter, Konstruktionsregeln, Relationen zwischen Bauteilen und der Modellierungshistorie sehr komplex werden, Methoden zur Analyse, Überprüfung, Komplexitätsreduzierung auf Grundlage der matrixbasierten Produktmodelle vorgestellt.

Für die **Akquisition des informalen und formalen Wissens** im Rahmen der manuellen Modellierung des (wissensbasierten) Produktmodells in Form von DSMs wird als Unterstützung MOKA (s. Anhang A.16) genutzt. Für diese Arbeit ist u. a. interessant, dass drei der vier prinzipiellen Typen bzw. Kategorien von DSMs (s. Kapitel 3.2.3) einem groben „Prozessdatenmodell“ zugeordnet werden (s. Bild 33).

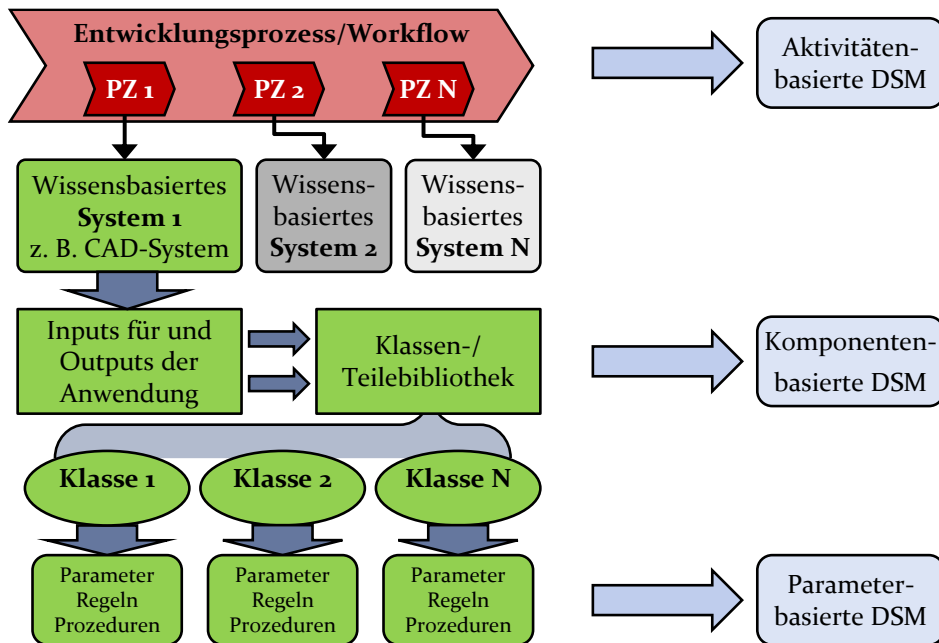


Bild 33: Verorten von DSM-Kategorien in ein „Prozessdatenmodell“ (in Anlehnung an [51])

Diese einzelnen DSMs sind in ein XML-Datenformat (s. [76]) zu überführen und via einer Lattix Data Import (LDI)-Schnittstelle in die Lattix Plattform zu importieren. Dabei weist BHASKARA darauf hin, dass eine **bidirektionale IT-Schnittstelle** zwischen CAD-Modellen und DSM-basierten

Produktmodellen zwar nicht Teil seiner Forschungsarbeiten ist, aber für die Effizienz und Anwendbarkeit v. a. in der industriellen Praxis erforderlich sei. Dies liegt nach seiner Auffassung insb. daran, dass dadurch einerseits die einzelnen Elemente und deren Relationen aus dem Strukturbaum des CAD-Modells ausgelesen und in eine Matrix überführt werden könnten und andererseits, dass CAD-Modelle auf Grundlage bereits bestehender, matrixbasierter Produktmodelle erstellt werden könnten.

Zudem sei auf das US-Patent von BHASKARA mit dem Titel „Modernization of legacy software systems based on modeled dependencies“ verwiesen [54], in dem eine **Methode zur Migration einer nicht objektorientierten Anwendung auf eine objektorientierte Plattform** mit 15 konkreten Ansprüchen beschrieben wird. Dieses Patent liefert insofern wichtige Impulse für diese Arbeit als die darin beschriebene Methode v. a. auf einigen Konzepten zur Generierung und Analyse von diversen DSMs beruht.

Weitere matrixbasierte Modellierungsansätze sind zusammengefasst in Anhang A.18 mit entsprechenden Literaturquellen zu finden. Weitere, auch für diese Arbeit relevante Ansätze sind auch in den vom Autor dieser Arbeit betreuten studentischen Arbeiten sowie in [378] zu finden.

3.3.2 Graphbasierte Visualisierungsansätze

Die Erstellung von matrixbasierten Modellen (s. Kapitel 3.2.3) mittels der dafür erforderlichen matrixbasierten Modellierungsansätze (s. Kapitel 3.3.1) können sehr gut in Form von Listen und Matrizen computerunterstützt modelliert und verarbeitet werden. Diese dienen als Datenbasis für deren visuelle Darstellung in Form von Graphen und Netzwerken (vgl. [555]). Aufgrund ihrer Eignung zur **übersichtlichen Darstellung komplexer Systeme sind graphbasierte Visualisierungsansätze** in der Komplexitäts- und Netzwerktheorie stark verbreitet und erleichtern die Lösungsfindung. Daher werden in diesem Kapitel zunächst die für ein erforderliches Grundverständnis relevanten Grundlagen der Graphentheorie erläutert, um darauf aufbauend prinzipielle Arten von Graphen sowie verschiedene Layouts von Graphen überblicksartig vorzustellen.

Grundlagen der Graphentheorie

Die Graphentheorie ist eine Teildisziplin der Mathematik. Dabei besteht ein Graph $G = (E, R)$ aus einer Menge von Knoten (engl. node, d. h. Elementen E) und Kanten (engl. link oder edges, d. h. Relationen R), die die Relation(en) zwischen Knoten bzw. Elementen repräsentieren. Statt der in der Graphentheorie üblichen Verwendung der beiden Begriffe Knoten und

Kanten werden für ein besseres Verständnis in dieser Arbeit die aus der Systemtheorie bekannten Synonyme Element und Relation verwendet. Zur **visuellen Veranschaulichung von Graphen** werden Elemente oft mit einem Punkt oder einem Symbol und Relationen zwischen Elementen mit einer Linie dargestellt [100]. Bild 34 zeigt den Prozess des Anwendungsbeispiels (vgl. drei Produkt- und vier Prozesselemente in Kapitel 1.5) als DSM und als Graph. Dabei lässt sich der Graph G mit der Elementmenge $E = \{P_1, P_2, P_3, P_4\}$ und der Relationenmenge $R = \{\{P_1, P_3\}, \{P_1, P_4\}, \{P_2, P_3\}, \{P_2, P_4\}\}$ mathematisch ausdrücken. Ist wie in diesem Beispiel $r = \{P_1, P_3\}$ eine Relation von G , dann sind die Elemente P_1 und P_3 zueinander adjazent oder benachbart und inzident zur Relation r . Somit lässt sich der Graph auch als sog. Adjazenz- bzw. Inzidenzmatrix darstellen [562].

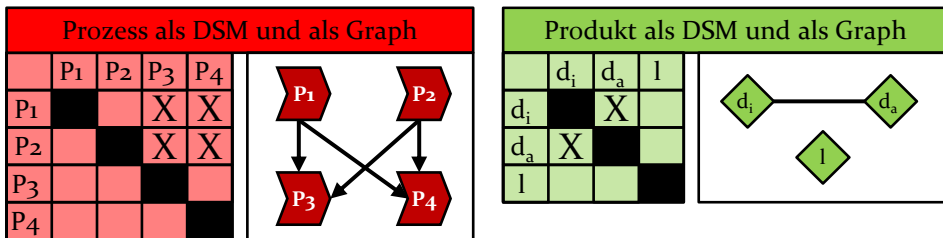


Bild 34: Vereinfachte Beispiele für matrix- und graphbasierte Repräsentationsformen

Sowohl in der Graphentheorie als auch für diese Arbeit ist die **Definition des Weges** grundlegend. Sofern ein Graph nur aus zusammenhängenden Relationen besteht und alle Relationen so durchlaufen werden können, dass keine Relationen mehrfach verwendet wird, wobei Elemente beliebig oft durchlaufen werden, ist der Weg gemäß Gleichung (2) definiert. Dabei treten Wege oft als Teilgraphen (vgl. Teilsystem) eines Graphens auf. Ein häufiger Anwendungsfall ist die Bestimmung der bestehenden Wege zwischen zwei Elementen (z. B. zwischen einem Produkt- und einem Spezifikationselement). Zur Vertiefung und Ergänzung der Grundlagen wird auf die v. a. auf die einschlägige Literatur verwiesen (u. a. [298], [100], [329], [562], [126], [555]).

$$R(P_n) = \{\{e_0, e_1\}, \{e_1, e_2\}, \dots, \{e_{n-1}, e_n\}\} \quad (2)$$

Grundlegende Arten von Graphen

Neben diesen Grundlagen von Graphen umfasst die Graphentheorie auch grundlegende Arten bzw. Typen von gewissen Graphen, die zur besseren **Beschreibung von Graphen** dienen. Die für die Arbeit relevanten Aspekte und Arten werden nachfolgend zusammengefasst.

Die drei **grundlegenden Arten von Graphen** sind gemäß BÜSING [100] und TITTMANN [555] der vollständige Graph, der Baum und der bipartiter Graph (s. Bild 35). Ein vollständiger Graph liegt vor, wenn jedes Element zu jedem anderen Element benachbart ist. Als Baum wird ein Graph benannt, wenn ein zusammenhängender, kreisfreier Graph vorliegt (d. h. keine Relation zu einem vorherigen Element). Ein bipartiter Graph ist dadurch charakterisiert, dass sich die Elemente in zwei Teilmengen X und Y aufteilen lassen und nur Relationen zwischen diesen zwei Teilmengen und nicht innerhalb einer Teilmenge bestehen. Dazu besitzt ein schlichter Graph weder Schlingen noch parallele Relationen [195]. Dabei sind Schlingen Relationen, die ein Element verlassen und in diesem wieder enden (d. h. Ausgangs- und Endelement sind identisch, s. Bild 35 Multigraph). Mit parallelen Relationen ist gemeint, dass mehrere Relationen zwischen den gleichen zwei Elementen existieren. Als Multigraphen werden Graphen bezeichnet, welche sowohl Relationen, die ein Element mit sich selbst verbinden, als auch mehrfache Relationen zwischen zwei identischen Elementen erlauben [329] (s. Anhang A.12). Im Multigraph in Bild 35 existieren zwischen den Elementen 3 und 4 zwei Relationen und Element 7 besitzt eine Schleife.

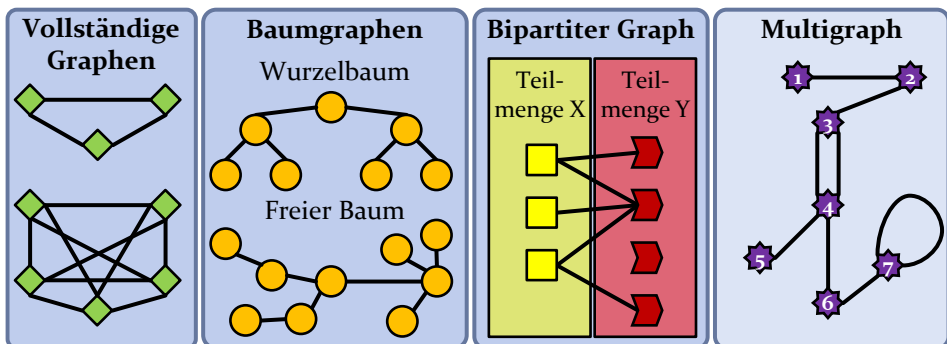


Bild 35: Grundlegende Arten von Graphen (vgl. u. a. [100], [555])

Grundsätzlich wird zwischen **ungerichteten und gerichteten Graphen** differenziert. Während ein ungerichteter Graph aus nicht gerichteten Relationen besteht (s. Bild 34 rechts der Produktgraph ohne Pfeile), sind in einem gerichteten Graph (auch als Digraph bezeichnet) die Richtungen der Relationen zwischen den Elementen zu finden (s. Bild 34 links der Prozessgraph mit Pfeilen). Daher ist im Gegensatz zum ungerichteten Graphen die Reihenfolge der Elementpaare beim gerichteten Graphen von Bedeutung. Dabei stellt das erstgenannte Element das Ausgangselement und das zweitgenannte das Endelement der Pfeilrichtung dar (s. die zwei Leserichtungen von Matrizen in Kapitel 3.2.3). Dagegen können bei ungerichteten Graphen die Einträge in den Matrixzellen an der Diagonalen gespiegelt werden.

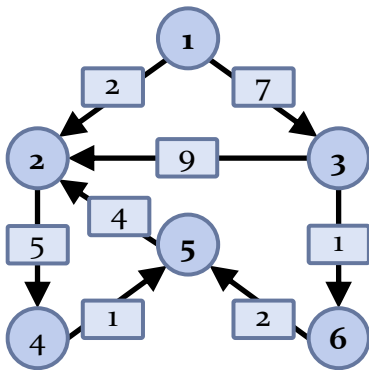
Die **Einträge zur Charakterisierung der Relationen in den Zellen der Matrix** können neben einer (noch) unbekannten Wertigkeit (oft mit einem „X“ gekennzeichnet, vgl. Bild 34) sowie einer möglichen Richtung auch die Stärke bzw. Gewichtung der Relationen zwischen den Elementen der Zeile und Spalte mittels einer reellen Zahl repräsentieren (d. h. gewichteter oder bewerteter Graph, s. [445], [562] für Details). Dabei beschreibt die numerische Wertigkeit „1“ eine Relation bereits genauer als mit „X“. Dabei gilt, dass ein Element durch ein anderes Element umso stärker beeinflusst wird, je höher die Wertigkeit auf einer definierten Skala ist. Die Einträge in den Diagonalen von Matrizen repräsentieren das Vorhandensein von Schlingen bzw. Selbstreferenzen (z. B. bedeutet kein Eintrag oder eine „0“ keine Schlinge). Für noch aussagekräftigere Visualisierungen werden Graphen oft um weitere Eigenschaften und Informationen ergänzt, wie z. B. farbige Elemente und Relationen.

Da in einem bewerteten bzw. gewichteten Graph **Relationen zwischen den Elementen Entfernungen** repräsentieren können, kann z. B. auch die kürzeste Verbindung zweier Knoten berechnet werden. Dazu werden alle Gewichte von Relationen auf dem (kürzesten) Pfad zwischen den zwei Elementen addiert. So wird in dem beispielhaften, bewerteten Graphen in Bild 36 die Entfernung d (engl. distance) von Element 1 zu Element 5 über die zwei Elemente 2 und 4 gemäß Gleichung (3) berechnet [329].

$$d(1,5) = 2 + 5 + 1 = 8 \quad (3)$$

Diese Gewichtungen können auch durch die entsprechende Adjazenzmatrix repräsentiert werden. Zudem kann eine **Entfernungsmatrix $D(G)$** nach Gleichung (4) definiert werden. Somit enthält eine Entfernungsmatrix $D(G)$ alle Entfernungen für alle Kombinationen der Elemente. Dabei ist $D(G)$ bei ungerichteten Graphen stets symmetrisch, bei gerichteten Graphen i. d. R. unsymmetrisch. So ist für den beispielhaften, bewerteten Graphen die entsprechende Entfernungsmatrix in Bild 36 rechts dargestellt. Darin sind auf der Diagonalen die Entfernungswerte „0“, da bei Graphen ohne Schleifen die Entfernung eines Knotens zu sich selbst als „0“ definiert ist. Existiert zwischen zwei Elementen kein Pfad, wie z. B. von allen Elementen zu Element 1 (dieser besitzt nur ausgehende Relationen), dann wird die Entfernung mit „ ∞ “ (d. h. unendlich) angegeben. Die Entfernung von Element 3 zu Element 4 wird in der Matrix mit 12 angegeben. Diese errechnet sich durch die Verbindung über die Elemente 6, 5 und 2.

$$D(G) = (d_{ij}) \text{ mit } d_{ij} = d(e_i, e_j) \quad (4)$$



Entfernungsmatrix						
$D(G) =$	0	2	7	7	8	8
	∞	0	9	5	6	10
	∞	7	0	12	3	1
	∞	5	14	0	1	15
	∞	4	13	9	0	14
	∞	6	15	11	2	0

Bild 36: Beispiel für einen gerichteten und gewichteten Graphen (in Anlehnung an [329])

Grundlagen zu verschiedenen Layouts von Graphen

Die Struktur einer DSM und einer Adjazenzmatrix des entsprechenden Graphen ist identisch. Daher werden auch Graphen mit steigender Vielzahl und Vielfalt an Elementen und Relationen zunehmend unübersichtlich. Daher ist eine zentrale Herausforderung bei graphischen Visualisierungen von Modellen komplexer Systeme die benutzerfreundliche Darstellung (vgl. [27]). Ein Graph ist zwar eindeutig durch seine Menge an Elementen und Relationen definiert, aber bzgl. der Gestaltung des Graphenlayouts gibt es in Abhängigkeit von der Graphenart viele verschiedene Optionen [561]. Aus diesem Grund ist eine für die jeweilige Zielsetzung geeignete Visualisierung komplexer Systeme mittels Matrizen und Graphen sehr bedeutend. Während bzgl. der Visualisierung die Möglichkeiten für Matrizen begrenzt sind (s. Kapitel 3.2.3 und 3.3.3), stehen bei graphbasierten Visualisierungsansätzen einige Layouts zur Verfügung, die Elemente und Relationen so anordnen, dass Zusammenhänge in komplexen Systemen leicht(er) erfasst werden können. Daher werden in diesem Abschnitt **Grundlagen und verschiedene Layouts von Graphen** vorgestellt (s. Bild 37). Mit der zweckmäßigen Repräsentation von Graphen beschäftigt sich das sog. Graphzeichnen (engl. Graph Drawing) als Teildisziplin der Informatik, die verschiedene Algorithmen zur Bestimmung einer (optimalen) Anordnung von Elementen entwickeln (s. für Details [125], [407], [555]).

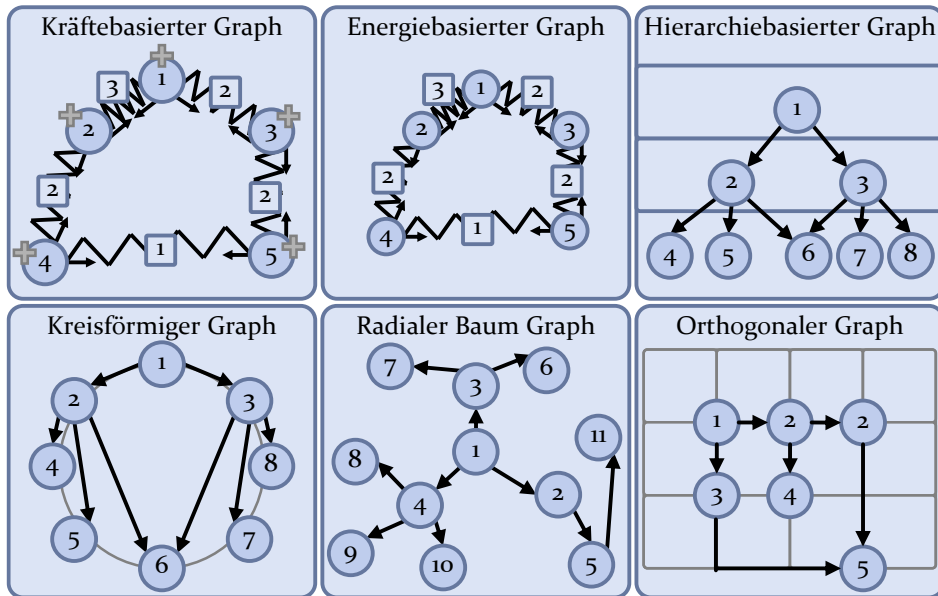


Bild 37: Beispiele für verschiedene Layouts von Graphen

Im Vergleich zur Visualisierung von ungerichteten Graphen sind bei gerichteten Graphen auch noch deren Richtung (Fluss) bei deren Darstellung zu beachten. Daher sind für eine möglichst gute Erfass-, Lesbar- und Interpretierbarkeit der Zusammenhänge in komplexen Systemen die folgenden **Optimierungsansätze für ein übersichtliches Layout** u. a. nach [546], [298], [446] zu befolgen, wobei dies i. d. R. bei komplexen (un)gerichteten Graphen nur bedingt möglich ist (s. für Gründe [18]):

- Optimierungsansätze für Elemente: Gleichmäßiges Verteilen von Elementen; Anordnung von (nicht) adjazenten Elementen nah (entfernt) zueinander;
- Optimierungsansätze für Relationen: Vermeiden von nach oben zeigenden Relationen; Minimieren von Kreuzungen von Relationen; Ausrichten von Relationen weitmöglichst in vertikaler Richtung; Darstellen von Relationen möglichst ohne Knicke, geradlinig, kurz, symmetrisch sowie gleichmäßig (sofern diese die Bewertung der Relationen zulässt).

Bei **kräftebasierten Ansätzen** wird der Graph mittels eines Kräftemodells abgebildet, d. h. einerseits stoßen sich die Elemente konstant gegenseitig ab (i. S. v. elektrisch geladenen Partikeln), werden aber andererseits durch die Relationen in Form von Federn zusammengezogen (vgl. Hookesches Gesetz) [284] (s. Bild 37). Mit diesem Kräftemodell erfolgt die Berechnung

eines Systems mit dem Optimierungsziel eines energieminimalen Zustands, also ein energetisches Gleichgewicht zwischen den Elementen [298]. Dabei werden die verschiedenen Wertigkeiten der Relationen (vgl. Distanzen zwischen Elementen in Bild 36) durch die Anziehungskraft der Federn ausgedrückt. Somit führt eine hohe Wertigkeit (bzw. Anziehungskraft) zwischen zwei Knoten zu einer kürzeren Distanz und folglich zu einer nahen Anordnung (s. [355] für Beispiele). Dementsprechend liegen auch adjazente Knoten nah beieinander und inzidente Knoten weit auseinander. Einen der ersten kräftebasierten Algorithmen stellte TUTTE bereits 1963 in [564] vor. Aufbauend u. a. auf dieser Arbeit zählen zu den am meisten verbreiteten Vertretern von kräftebasierten Ansätzen, die i. d. R. nur geringe Unterschiede aufweisen, nach KAUFMANN [298] der spring-embedder-Algorithmus [139] (s. auch [140]) und der Fruchterman-Reingold-Algorithmus [196] (s. [263]).

Den **energiebasierten Ansätzen**, die den kräftebasierten Ansätzen sehr ähnlich sind, liegen Energiemodelle zur Erstellung von Layouts zugrunde (s. Bild 37). Ein verbreiteter Vertreter ist der Algorithmus von KAMADA & KAWAI [288] nach dem das Optimierungsziel die Minimierung der Gesamtenergie des Systems ist. Im Gegensatz zu den kräftebasierten Ansätzen herrschen zwischen den Elementen keine abstoßenden Kräfte. Daher wird die Anordnung der Elemente nur von den Relationen in Form von Federn, welche eine potenzielle Energie besitzen, bestimmt. Im Vergleich zu den kräftebasierten Ansätzen ergeben sich somit kompaktere Graphenlayouts, da aufgrund der fehlenden Abstoßung der Elemente diese durch die Federkräfte stärker zusammengezogen werden. Dabei ist die Länge (i. S. v. Distanz) der Federn proportional zur minimalen Anzahl der Relationen zwischen zwei Elementen. Die Berechnung der Längen erfolgt mit Hilfe von Gleichungen, welche die potenziellen Energien der einzelnen Federn mit deren Abstand zueinander in Verbindung bringt. Somit liegen adjazente Elemente, die nur durch eine Relation verbunden sind, am nächsten beieinander. Daraus ergibt sich der Vorteil, dass Elemente mit vielen Relationen enger zusammenliegen und somit schneller identifiziert werden können (sofern die Lesbarkeit bei zu naheliegenden Elementen nicht eingeschränkt wird).

Eine weitere Möglichkeit sind **hierarchie- bzw. ebenenbasierte Ansätze**, die mit Hilfe von Schichtmodellen Graphen visualisieren [546] (s. Bild 37 und auch [545], [518]). Das Optimierungsziel ist, dass möglichst alle Relationen (möglichst) senkrecht verlaufen, wobei bei gerichteten Graphen je nach Richtung zwischen „nach oben“ (engl. upward) und „nach unten“ (engl. downward) zu differenzieren ist. Dazu wird zuerst jedes Element in

eine Schicht bzw. Ebene durch die Bildung von Mengen und Äquivalenzklassen verortet (z. B. Verortung der Elemente mit der geringsten Menge an eingehenden Relationen in die oberste Ebene). Danach werden die Elemente so angeordnet, dass die Relationen möglichst überschneidungsfrei sind (z. B. durch Einfügen „künstlicher“ Elemente) und Zickzacklinien durch genaue Koordinaten von Elementen vermieden werden. Die hierarchiebasierten Ansätze sind v. a. bei Graphen mit Baumstruktur verbreitet, wie z. B. für Organigramme, Datenstrukturen, Klassen-, Datenfluss- und Workflowdiagramme. Ein grundlegender und bekannter Vertreter hierarchiebasierter Ansätze ist der Algorithmus von SUGIYAMA [538] (s. [537]).

Kreisbasierte Ansätze (engl. Circularlayout) veranschaulichen Systeme dadurch, dass diese Elemente auf einem Kreisradius mit gleichem Abstand zu ihren Nachbarelemente positionieren [284] (s. Bild 37 und auch [518]). Dabei steigt der Radius mit der zunehmenden Anzahl an Elementen. Auch wenn diese Darstellung zu vielen Überschneidungen von Relationen führt, können mittels des kreisförmigen Layouts Relationen gut nachvollzogen werden.

Im Gegensatz zum Kreislayout wird beim **Radialen Baum-Layout Ansatz** (engl. Radial Tree Layout) ein Element im Kreismittelpunkt als Zentrum definiert [284] (s. Bild 37 und auch [518]). Dabei liegen die adjazenten Elemente auf dem konzentrischen Kreis, der die geringste Entfernung vom Zentrum hat. Je größer der Radius ist, auf dem ein Element konzentrisch angeordnet ist, und folglich je mehr Elemente zwischen dem Zentrum und diesem Element liegen, desto größer ist der Abstand und damit die Distanz des Elements zum Element im Zentrum. Der Abstand zwischen den einzelnen Elementen stellt die jeweilige Distanz dar. Sofern ein Graph als Baumstruktur und damit ohne rücklaufende Relationen vorliegt, kann dieser ohne Überschneidung visualisiert werden.

Bei **orthogonalen Ansätzen** werden die einzelnen Elemente auf planaren, orthogonalen Gittern positioniert, wobei die Relationen nur auf diesen vertikalen und horizontalen Gitternetzlinien verlaufen [537] (s. Bild 37). Dazu werden die Elemente mit den Zielen der Minimierung von Knicken, der Fläche und der Gesamtlänge der Relationen möglichst optimal auf einem orthogonalen Gitter positioniert.

Die **spektralen Ansätze** verwenden die Eigenvektoren der Elemente zur Positionsbestimmung sowie Vektoren und Matrizen zur Raumaufteilung [318]. Zur Berechnung der hierfür erforderlichen Laplacematrix wird die Differenz zwischen einer Gradmatrix und einer gewichteten Adjanzenzmatrix gebildet (vgl. [298], [445]). Dabei stellt die Gradmatrix eine

beschriftete Diagonale einer $n \times n$ -Matrix dar, welche die Elementgrade aller Elemente aufweist. Der Elementgrad ist das Ergebnis der Addition aller Gewichte der adjazenten Relationen, welche eine Relation zu diesem Element haben [445]. Dabei bilden Elemente mit einer engeren Beziehung Cluster, wobei in einem Cluster Elemente auch auf derselben Position liegen können, da sich diese nicht gegenseitig abstoßen. Mit speziellen Laplacematrizen können unter gewissen Voraussetzungen Layouts verbessert werden, wie z. B. lose Elemente nach innen und Elemente höheren Grades nach außen gezogen werden (vgl. [445]).

Zur Gestaltung des Layouts von Graphen kann neben der Anordnung von Elementen und Relationen bspw. auch auf unterschiedliche Symbole und Linienarten zurückgegriffen werden (s. auch [537]). Die **Wahl des Layouts von Graphen zur Visualisierung komplexer Systeme** ist daher immer abhängig von der Art des Graphens und dem Zweck der Visualisierung mit dem Ziel für den Nutzer alle kontextbezogenen, erforderlichen Informationen möglichst transparent, eindeutig und verständlich zu präsentieren (s. auch [196]). Zur computerunterstützten Anwendung dieser Algorithmen stehen geeignete Softwareprogramme zur Verfügung (s. Kapitel 3.3.4).

3.3.3 Mathematische Analyseansätze

Nachfolgend werden ausgewählte **Ansätze zur mathematischen Analyse** von matrix- und graphbasierten Modellen überblicksartig beschrieben. Diese dienen v. a. zur Analyse komplexer (Teil-)Systeme und sind daher auch im Assistenzsystem zum Komplexitätsmanagement in der PE prototypisch implementiert. In einigen von den vom Autor dieser Arbeit betreuten studentischen Arbeiten sind tiefergehende Erläuterungen zu finden.

Grundsätzlich sind nicht alle Analysemethoden für jedes System geeignet, so dass eine **Kategorisierung von Analysemethoden** hilfreich ist. So wird in Bild 38 zwischen Analysemethoden für Elemente und Relationen, Analysemethoden für individuelle bzw. domänenspezifische DSMs sowie Analysemethoden für holistische bzw. domänenübergreifende Systeme differenziert. Dabei liegen den Analysemethoden i. d. R. die mathematischen Grundlagen aus der Netzwerk- und Graphentheorie zugrunde.

Analysemethoden für Elemente und Relationen			
▪ Aktivsumme	▪ Aktivität	▪ Startelement	▪ Transitelement
▪ Passivsumme	▪ Kritikalität	▪ Endelement	▪ Bus
Individuelle, domänenspezifische Analysemethoden			
Methoden zur Analyse komponenten- und organisationsbasierter DSMs (z. B. Produktanalyse)		Methoden zur Analyse aktivitäten- und parameterbasierter DSMs (z. B. Prozessanalyse)	
Clustering	Banding – Aktiv- und Passivsumme	Partitioning	Banding – Vorgänger und Nachfolger
		Tearing	
Holistische, domänenübergreifende Analysemethoden			
Methoden zur Analyse von Änderungen und deren Auswirkungen		Methoden zur Analyse von weiteren Zielsetzungen	
Vorwärtsgerichtete Analyse		Indirekte Relationen	
Rückwärtsgerichtete Analyse		Maß an Verbundenheit	
Wirkungsnetze/Beeinflussungsmatrix		Geschlossene Kreise	
Beeinflussungspfadsuche		Zyklus-Analysen	
		Partitioning mit Änderungsanfälligkeit	
		Banding einer MDM	

Bild 38: Übersicht und Kategorisierung ausgewählter Analysemethoden

Im Rahmen der Analyse von Systemen werden in einem ersten Schritt oft zuerst die **Analysemethoden für Elemente und Relationen** eingesetzt, da deren Ergebnisse als Grundlage für die folgenden Analysen der Systeme dienen. Eine Übersicht über relevante Methoden zur Analyse einzelner Elemente und Relationen ist in Bild 39 mit Literaturverweisen zu finden.

Individuelle, domänenspezifische Analysemethoden

In diesem Abschnitt werden **Analysemethoden von statischen als auch zeitbasierten DSMs** (vgl. Kapitel 3.2.3) erläutert. Dabei werden die in der DSM-Forschung verbreiteten und für diese Arbeiten relevanten Analysemethoden Partitioning, Tearing, Banding und Clustering kurz beleuchtet.

Analyse- methode	Formel oder Definition	Erläuterung mit kurzer Interpretation	Ver- weis
Aktiv- summe	Summe der ausgehenden Relationen eines Elements	Die Aktivsumme eines Elements beschreibt den Einfluss eines Elements bei dessen Änderung auf andere Elemente. Je größer die Aktivsumme ist, desto größer ist der Einfluss und/oder desto mehr weitere Elemente werden beeinflusst.	[117]
Passiv- summe	Summe der eingehenden Relationen eines Elements	Die Passivsumme eines Elements beschreibt die Beeinflussung eines Elements durch Änderungen anderer Elemente. Je größer die Passivsumme ist, desto größer ist der Einfluss und/oder desto mehr andere Elemente beeinflussen das Element.	[117]
Aktivität	Aktivität ist der Quotient aus Aktiv- und Passivsumme	Die Aktivität veranschaulicht die Involviertheit des Elements an Änderungen und beschreibt daher die Beeinflussung des Elements auf andere.	[117]
Kritika- lität	Kritikalität ist das Produkt aus Aktiv- und Passivsumme	Die Kritikalität ist ein Art „Integrationsmaß“ von Elementen bei Änderungen und beschreibt daher die Vernetzungsintensität und damit die „Anfälligkeit“ für bzw. auf Änderungen.	[117]
Start- element	Element mit nur ausgehenden Relationen	Ein Startelement kann nur Änderungen / Input in das System einspeisen, aber selbst nie von Änderungen betroffen sein kann.	[355]
End- element	Element mit nur eingehenden Relationen	Ein Endelement kann nur durch Änderungen von anderen Elementen und sich selbst betroffen sein, aber selbst nie Änderungen verursachen.	[355]
Transit- element	Element mit einer ein- und einer ausgehenden Relation	Ein Transitelement leitet Änderungen nur weiter und zwar i. d. R. ohne diese Änderungen zu verstärken oder abzuschwächen.	[355]
Bus- element	Element mit vielen ein- und/oder ausgehenden Relationen	Ein Bus ist ein Verbindungselement. Je mehr ausgehende Relationen ein Bus hat, desto mehr Elemente sind bei Änderungen betroffen. Je mehr eingehende Relationen ein Bus hat, desto wahrscheinlicher ist dieser von Änderungen betroffen.	[355]

Bild 39: Übersicht über Analysemethoden für Elemente und Relationen (vgl. [117], [355])

Partitioning (s. [531], s. auch [507]): Partitioning, auch als Sequencing bezeichnet, hat das Ziel einer optimalen Prozessabfolge i. S. einer Minimierung von Feedback-Schleifen durch eine entsprechende Sortierung von Zeilen und Spalten zeitbasierter DSMs zu realisieren. Daher ist das übergeordnete Ziel eine Dreiecksmatrix rechts der Diagonalen zu verwirklichen, um Feedback-Schleifen zu eliminieren. Sofern dies aufgrund bestimmter Relationen nicht möglich ist, wird das Ziel verfolgt, die Relationen möglichst in quadratische Cluster bzw. nah entlang der Diagonalen zu bringen, damit die Feedback-Schleifen möglichst kurzgehalten werden (i. S. v.

kleine Iteration). Beim Partitioning ist darauf zu achten, dass durch das Eliminieren von Feedback-Schleifen keine wichtigen Informationsflüsse und -rückflüsse, wie bspw. Qualitätstestergebnisse und Lessons-Learned, verloren gehen (s. [119]).

Tearing (s. [531], s. auch [33]): Die Ausgangsmatrix für das Tearing ist die Ergebnismatrix des Partitionings. Das Ziel von Tearing ist die noch vorhandenen Feedback-Schleifen zu minimieren, in dem die Einträge, auch als Tears bezeichnet, außerhalb der Dreiecksmatrix gelöscht werden. Dabei ist zwischen zwei Tearing-Verfahren zu differenzieren. Dem ersten Verfahren liegt die Annahme zugrunde, dass viele Iterationen der Kontrolle dienen und damit vermeidbare Iterationen sind (z. B. Überprüfung, ob ein Rohrdurchmesser bestimmten Belastungen standhält). Das zweite Verfahren zielt darauf ab, Iterationen innerhalb einer übergreifenden Iteration zu eliminieren. Diese oft unnötigen, zeit- und kostenaufwendigen Feedback-Schleifen können insb. durch entwicklungsbegleitende Berechnungs- und Simulationsverfahren möglichst früh im PEP minimiert werden. Allerdings ist beim Tearing darauf zu achten, dass kein Informationsverlust entsteht. Daher dürfen Tears nur dann gelöscht werden, wenn Iterationen keinen Nutzen haben und wenn z. B. neue Festigkeitsberechnungen anstatt Materialtests genutzt werden und so Iterationen überflüssig machen (s. [33]).

Banding (s. [507], s. auch [33], [86]): Banding hat das Ziel voneinander unabhängige Elemente in statischen und zeitbasierten DSMs mit Hilfe von Streifen in verschiedenen Farben visuell darzustellen. So werden bspw. in Produktmodellen Elemente identifiziert, die viele/wenige Relationen zu anderen Elementen haben, und in Prozessmodellen Prozesse erkannt, die simultan ablaufen können. Da Banding v. a. zur Visualisierung dient, entsteht erst durch die Interpretation und Ableitung von Maßnahmen ein Mehrwert, wie z. B. Prozessverkürzung durch Simultaneous Engineering und Produktänderungsaufwandsreduktion durch Modularisierung.

Clustering (s. [33], s. auch [161], [355] und für mathematische Grundlagen [166]): Das Ziel von Clustering ist die Bildung von (möglichst) überschneidungsfreien Substrukturen (d. h. Teilsystemen, Blöcken) mit Elementen, die viele Relationen zu Elementen innerhalb und möglichst wenige Relationen zu Elementen außerhalb des Clusters haben. Daher sollen Cluster nur wenige Schnittstellen zu anderen Clustern haben und somit zum Systemverständnis beitragen. So unterstützt das Clustering einer komponentenbasierten DSM bei der Definition von Baugruppen und Modulen sowie das Clustering einer organisationsbasierten DSM bei der Bildung von Teams und Abteilungen.

Für das Clustering gibt es eine **Vielzahl an Clusterverfahren**. Im Rahmen der Erstellung von Cluster werden die Zeilen und Spalten einer Matrix (neu) sortiert, so dass die Relationen der Elemente in Cluster entlang der Diagonalen gruppiert werden. Die Gruppierung erfolgt anhand zuvor definierter Kriterien, wie bspw. der Anzahl und Stärke von Relationen, einer bestimmten Anzahl von Clustern oder eines gewissen Abstands. Im mathematischen Sinne ist in einem Cluster der Abstand der einzelnen Elemente bzw. Datenpunkte zueinander i. d. R. kleiner als zwischen Elementen zu anderen Clustern. Daher erfolgt die Erstellung von Clustern auf Basis der durch Distanzfunktionen ermittelten Abstandsmaße zwischen den Elementen. Das üblichste Abstandsmaß ist der Euklidische Abstand, wobei je nach Zielsetzung andere Abstandsmaße zweckdienlicher sind, z. B. die Summe der Abstandsquadrate, der Abstand des max. Elements und der Manhattan-Abstand (s. [166] zur Vertiefung). Im Rahmen des Komplexitätsmanagements auf Grundlage der Analyse von Matrizen werden verschiedene Verfahren eingesetzt. Daher ist nach einer groben Kategorisierung von Clusterverfahren ein Überblick über zehn ausgewählte Clusterverfahren (z. B. k-Means-Algorithmus) mit deren jeweiligen Funktionsweisen, Besonderheiten und Anwendungsbereichen sowie mit Literaturhinweisen zur Vertiefung im Anhang A.20 zu finden.

Methoden zur Analyse von Änderungen und deren Auswirkungen

Clusterverfahren haben durch das Clustern von Elementen die Zielsetzung, die Auswirkungen von Änderungen auf Systemen i. S. der Strategien des Komplexitätsmanagements (s. Kapitel 2.2.3 und Anhang A.10) möglichst zu reduzieren, zu beherrschen und zu vermeiden. Allerdings ist dies aufgrund zunehmend komplexer Systeme (s. Kapitel 2.1) oft nur bedingt möglich. Daher sind v. a. Methoden zur Analyse von Änderungen und deren Auswirkungen i. S. v. Änderungsweitergabe in Form von Kettenreaktionen bzw. Dominoeffekten (engl. engineering change management und insb. change propagation und change prediction) von Bedeutung. So kann bspw. durch Prognosen über Ausbreitung und Art von Änderungen von Elementen bereits in der Projektangebotsphase die Projektplanung und darauf aufbauende Modularisierungs- und Anpassungskonstruktionsschritte unterstützt werden (vgl. [106], [423]). Eine gute Vorhersage umfasst neben den Gründen für die jeweiligen Änderungen auch die von diesen Änderungen betroffenen Elemente und somit die oft ausgelösten Kettenreaktionen [365].

Im Bereich des (technischen) Änderungsmanagements gibt es eine Vielzahl verschiedener Ansätze, Vorgehensweise und Methoden. So haben z. B. HAMRAZ ET AL. für ihren ganzheitlichen Kategorisierungsrahmen von

Publikationen im Bereich des technischen Änderungsmanagements mit Fokus auf PE neben 43 Büchern bzw. Buchkapiteln weitere 384 relevante Publikationen identifiziert, die im Zeitraum von Januar 1996 bis September 2011 jeweils zu Hälfte in einschlägigen Zeitschriften (z. B. Research in Engineering Design, Journal of Engineering Design, Systems Engineering) und Konferenzen (z. B. ICED, DESIGN, DETC und DSM) publiziert wurden [243]. Auf Grundlage ihrer ganzheitlichen, prozessorientierten Kategorisierung der Literatur schlussfolgern HAMRAZ ET AL., dass viele Forschungsansätze erstens zur Verringerung der produkt- und prozessorientierten Auswirkungen von Änderungen, zweitens zur Erleichterung des Umgangs mit Auswirkungen und drittens zur Erhebung von Quellen und Auswirkungen von Änderungen veröffentlicht wurden. Im Gegensatz dazu sind nach deren Forschungsarbeiten nur wenige Ansätze für organisations- und kostenorientierte Reduktionsmaßnahmen, für die Unterstützung bei der Implementierung von Änderungen sowie für den Umgang der Auswirkungen von Änderungen auf die Qualität und die der PE folgenden Phasen publiziert. Die grundsätzliche Schlussfolgerung von HAMRAZ ET AL. ist, dass das (technische) Änderungsmanagement ein im Vergleich zu anderen Forschungsbereichen junger Bereich mit einem hohen Forschungsbedarf ist.

In Bild 40 wird eine **Übersicht über grundlegende Analysemethoden von Änderungen** und deren Auswirkungen gegeben, die auf Grundlage von DSM, DMM und MDM angewendet werden. Zusätzliche, z. T. darauf aufbauende und für diese Arbeit relevante Ansätze zur Analyse von Änderungen werden im Anhang A.20 überblicksartig vorgestellt.

Methoden zur Analyse von weiteren Zielsetzungen

Nachfolgend werden **weitere Analysemethoden** kurz beschrieben. Diese werden Analyse von matrixbasierten Modellen bzgl. diverser Ziele genutzt.

Konnektivitätsgrad (s. [442], [118], [33]): Der Konnektivitätsgrad (i. S. v. Maß an Verbundenheit) beschreibt das Verhältnis zwischen in einer DSM bestehenden Relationen und der Anzahl möglicher Relationen. Dabei berechnet sich die Anzahl möglicher Relationen (bei einer leeren Diagonalen) nach Gleichung (5) und der Konnektivitätsgrad nach Gleichung (6). Der Konnektivitätsgrad dient zur Abschätzung der Systemkomplexität. Mit steigendem Konnektivitätsgrad wird das System i. d. R. komplexer (z. B. aufgrund zunehmender Feedback-Schleifen, s. Tearing) und Analysen somit zeit- und kostenaufwendiger. Zudem kann mittels des Konnektivitätsgrads ermittelt werden, ob zuerst eine Reduzierung der Systemkomplexität erreicht werden muss bevor sog. Realoptionen bzw. flexible Elemente (vgl. Change Prediction Index in Anhang A.20) entwickelt werden können.

Flexible Elemente sind Elemente, die bei einer Änderung keine oder nur kleine Änderungsketten verursachen. Mittels der Identifikation oder gezielten Nutzung dieser flexiblen Elemente wissen Entwickler, welche Elemente ohne (weitere) Auswirkungen geändert werden können oder welche Elemente die Änderungskette beenden können.

$$Anzahl_{Mögl.Relationen} = (Anzahl_{Elemente})^2 - Anzahl_{Elemente} \quad (5)$$

$$Konnektivitätsgrad = \frac{Anzahl_{Bestehender\ Relationen}}{Anzahl_{Mögl.Relationen}} \quad (6)$$

Analyse- methode	Erläuterung mit z. T. kurzer Interpretation, Bewertung	Ver- weis
Beeinflussungsmatrix und Wirkungs- netzwerke	▪ Darstellung von Wirkungen einzelner Elemente auf andere mittels einer Skala in einer Beeinflussungsmatrix, in der die Beeinflussung eines jeden Elements angegeben wird, auch wenn diese nur über indirekte Relationen besteht ▪ Addition der Werte in den Zeilen ergibt die Aktivsumme und Addition der Werte in den Spalten ergibt die Passivsumme ▪ Berechnung der Aktivität und Kritikalität eines jeden Elements ▪ Abschätzung der Stärke der Auswirkungen von Änderungen auf die einzelnen Elemente auf Basis der Beeinflussungsmatrix ▪ Transformation der Beeinflussungsmatrix in Wirkungsnetze bzw. Graphen, die visualisieren welche Elemente andere Elemente beeinflussen und welche von welchem Element beeinflusst werden	[117]
Vorwärts- gerichtete- Analyse	▪ Ausgangsmatrix ist die Beeinflussungsmatrix ▪ Identifizierung und Abbildung der Änderungskette aller von einem sog. Startelement beeinflussten Elemente ▪ Hinzunahme von weiteren Beeinflussungsgraden bzw. -ebenen erfolgt solange, bis alle beeinflussten Elemente identifiziert sind ▪ Abschätzung der Auswirkungen einer Änderung und insb. deren Kettenreaktion gut möglich ▪ Darstellung erlaubt gutes Nachvollziehen des Änderungsverlaufs	[266], [376]
Beeinflussungspfad- suche	▪ Spezielle Anwendungsart der vorwärtsgerichteten Analyse ▪ Ermittlung des Beeinflussungspfads zwischen einem Startelement und einem Element von besonderem Interesse, z. B. mit hoher Beeinflussungskraft und/oder hohem Stellenwert ▪ Abschätzung von Auswirkungen infolge Änderungen auf Elemente, die z. B. große Änderungsketten oder hohe Änderungskosten haben, zur Beurteilung des Änderungsaufwands	[266], [376]
Rückwärts- gerichtete- Analyse	▪ Gegenstück zur vorwärtsgerichteten Analyse ▪ Identifizierung und Abbildung aller möglichen Änderungsketten ausgehend von einem beeinflussten Element zur Ermittlung aller möglichen Ursachen	[266], [376]

Bild 40: Übersicht über grundlegende Analysemethoden von Änderungen (vgl. [177], [266], [376])

Geschlossene Schleifen (s. [153]): Dieser Ansatz dient der Identifikation domänenübergreifender Schleifen (i. S. v. geschlossene Kreise) in MDMs. Die Identifikation geschlossener Schleifen erfolgt schrittweise in einem iterativen Prozess. Ist eine geschlossene Schleife bspw. zwischen zwei Produkt- und zwei Prozesselementen identifiziert, müssen Entwickler beachten, dass eine Änderung an einem dieser vier Elemente alle anderen Elemente und deren Domänen betrifft sowie sich v. a. auch zurück auf das Ausgangselement auswirkt (vgl. sog. Endlosschleife). Daher sind geschlossene Schleifen aufzulösen.

Zyklusanalyse (s. [56]): Da eine MDM eine abstrakte Form einer DSM darstellt, übertragen BIEDERMANN & LINDEMANN die für die DSM verbreitete Zyklusanalyse auf die MDM. So lassen sich z. B. mittels der Zyklusanalyse DSMs aus bestehenden DMMs auf verschiedene Arten ableiten. Zur Auswahl der richtigen bzw. geeigneter Berechnungsverfahren von DSMs und deren anschließender Integration in die MDM zur Systemanalyse ist insb. der Zweck der jeweiligen matrixbasierten Modellierung als auch Domänenwissen erforderlich.

3.3.4 Ansätze zur Softwareunterstützung

Die skizzierten matrixbasierten Modellierungsansätze (Kapitel 3.3.1), graphbasierten Visualisierungsansätze (Kapitel 3.3.3) und mathematischen Analyseansätze (Kapitel 3.3.3) dienen zum Komplexitätsmanagement in der PE. Allerdings ist die Anwendung dieser Ansätze in analoger Art und Weise extrem zeit- und kostenaufwendig, was insb. für umfangreiche, komplexe Systeme zu einem negativen Kosten-Nutzen-Verhältnis führt. Aus diesem Grund ist eine **Vielzahl verschiedener Software-Werkzeuge** entwickelt worden, die bei der Modellierung, Visualisierung und Analyse von komplexen Systemen in der PE unterstützen.

Eine umfassende **Übersicht von IT-Werkzeugen**, die für die matrixbasierte Modellierung, graphbasierte Visualisierung und mathematische Analyse insb. auf Basis von DSMs eingesetzt werden, bietet die Webseite dsmweb.org [560], welche vom Lehrstuhl für Produktentwicklung und Leichtbau der TUM gepflegt wird. Diese werden in die zwei Kategorien der akademischen, frei verfügbaren und kommerziellen Softwarewerkzeuge gegliedert. Für beide Kategorien werden nachfolgend die für die Arbeit relevanten IT-Werkzeuge in alphabetischer Abfolge skizziert.

Akademische Softwarewerkzeuge

Antares DSM (s. [402], [403]): Antares DSM unterstützt bei der Analyse von DSMs mittels einer sog. Quick Triangular Matrix, einem Partitionierungsalgorithmus zur Identifizierung unerwünschter Zyklen. Somit dient Antares DSM zur Optimierung von Systemabhängigkeiten. Weitere Analyse- sowie generelle Visualisierungsmöglichkeiten und eine hilfreiche Dokumentation fehlen. Antares DSM wird aktuell nicht weiterentwickelt.

Cambridge Advanced Modeller (CAM) (s. [623], [622], [621]): Der CAM (vorher als P3- Signposting bezeichnet), bietet eine Vielzahl verschiedener Modellierungs- (z. B. DSMs und Diagrammen), Analyse- (z. B. Clustering) und Visualisierungsmöglichkeiten (z. B. kräftebasiertes Graph-Layout). Der modulare Aufbau von CAM erlaubt zudem eine Erweiterung durch diverse Java-Plug-ins. CAM wurde von mehreren Wissenschaftlern am Cambridge Engineering Design Centre entwickelt und zur Unterstützung von Forschungsprojekten eingesetzt. In diesen Forschungsarbeiten wurden insb. Modellierungs- und Analysemethoden für Systeme entwickelt, die eine gute Alternative zu anderen Diagrammwerkzeugen oder Tabellenkalkulationen für die Arbeit mit miteinander in Beziehungen stehenden Daten bieten (z. B. für Elemente und Relationen in Systemen).

DeMAID (s. [456]): Das von der NASA entwickelte IT-Werkzeug DeMAID (für engl. Design Manager's Aid for Intelligent Decomposition) unterstützt bei der Optimierung von PEPs. Dabei liegt der Fokus insb. auf der Verbesserung von vielschichtigen, iterativen Prozessen mittels der DSM-basierten Modellierung und Analyse dieser Prozesse. Eine aktuelle Version des Werkzeugs steht gegenwärtig nicht zur Verfügung.

Excel-Makros (s. [560]): Auch die auf MS Excel basierende Makros auf der Webseite dsmweb.org, die von Studierenden am MIT programmiert wurden, unterstützen beim Umgang mit gängigen DSM-Operationen (z. B. Partitionierung, Tearing, Banding und einfachen Simulationen). Allerdings fehlen unterstützende Visualisierungsmöglichkeiten. Grundsätzlich von Vorteil ist, dass durch die Verwendung der allgemein verfügbaren Standardsoftware MS Excel (bzw. alternativ OpenOfficeCalc und LibreOffice) diese Markos überall angewendet und in bestehende IT-Architekturen integriert werden können, was nach [312] zur Akzeptanz und Verbreitung in der Industrie führt. Auch in der vom Autor dieser Arbeit betreuten studentischen Arbeit [S58] wurde durch eine ausführliche, vergleichende Bewertung von CAM, Looemo und MS Excel gezeigt, dass MS Excel (mit entsprechenden Makros) bzgl. vieler Bewertungskriterien vorteilhafter als CAM und Looemo ist (z. B. Anpassbarkeit, Import und Export in IT-Werkzeuge).

Matlab-Makros (s. [548], [560]): Auch ein Matlab-Makro ist auf der Webseite dsmweb.org zu finden, welches der Clusterung von DSMs dient. Weitere Matlab-Makros zur Unterstützung von Analysen und Manipulationen von Matrizen sind auch denkbar. Auch vom Autor dieser Arbeit wurde in [P32] ein auf Basis von Matlab entwickeltes Software-Werkzeug zur graphbasierten Visualisierung von Relationen zwischen Produkteigenschaften und -merkmalen vorgestellt und in [P14] Matlab zur Analyse von Matrizen verwendet. Dieses Matlab-Werkzeug unterstützt Entwickler alle Relationen der MDM mit vier verschiedenen Ansichten zu visualisieren: Gesamtansicht über alle Produkteigenschaften und -merkmale sowie deren Relationen; Teilansicht auf die Relationen zwischen den Merkmalen; Teilansicht auf die Relationen zwischen Eigenschaften; Teilansicht auf die der Relationen zwischen Merkmalen und Eigenschaften.

Weitere akademische, frei verfügbare IT-Werkzeuge im Forschungsbereich der DSM sind der Webseite dsmweb.org [560] und der einschlägigen Literatur zu entnehmen. Diese sind i. d. R. sehr spezifisch, wie der Online-Algorithmus zur Triangularisierung zur Sequenzoptimierung von DSM-Matrizen [340], oder nur sehr rudimentär dokumentiert, wie z. B. das PSM₃₂ Programm [560].

Kommerzielle Softwarewerkzeuge

Acclaro DFSS (s. [198], [560]): Acclaro dient der Definition, Modellierung und Validierung von Anforderungen bis zu Lösungskonzepten. Als eine Art Werkzeugkasten umfasst Acclaro neben DSM-Methoden (z. B. Clustering, Partitioning, Tearing) das Axiomatic Design Quality Framework mit den DFSS-Qualitätsprozessen VOC, QFD, FMEA, TRIZ, die Pugh-Konzeptanalyse sowie Methoden zum Projektmanagement und zur Produktrisikobewertung. Dabei ist der Export von Daten und Reports in bspw. MS Project von Vorteil, von Nachteil sind neben den Kosten insb. die Visualisierungsmöglichkeiten sowie die Anpassbarkeit.

ADePT Design Software (s. [4], [560]): ADePT dient zum Management iterativer und informationsgetriebener Prozesse in Projekten mittels matrixbasierter Prozessmodellierung, -analysen und -visualisierungen (z. B. Tearing). Die Relationen zwischen den Prozessen werden bspw. in Form von Gantt-Diagrammen mit Primavera oder MS Project abgeleitet. Auch wenn der Fokus auf der Prozesskomplexität dient, lassen sich einige Ansätze dennoch für die anderen Domänen anwenden. Die begrenzten Visualisierungsoptionen, hohen Kosten und die eingeschränkten Import- und Exportmöglichkeiten sind besonders von Nachteil.

Lattix Plattform (s. [342], [560]): Die Lattix Plattform unterstützt v. a. durch DevOps-Ansätze und einer zentralisierten Ansicht beim Management von komplexem Softwaredesign und IT-Architekturen. Lattix dient zur effizienten Analyse von IT-Architekturen auf Grundlage von DSMs und auch MDMs und stellt diese visuell in Form von Matrizen dar, wobei graphbasierte Visualisierungsmöglichkeiten fehlen. Auch wenn der Fokus von Lattix auf Software Engineering liegt, können damit auch andere Domänen modelliert und analysiert werden. Hierfür sind sowohl die vielen Importmöglichkeiten (z. B. SQL, XML, MS Excel) als auch die vielen Matrixoperationen vorteilhaft. Neben den eingeschränkten Visualisierungsmöglichkeiten und den relativ hohen Kosten sind v. a. die schlechte intuitive Bedienbarkeit (insb. für eine kommerzielle Software) sowie der relativ hohe Einarbeitungsaufwand für Nutzer in der PE als negativ zu bewerten.

Loomeo (s. [357], [560]): Die Basis dieser Software bilden die generische DSM-Methodik sowie Ansätze aus der Graphentheorie. Der Fokus von Loomeo liegt auf der Modellierung, Visualisierung und Analyse komplexer Systemarchitekturen. Loomeo ist eines der sehr wenigen IT-Werkzeuge, die bei der Modellierung von Systemelementen und -relationen mit MDMs bestehend aus DSMs, DMMs explizit unterstützen. Für die Darstellung von Systemstrukturen stehen sowohl matrixbasierte als auch graphbasierte Visualisierungsansätze zur Verfügung. Zur Analyse (z. B. Clustering, Partitioning, strukturelle Analysen) dient ein Algorithmus- und Filter-Assistent, der bei der einfachen Erstellung benutzerdefinierter Analysen unterstützt. Von Nachteil sind neben dem verhältnismäßig hohen Preis und der geringen Weiterentwicklung v. a. auch, dass nur Matrizen und keine Graphen modelliert werden können und außerdem keine Möglichkeit besteht, DMMs als Graph anzuzeigen. Loomeo ist mit Java programmiert und kann nicht eigenständig erweitert werden.

Soley (s. [515], [560]): Der wesentliche Fokus von Soley liegt auf der Datenanalysen auf Basis der Graphentheorie. Zur Darstellung der Datenelemente und deren Relationen stehen neben den Matrizen auch verschiedene Graphen- und Diagrammlayouts zur Verfügung. Für die Analyse bietet Soley zwar einige Möglichkeiten, jedoch ist z. B. Clustering und Partitioning nur eingeschränkt möglich. Als Ergänzung zu Soley Desk dient das verhältnismäßig kostenintensive Soley Studio, welches die Entwicklung von Workflows und eigenen Meta-Modellen unterstützt und die anschließend in Soley Desk als App verwendet werden können. Dadurch kann Soley in einem gewissen Rahmen auch selbstständig angepasst werden. Während in der kostenlosen Version von Soley nur Daten aus MS Excel importiert

werden können, stehen weitere Datenformate für den Import- und Export (z. B. aus/in Datenbankformaten) in der kostenpflichtigen Version bereit.

Weitere kommerzielle IT-Werkzeuge auf Basis v. a. der DSM-Methodik sind auf der Webseite dsmweb.org [560] und in diversen, dort verlinkten Veröffentlichungen zu finden. Diese sind oft spezifisch für die Softwareentwicklung, wie z. B. NDepend zur Visualisierung von Abhängigkeiten im Softwarecode, oder spezifisch für die Projektplanung, wie z. B. ProjectDSM zur Kombination von konventionellen Projektmanagementmethode mit Mind Mapping und DSM, oder – ähnlich wie Soley – v. a. auf die Analyse von Abhängigkeiten zwischen Datenelementen fokussiert, wie z. B. die sog. BOXARR Plattform zur Lösung komplexer Abhängigkeiten zwischen Daten.

Diese und eigne, weitere akademische und kommerzielle IT-Werkzeuge zur matrixbasierten Modellierung, Visualisierung und Analyse wurden anhand einer Vielzahl von relevanten Kriterien (z. B. intuitive Bedienbarkeit, Schnittstellen für den Datenimport und -export, Anzahl und Güte von Clustering) und aus diversen Perspektiven (z. B. Produktmodellierung, Prozessvisualisierung, Wissensanalyse) evaluiert. Diese **Evaluierung der IT-Werkzeuge erfolgte weitestgehend auf Basis deren Anwendung in Fallbeispielen** durch den Autor dieser Arbeit selbst (s. v. a. in [P27] und für Loomio auch in [P16] und für Soley auch in [P34]) und vom Autor betreuten studentischen Arbeiten (s. u. a. [S3], [S5], [S6] [S8], [S9], [S11], [S16], [S23], [S25] [S29], [S37], [S53]). Eine Zusammenfassung dieser Evaluationsergebnisse dieser zuvor skizzierten IT-Werkzeuge ist in Bild 41 dargestellt. Darin ist die Vielzahl relevanter Kriterien in folgende sechs Gruppen von Kriterien kategorisiert worden: Benutzerfreundlichkeit (UX), Funktionsumfang (FU), IT-/Programmierschnittstellen (IT), Modellierung (MO), Visualisierung (VI), Analyse (AN). Dabei sei angemerkt, dass zum einen die Vergleichbarkeit dieser Software-Werkzeuge nur eingeschränkt möglich ist und die zusammengefassten Evaluierungsergebnisse nur eine qualitative Indikation darstellen können. Detailliertere Bewertungen mittels konkreter Kriterien sowie der Evaluation zugrunde liegenden Anwendungsbeispiele sind den zuvor angeführten Forschungsarbeiten zu entnehmen.

Software-Werkzeug	Akademisch/ kommerziell	Bewertung						Verweise zur Evaluationsbasis
		UX	FU	IT	MO	VI	AN	
Antares DSM	Akademisch							[402], [403], [560], [P27], [S3], [S8], [S9]
CAM	Akademisch							[621], [622], [623], [560], [P27], [S3], [S5], [S8], [S9], [S11], [S16], [S25], [S29], [S37]
DeMAID	Akademisch							[456], [560], [S3], [S16]
Excel-Makros	Akademisch							[312], [560], [P27], [S3], [S6], [S8], [S11], [S23], [S58]
Matlab-Makros	Akademisch							[548], [560], [P27], [P32], [P14], [S3], [S6], [S8], [S11], [S53]
Acclaro DFSS	Kommerziell							[198], [560], [P16], [P27], [S3], [S11]
ADePT Design software	Kommerziell							[4], [560], [P27], [S3], [S11]
Lattix Plattform	Kommerziell							[342], [560], [P16], [P27], [S3], [S11], [S16]
Loomio	Kommerziell							[357], [560], [P16], [P27], [P34], [S3], [S5], [S8], [S9], [S11], [S16], [S23], [S29], [S37], [S53]
Soley	Kommerziell							[515], [560], [P34], [S3], [S8], [S9], [S16], [S23], [S25], [S29]
Legende		sehr schlechte Erfüllung sehr gute Erfüllung						

Bild 41: Zusammengefasste Evaluationsergebnisse ausgewählter IT-Werkzeuge

4 Handlungsbedarfe und Ziele

„Wirkliches Neuland in einer Wissenschaft kann wohl nur gewonnen werden, wenn man an einer entscheidenden Stelle bereit ist, den Grund zu verlassen, auf dem die bisherige Wissenschaft ruht, und gewissermaßen ins Leere zu springen.“

WERNER K. HEISENBERG (1901 – 1976)

Nach dem Zitat von HEISENBERG, der für die Formulierung der Grundlagen der Quantenmechanik mit dem Nobelpreis für Physik ausgezeichnet wurde, können neuartige wissenschaftliche Erkenntnisse insb. dadurch gewonnen werden, indem die vorhandenen wissenschaftlichen Grundlagen (s. Kapitel 2) sowie der Stand der Forschung und Technik (s. Kapitel 3) an entscheidenden Stellen verlassen wird. Daher sollen in Kapitel 4.1 der Stand der Forschung und Technik zum einen aus Sicht der Wissenschaft in Kapitel 4.1.1 und zum anderen aus Sicht der industriellen Praxis (Wirtschaft) in Kapitel 4.1.2 diskutiert werden. Darauf aufbauend werden in Kapitel 4.2 **Ziele und Anforderungen der vorliegenden Arbeit** zum Komplexitätsmanagement in der PE abgeleitet.

4.1 Diskussion des Stands der Forschung und Technik

Wie in der Einleitung in Kapitel 1 aufgezeigt, stellt die PE ein komplexes System dar, welches aus unterschiedlichen Domänen bzw. Teilsystemen besteht (vgl. u. a. [355], [161]). Daher ist für das **Management der Komplexität in der PE die Modellierung des ganzen Systems notwendig**. Somit sind sowohl die Elemente der verschiedenen Domänen als auch deren Relationen in einem holistischen Systemmodell abzubilden. Besonders in Folge der VUCA-Rahmenbedingungen (s. Kapitel 1 und Anhang A.2) steigt der Bedarf an integrativen Methoden und IT-Werkzeugen zur Unterstützung des systemischen Denkens und Handelns. Auf diesen steigenden Bedarf weisen u. a. auch EHRENSPIEL & MEERKAMM hin, um zum einen Modelle für die diversen Domänen integrieren zu können und zum anderen Integrationsarbeit durch Generalisten leisten zu können (im Gegensatz zu Fach- bzw. Domänenspezialisten, vgl. Komplexitätsingenieur in Kapitel 1.4) [151], [149], [150].

Die im Stand der Forschung und Technik in Kapitel 3 beschriebenen Ansätze, Methoden und Werkzeuge zum Komplexitätsmanagement in der PE sind in den letzten Jahren zwar weiterentwickelt worden und (noch)

vielfältiger geworden, aber zur Unterstützung von Entwicklern (insb. von Komplexitätsingenieuren) **beim holistischen Komplexitätsmanagement in der PE nicht ausreichend**. So sind viele domänenspezifische Modellierungsansätze (s. Kapitel 3.1) vorhanden, die entweder im Hinblick auf die Modellierung von Systemelementen und deren Relationen zu abstrakt sind (z. B. phasen- und zustandsorientierte Ansätze) oder nur für bestimmte Domänen geeignet sind (z. B. CPM/PDD Ansatz).

Zudem gibt es nur **wenige, generell geeignete domänenübergreifende Modellierungsansätze** (s. Kapitel 3.2). Die neben dem matrixbasierten Modellierungsansatz (Kapitel 3.2.3) nach Auffassung des Autors beiden vielversprechendsten Ansätze sind das MBSE (v. a. SysML) (Kapitel 3.2.1) sowie das auf einer Ontologie basierende Systemmodell (Kapitel 3.2.2). Beide Ansätze wurden im Hinblick auf ihre Eignung zur Unterstützung für ein holistisches Komplexitätsmanagement evaluiert und v. a. aus den nachfolgenden Gründen als Grundlage für diese Arbeit verworfen.

Mit Hilfe von **SysML** kann ein Systemmodell aus sehr vielen verschiedenen Perspektiven betrachtet werden, um bspw. Informationsflüsse und hierarchische Strukturen darzustellen (vgl. Kapitel 3.2.1). Die SysML besteht aus neun Diagrammart, mit welchen die Modellierung von Elementen und deren Relationen auf vielfältige Art und Weise gut unterstützt wird. Zudem können die Diagrammart miteinander verbunden werden, indem in den einzelnen Diagrammen die Blöcke des Blockdefinitionsdiagramms referenziert werden. Dadurch kann aber nur ein begrenzter Überblick über das Gesamtsystem realisiert werden, welcher mit zunehmender Zahl an Diagrammen, Diagrammart und deren Größe stark beeinträchtigt wird. Überdies werden die Relationen zwischen den Diagrammart nur mittels Referenzierens von Elementen modelliert und sind daher nicht anschaulich darstellbar (v. a. im Vergleich zu graphbasierten Visualisierungsansätzen). Ein weiterer Nachteil ist, dass die Relationen in den Diagrammart zwar ähnlich (i. d. R. mit einem Pfeil) dargestellt werden, aber abhängig von der Diagrammart eine unterschiedliche Funktion bzw. Bedeutung aufweisen, was für diagrammübergreifende (i. S. v. domänen- bzw. teilsystemübergreifende) Modelle und Analysen problematisch ist. Aufgrund der Vielzahl und Vielfalt der zur Erstellung eines Modells für ein komplexes System erforderlichen Elemente und Relationen ist insb. die Modellierung sehr zeit- und kostenaufwendig und oft auch fehleranfällig. Dabei ist v. a. das Referenzieren der Systemelemente in den einzelnen Diagrammart bzw. die Überprüfung auf Vollständigkeit der Referenzen sehr aufwendig.

Mit Hilfe von **Ontologien** (i. S. v. speziellen Repräsentationssprachen wie z. B. OWL) kann ein Systemmodell durch die Externalisierung von Wissen über Elemente und deren Relationen erstellt werden (vgl. Kapitel 3.2.2). Somit kann durch geeignete Anwendungsprogramme das modellierte Wissen mittels intelligenter Suchanfragen effektiv, effizient und kontextbezogen zur Verfügung gestellt werden (vgl. u. a. Interferenz in Anhang A.17). Prinzipiell lassen sich mittels Ontologien auch Modelle für die verschiedenen Domänen der PE formal beschreiben und miteinander integrieren. Dazu wird in der Konzeptualisierung der Ontologie beschrieben, welche Attribute einzelne Klassen haben und mit welchen Relationen diese miteinander in Beziehung stehen (z. B. mit der Software Protegé). Zudem ist auch die Verknüpfung von Instanzen mit konkreten Elementwerten (z. B. geometrische Merkmale eines Rohrs) mittels Data Properties von Vorteil. Ein weiterer Vorteil ist die Modellierung von Relationen zwischen Klassen untereinander und zu Instanzen durch Object Properties. Allerdings ist u. a. der Aufwand für die erforderliche Definition von Klassen, Instanzen und Objektrelationen im Vergleich zu bspw. matrixbasierten Modellierungsansätzen deutlich höher. Außerdem können Relationen zwischen z. B. geometrischen Merkmalen nur sehr eingeschränkt modelliert und schwer interpretiert werden. Zudem sind ontologiebasierte Modelle von komplexeren Systemen aufgrund ihres strukturellen Aufbaus nur eingeschränkt anschaulich visualisierbar (v. a. auch im Vergleich zu den matrixbasierten Modellen). Daher sind die stark begrenzten Darstellungsmöglichkeiten von ontologiebasierten Modellen komplexer Systeme oft schwer verständlich (u. a. aufgrund ihrer Größe auch im Vergleich zu Visualisierungen von Matrizen) und v. a. auch von Anwendern ohne Domänenexpertise kaum zu interpretieren. Zudem sind auch die meisten in Kapitel 3.3.3 beschriebenen Analysemethoden nur bedingt auf ontologiebasierte Modelle übertragbar.

Für die bestmögliche Erreichung der in Kapitel 1.3 formulierten Zielsetzung dieser Arbeit sind die **Ansätze für das matrixbasierte Komplexitätsmanagement** (vgl. Kapitel 3.2.3) aufgrund der folgenden Gründe am besten geeignet. Bevor nachfolgend der Handlungsbedarf bzgl. matrixbasierter Modelle abgeleitet wird, werden ausgewählte Argumente für diese Auffassung aus abstrakter und allgemeingültiger Sicht skizziert:

- Da ein **System** grundsätzlich stets aus Elementen und Relationen besteht – unabhängig von dessen Domäne und Komplexität – (s. Kapitel 2.1) lässt sich dieses mathematisch am besten mittels einer Matrix beschreiben, womit dieses formalisiert und somit auch grundsätzlich rechnerinterpretierbar ist

- Alle vier Dimensionen, Grade und auch Arten von **Komplexität** (s. Kapitel 2.2 und Anhang A.7 und A.8) können grundsätzlich mit Hilfe von Matrizen modelliert werden
- In allen **Modellen** und Modellierungsansätzen sind mind. Elemente und deren Relationen zu finden, und zwar völlig unabhängig von deren Domäne und Komplexität (s. Kapitel 2.3 und Kapitel 3.1)
- **Matrixbasierte Modelle** (s. Kapitel 3.2.3) eignen sich sehr gut zum Navigieren durch den Modellraum (vgl. Kapitel 2.3.2) entlang der drei Dimensionen mit den sechs generellen Richtungen der PE
- **Matrixbasierte Ansätze**, Methoden und IT-Werkzeuge insb. zur Modellierung, Analyse und Visualisierung komplexer Systeme sind seit vielen Jahren sowohl Gegenstand der Forschung als auch der industriellen Praxis (s. Kapitel 3.2.3, Kapitel 3.3 und Anhang A.18)

Im Rahmen des matrixbasierten Komplexitätsmanagements lassen sich im Allgemeinen die beiden grundsätzlichen Vorgehensweisen bzw. **Phasen der Systemmodellierung und der Systembetrachtung** unterscheiden. Während in der Systemmodellierungsphase die MDM mit allen relevanten Partialmodellen inkl. der entsprechenden Elemente und deren Relationen erstellt wird, werden in der Systembetrachtungsphase die Charakteristika des gesamten Systems oder von Teilsystemen visualisiert, analysiert und bewertet. Dabei ist analog zum iterativen Wechsel zwischen Prozessen der Analyse (i. S. v. Zerlegen des Systems in seine Elemente) und der Synthese (i. S. v. Zusammenfügen von Elementen zu einem System) während des PEPs (vgl. Kapitel 2.3.2) auch das matrixbasierte Komplexitätsmanagement durch ein Hin- und Herspringen zwischen der Phase Systemmodellierung und Systembetrachtung geprägt. Daher wird auch bei der nachfolgenden Ableitung des Handlungsbedarfs grob zwischen diesen zwei Phasen differenziert. So werden zuerst die Handlungsbedarfe bzgl. der Systemmodellierung (i. S. v. Erstellung der erforderlichen Systemmodelle) und danach bzgl. der Systemvisualisierung, -analyse und -bewertung skizziert.

4.1.1 Forschungsbedarf aus Sicht der Wissenschaft

Die in Kapitel 3.3 und im Anhang A.18 erläuterten Ansätze, Methoden und IT-Werkzeuge für das matrixbasierte Komplexitätsmanagement sind für verschiedene Problem- und Aufgabenstellungen durchaus hilfreich. Allerdings bestehen aus Sicht der Wissenschaft noch **vielfältige Handlungsbedarfe bzgl. des matrixbasierten Komplexitätsmanagements**. Diese Bedarfe lassen sich im Hinblick auf die Zielsetzung der Arbeit (Kapitel 1.3) in folgende Handlungsfelder gliedern, die im Folgenden skizziert werden:

- **Modellierung von DSMs** auf Grundlage der bereits bestehenden Modelle aus den diversen Domänen (s. Kapitel 3.1)
- **Modellierung von DMMs** auf Grundlage der erstellten DSMs
- **Visualisierung** von einzelnen DSMs und DMMs sowie der MDM (s. Kapitel 3.3.2)
- **Analyse** und Bewertung von einzelnen DSMs und DMMs sowie der MDM (s. Kapitel 3.3.3)

Diese Handlungsfelder bauen aufeinander auf und sind in einem **Gesamtkonzept integrativ miteinander zu verknüpfen**, weshalb nachfolgend die Handlungsbedarfe je Feld kurz beschrieben werden. Dabei besteht auch ein hoher Bedarf nach Softwareunterstützung bei der Entwicklung einer Methodik mit geeigneten Modellierungs-, Analyse-, Visualisierungs- und Bewertungsmethoden (s. Kapitel 3.3.4), die Komplexitätsingenieure beim holistischen Komplexitätsmanagement in der IPE geeignet unterstützt.

Handlungsfeld Modellierung von DSMs

Domänenspezifische Modelle sind zur Unterstützung der IPE und für eine ganzheitliche, domänenübergreifende Systembetrachtung in ein holistisches Systemmodell zu überführen (s. Ziele der WiGeP, die den modernen Maschinenbau als Integrationswissenschaft auffasst [608], vgl. auch [156]). Im Hinblick auf die **matrixbasierte Modellierung** ergibt sich daraus der Bedarf, in einem ersten Schritt die je nach Modellzweck (s. Kapitel 2.3.1) relevanten, domänenspezifischen Modelle (s. Kapitel 3.1) in entsprechende DSMs zu überführen. Dabei soll nach EPPINGER & BROWNING eine Matrix nur bis zu 50 Elemente umfassen, um v. a. einen geringen Aufwand bei der Modellerstellung zu gewährleisten (s. Kapitel 3.2.3). Allerdings sind komplexe Systeme zum einen durch eine hohe Vielzahl und Vielfalt an Elementen und Relationen und zum anderen durch eine hohe Veränderlichkeit charakterisiert (s. Kapitel 2.2.2). Somit ist aufgrund der komplexen Systemmodelle deren Erstellung als auch deren bedarfsgerechte Aktualisierung so zu unterstützen, dass das Aufwand-Nutzen-Verhältnis positiv ist.

Aus diesem Grund besteht ein signifikanter **Forschungsbedarf bzgl. der geeigneten Transformation von domänenspezifischen Modellen in DSMs**. Die dafür erforderliche Akquisition von Daten, Informationen und z. T. Wissen wird in der Literatur als das Forschungsgebiet mit der größten Bedeutung und dem größten Forschungsbedarf angesehen (vgl. v. a. [623], [57], [88], [378]). Daher wurde z. B. von BIEDERMANN ET AL. ein Ansatz zur verbesserten Datenakquisition im Rahmen von Workshops für die Modellierung von (kleineren) MDMs erarbeitet [58]. Da Modelle von komplexen Systemen eine Vielzahl von DSMs umfassen können (bzw. müssen), um ein

holistisches Komplexitätsmanagement zu ermöglichen, sind zahlreiche Matrizen mit vielen Elementen mit deren Relationen zu erstellen. Die dafür benötigten Daten und Informationen sind aufgrund des Zeitbedarfs nicht nur durch Workshops oder Expertenbefragungen zu akquirieren. Zudem ist die nur manuelle Modellierung umfangreicher matrixbasierter Systemmodelle auch nach bisherigen Erfahrungen des Autors nicht nur zu zeit- und kostenaufwendig, sondern auch zu fehleranfällig (s. [339], [P16]). Allerdings konzentrieren sich bisher nur wenige Forschungsarbeiten (vgl. Kapitel 3.3.1 und [32], [131]) auf diesen Akquisitionsprozess [377], [378]. Auch daher ist nach BROWNING die größte Herausforderung der Umgang mit der Vielzahl an Elementen und Relationen, die für die Erstellung eines umfassenden Strukturmodells komplexer Systeme erforderlich sind [88]. Dabei hebt BROWNING das Fehlen einer geeigneten, assistierenden und benutzerfreundlichen Softwareunterstützung für die Modellierung, Manipulation und Analyse matrixbasierter Modelle hervor (vgl. Kapitel 3.3.4). Das Extrahieren erforderlicher Daten aus anderen Quellen zur schnellen Modellerstellung führt er als einen vielversprechenden Ansatz an [88].

Daher wird ein Konzept zur computerunterstützten Modellierung von DSMs auf Grundlage bestehender domänenspezifischer Modelle in dieser Arbeit entwickelt. Zur beispielhaften Veranschaulichung sind in Bild 42 für die vier Domänen Produkt, Prozess, Wissen und Organisation jeweils einige relevante Modelle mit einigen Elementen dargestellt.

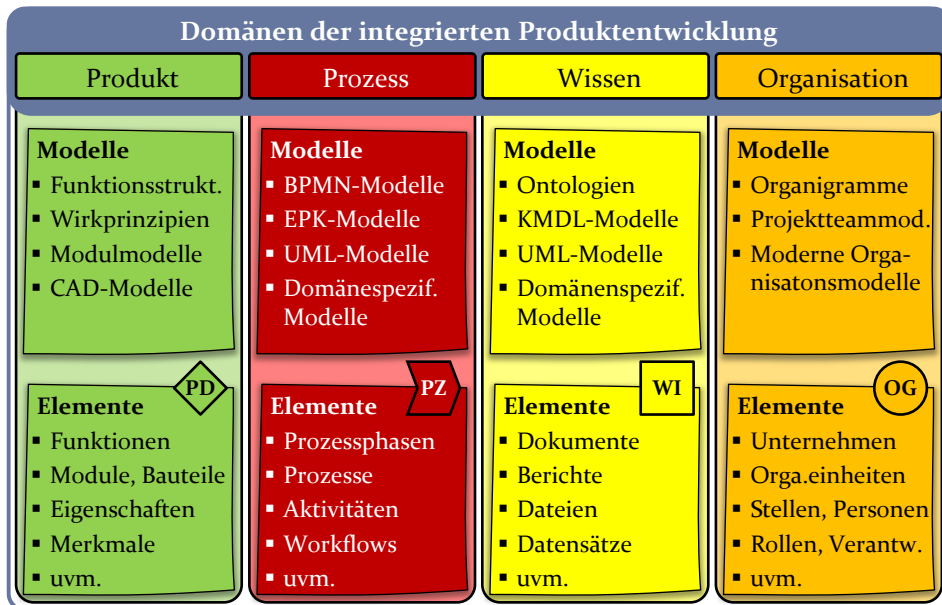


Bild 42: Überblick über beispielhafte domänenspezifische Modelle mit Elementen

Handlungsfeld Modellierung von DMMs

Neben den DSMs sind die entsprechenden DMMs für ein ganzheitliches Systemmodell (MDM) zu modellieren, da für das holistische Komplexitätsmanagement insb. die **domänenübergreifenden Relationen in den DMMs von großem Interesse** sind. Dabei sind sowohl die DMMs innerhalb einer Domäne (z. B. zwischen Produktmerkmalen und -eigenschaften der Produktdomäne) als auch zwischen Domänen (z. B. zwischen Anforderungen der Spezifikationsdomäne und Produktverhalten der Produktdomäne) zu betrachten. Jedoch gibt es zur Unterstützung für die Modellierung von DMMs kaum konzeptuelle Ansätze (vgl. Kapitel 3.3.1) und auch die wenigen explizit DMMs unterstützenden IT-Werkzeuge (s. Kapitel 3.3.4) sind wenig hilfreich, da diese entweder eine manuelle Erstellung oder den Import der Matrizen erfordern. Daher sind analog zur Erstellung von DSMs auch bei der Modellierung von DMMs der Zeit- und Kostenaufwand sowie die Fehleranfälligkeit durch geeignete Konzepte sowie mittels Computerunterstützung zu minimieren.

Handlungsfeld Visualisierung und Kommunikation

Während Matrizen für die Modellerstellung – v. a. in der computerunterstützten Modellierung – einen großen Nutzen haben, sind graphbasierte Repräsentationsformen v. a. bei der Modellanalyse sowie bei der Modellnutzung und -kommunikation im Vergleich zu matrixbasierten Darstellungen dienlicher (s. Kapitel 3.3.2 und A.15). Daraus lässt sich folgern, dass neben der Computerunterstützung der Matrixerstellung auch deren **computerunterstützte Repräsentation mit Hilfe von graphbasierten Visualisierungsansätzen** von großer Bedeutung ist (s. Kapitel 3.3.2). Dies liegt daran, dass mittels der Darstellung in Form von Graphen Entwickler und Komplexitätsingenieure wesentlich schneller und intuitiver besondere Charakteristika eines Systemmodells erkennen können (z. B. Auswirkungen auf Systemelemente in Folge der Änderung eines Elements oder dessen Wert).

Bisherige IT-Werkzeuge zum Komplexitätsmanagement unterstützen nur z. T. mit Hilfe von verschiedenen Graphen-Layouts (s. Kapitel 3.3.4). Besonders viele und hilfreiche Möglichkeiten zur Visualisierung von Matrizen haben v. a. die kommerziellen Softwarewerkzeuge Loomio und Soley. Die besten Visualisierungsmöglichkeiten der akademischen IT-Werkzeuge bietet CAM, wobei diese bzgl. der möglichen Graphen-Layouts stark eingeschränkt sind und die nutzbaren Graphen wenig geeignet und nutzerfreundlich sind. Alle IT-Werkzeuge ermöglichen lediglich die Nutzung einer kleinen Auswahl der relevanten und hilfreichen in Kapitel 3.3.2

dargestellten Graphen-Layouts, die zudem nicht oder nur in geringem Ausmaß den Nutzerbedürfnissen anpassbar sind. Demzufolge besteht ebenfalls erheblicher **Forschungsbedarf bzgl. geeigneter Visualisierungsansätze von DSMs, DMMs und insb. komplexer MDMs**. Dabei ist die wesentliche Herausforderung die Lesbarkeit eines Graphen aufgrund sehr vieler zu visualisierender Elemente (bzw. Knoten) und Relationen (bzw. Kanten) sicherzustellen und zu verbessern.

Handlungsfeld Analyse und Bewertung

Der zentrale Zweck der matrixbasierten Modellierung ist die Analyse dieser Modelle mittels mathematischer Ansätze, um auf Grundlage der gewonnen Ergebnisse und Erkenntnisse Schlussfolgerungen für die Verbesserung der modellierten Systeme zu erhalten. Daher gibt es bereits eine Vielzahl von **mathematischen Ansätzen, die sich explizit für die Analyse von Matrizen eignen** oder sich auf diese übertragen lassen. Die in Kapitel 3.3.3 erläuterten bestehenden Analysemethoden eignen sich sehr gut für das Komplexitätsmanagement in der PE. So unterstützt das Clustering bei der Verbesserung von modularen Produktstrukturen auf Basis der Relationen in und zwischen Teilsystemen. Die zeitbasierten DSMs dienen dazu, Partitioning-, Banding- und Tearing-Prozesse effizienter zu gestalten, indem die optimale bzw. logische Abfolge von Prozessschritten ermittelt sowie Möglichkeiten für Simultaneous Engineering aufgezeigt werden.

Allerdings besteht auch **bzgl. mathematischer Analyseansätze noch ein signifikanter Forschungsbedarf**. Von den kommerziellen IT-Werkzeugen haben lediglich Loomio und Lattix Plattform geeignete Analyseverfahren. Daher ist es auch nicht verwunderlich, dass viele Wissenschaftler und Praktiker als auch der Autor dieser Arbeit und von ihm betreute Studierende zur Analyse von Matrizen oft auf Excel- oder Matlab-Makros zurückgegriffen haben (s. Kapitel 3.3.4). Aus Sicht des holistischen Komplexitätsmanagements sind insb. Clusterverfahren für die computerunterstützte Anwendung zur Analyse von DSM, DMM und MDM zu verbessern. So kann bspw. mit CAM nur für eine binäre, quadratische Matrix ein Clustering durchgeführt werden, und zwar ohne explizite Clusterüberschneidung und ohne die Möglichkeit der Angabe einer Clusteranzahl (vgl. Vorteile anderer Clusterverfahren, wie z. B. k-Means-Algorithmus, Clustering-Algorithmus von ZAKARIAN, IGT-Algorithmus, vgl. Anhang A.20 für Details). Neben diesen Clusterverfahren sind für ein ganzheitliches Komplexitätsmanagement auch Ansätze zur holistischen Analyse erforderlich, die weitgehend noch nicht computerunterstützt werden. Dazu gehören Analysen der Auswirkungen in Folge von Änderungen (vgl. Kapitel 3.3.3 und Anhang A.20).

Dadurch kann der Komplexitätsingenieur kritische Elemente, die bei einer Änderung erhebliche oder bestimmte Auswirkungen im gesamten System auslösen (können), besser identifizieren. Bei Bedarf kann er diese Änderung vermeiden, um das Ausmaß der Änderungsweitergabe bzw. -fortpflanzung und somit Komplexitäts- und Änderungskosten zu verhindern. Hierfür eignet sich z. B. die Change Prediction Methode (s. Anhang A.20), die nur in CAM implementiert ist. Weiterer Verbesserungsbedarf besteht bzgl. der Computerunterstützung der folgenden domänenübergreifenden Analysemethoden: Vorwärts- und rückwärtsgerichtete Analyse, Partitioning mit Änderungsanfälligkeit und Banding einer MDM (s. Kapitel 3.3.4).

In den bisher bestehenden IT-Werkzeugen (s. Kapitel 3.3.4) können z. T. auch die Richtungen und Stärken von Relationen modelliert und damit analysiert werden (z. B. Loomeo). Für eine (noch) bessere mathematische Analyse der Relationen zwischen z. B. Anforderungen, Produktmerkmalen, und -eigenschaften besteht insb. noch **Forschungsbedarf bzgl. einer mathematischen Quantifizierung von Relationen.**

4.1.2 Handlungsbedarf aus Sicht der Wirtschaft

Neben den Forschungsbedarfen aus Sicht der Wissenschaft sind auch die **Handlungsbedarfe aus Sicht der Wirtschaft wichtig**, da u. a. die WiGeP die Intention hat, entscheidende Impulse für die Entwicklung von Produkten für die zukünftigen Märkte sowohl durch Forschung als auch durch Austausch mit der Wirtschaft zu geben [608]. Demzufolge sollen auch die Anforderungen aus der industriellen Praxis zur (Weiter-)Entwicklung von Konzepten und IT-Werkzeugen berücksichtigt werden.

Aufgrund der steigenden VUCA-Rahmenbedingungen (s. Kapitel 1) nimmt die Bedeutung des Komplexitätsmanagements in der PE in Unternehmen zu. Dies lässt sich u. a. an der Zunahme an Stellen, die EHRENSPIEL & MEERKAMM als Integratoren bezeichnen (s. Anhang A.2), belegen (z. B. Systemingenieure, (IT-)Systemarchitekten). Übertragen auf das Komplexitätsmanagement entspricht dies der Tätigkeit des Komplexitätsingenieurs (s. Kapitel 1.4). Nach LINDEMANN ET AL. lässt sich dies zusammenfassen mit Erfassen und Bewerten, Vermeiden und Reduzieren sowie Beherrschen und Kontrollieren von Komplexität [355] (s. [252] und Anhang A.10). Dabei sind für den Unternehmenserfolg die Erfassung der Auswirkungen von Komplexität und darauf aufbauend die **Vermeidung und Reduzierung der Komplexitätskosten** relevant (z. B. mittels Modularisierung von Produkten und Prozessen, s. Anhang A.9). Die sich daraus ergebenden Handlungsbedarfe aus Sicht der Wirtschaft werden nachfolgend skizziert.

Handlungsfeld Zeit- und Kostenreduzierung

Die bereits in Kapitel 4.1.1 erwähnte **zeit- und damit kostenaufwendige manuelle Erstellung matrixbasierter Modelle** muss insb. aus Unternehmensperspektive minimiert werden. Zur Veranschaulichung des dafür erforderlichen Aufwands soll das in Kapitel 1.5 skizzierte Anwendungsszenario mit nur 16 Produkt- und sechs Spezifikationselementen dienen (s. Kapitel 1.5 und Anhang A.4). Unter der Annahme, dass für das Identifizieren der 16 Produktelemente v. a. im CAD-Modell und für das Übertragen in das matrixbasierte Modell jeweils eine Minute und für die sechs Spezifikationselemente in diversen Systemen im Durchschnitt jeweils vier Minuten benötigt werden, erfordert allein das Erfassen der Elemente 40 Minuten. Zudem ergeben sich bei 22 Elementen 462 (d. h. $22 * 22 - 22$) mögliche Relationen zwischen den Elementen. Sofern zur Modellierung bzw. Überprüfung einer (möglichen) Relation durchschnittlich eine Minute benötigt wird, erfordert allein die Modellierung der Relationen fast acht Stunden. Daher ist für die gesamte Modellerstellung ein Mitarbeiter einen Tag lang beschäftigt. Dadurch wird der Bedarf der Wirtschaft für eine computerunterstützte und v. a. (teil-)automatisierte Erstellung von Modellen deutlich.

Zudem drängt sich die Frage auf, warum die in der PE digital modellierten Elemente (z. B. Produktmerkmale inkl. Symbole, Einheiten und Werte, s. Kapitel 1.5 und Anhang A.4) in entsprechenden IT-Systemen (z. B. CAD-Systemen) nicht auch **in die matrixbasierten Modelle für das Komplexitätsmanagement übernommen** werden können. Dies ist nicht nur sinnvoll, um Zeit und Kosten zu sparen, sondern auch um schnell und agil die matrixbasierten Modelle bei Änderungen in entsprechenden domänenspezifischen Modellen aktualisieren zu können.

Aus der Beispielrechnung wird aufgrund der **Vielzahl an Elementen und v. a. Relationen die Fehlerwahrscheinlichkeit** erkennbar. So können z. B. Übertragungsfehler auftreten, indem Elemente oder Relationen einer falschen Reihe bzw. Spalte zugeordnet werden. Dabei wird bereits bei Fehlen oder Setzen einer falschen Relation die Aussagekraft des ganzen Systemmodells signifikant eingeschränkt und die Gefahr von falschen Schlussfolgerungen und Fehlentscheidungen erhöht.

Handlungsfeld Adaptier- und Erweiterbarkeit

Im Rahmen des Komplexitätsmanagement ist es für Unternehmen wichtig, dass die genutzten Methoden und assistierenden IT-Werkzeuge sich den **veränderten Rahmenbedingungen in diversen Domänen anpassen und für weitere Anwendungsbereiche erweitern lassen**. So kommt es

häufig vor, dass eine neue Methode zusammen mit einer entsprechenden Software zuerst für eine Domäne eingeführt und sukzessive auf weitere Domänen ausgerollt wird. Dabei ist es erforderlich, dass das IT-Werkzeug zum einen i. S. eines holistischen Komplexitätsmanagements die verschiedenen domänenspezifischen Modelle integrativ in ein Gesamtmodell aufnehmen und zum anderen auch auf die domänenspezifischen Charakteristika adaptiert und erweitert werden kann. Hierfür sind v. a. geeignete Schnittstellen zu den jeweiligen IT-Systemen der Domänen für die computerunterstützte Akquise von Elementen und Relationen zur Erstellung der matrixbasierten Modelle essenziell. Dies soll vereinfacht in Bild 43 anhand ausgewählter IT-Systeme diverser Domänen dargestellt werden.

Domänen der integrierten Produktentwicklung			
Produkt	Prozess	Wissen	Organisation
IT-Systeme mit Fokus auf CAD-Systeme ▪ Catia ▪ PTC Creo ▪ Siemens NX ▪ u. v. m.	IT-Systeme mit Fokus auf Prozessmodellierungssys. ▪ Adonis ▪ Aris ▪ Modelio ▪ u. v. m.	IT-Systeme mit Fokus auf PDM-S. und Laufwerke ▪ Aras Innovator ▪ Laufwerke ▪ u. v. m.	IT-Systeme mit Fokus auf Organ.-modellierungssys. ▪ Aeneis ▪ Aris ▪ Modelio ▪ u. v. m.

Bild 43: Überblick einiger domänenspezifischer Anwendungs- bzw. Autorensysteme

Handlungsfeld Nützlichkeit und Einfachheit

Die Nützlichkeit ergibt sich aus **einer effektiven, effizienten und zweckdienlichen Anwendung der Methode mit dem IT-Werkzeug** für die Ziele des Komplexitätsmanagements (s. Kapitel 1.3 und Anhänge A.6, A.7, A.8) und einem positiven Aufwand-Nutzen-Verhältnis. Mit Einfachheit sind aus Unternehmenssicht v. a. die intuitive Nutzung, hohe Benutzerfreundlichkeit der Funktionalitäten sowie die geringe Einarbeitungszeit zur Modellierung, Visualisierung und Analyse gemeint (vgl. Kapitel 3.3.4). Zudem muss das assistierende IT-System auch manuelle Änderungen am Softwarecode mittels verbreiteter Programmiersprachen (z. B. Python) ermöglichen und für kleine Modellierungs- und Analyseaufgaben den Import und Export via MS Excel und ähnlich gängiger Datenformate erlauben. Da Unternehmen möglichst wenige IT-Werkzeuge verwenden möchten, muss das IT-Werkzeug zwar alle erforderlichen Funktionalitäten bereitstellen, aber dennoch so offen und modular aufgebaut sein, dass weitere, für bestimmte Zwecke geeignetere IT-Systeme verbunden werden können.

4.2 Ziele und Anforderungen

Wie die Ausführungen in der Ausgangssituation und der Problemstellung aufgezeigt haben (s. Kapitel 1), nimmt die **Bedeutung des Managements komplexer Systeme insb. in der integrierten Entwicklung weiter zu**. Aufbauend auf einem tiefgreifenden Verständnis für die Modellierung komplexer Systeme in Folge der Ausführungen in Kapitel 2 hat sich gezeigt, dass die in Kapitel 3.1 umrissenen diversen domänenspezifischen Modelle für das Management komplexer Systeme lediglich für bestimmte Problem- und Aufgabenstellungen innerhalb einer Domäne hilfreich sind. Da für ein ganzheitliches Komplexitätsmanagement integrierende bzw. domänenübergreifende Ansätze, Methoden und Softwarewerkzeuge dienen, sind drei dieser domänenübergreifenden Ansätze in Kapitel 3.2 vorgestellt und in Kapitel 4.1.1 diskutiert worden. Aufgrund der besten Eignung für die Zielsetzung dieser Arbeit (Kapitel 1.3) sind die wesentlichen Ansätze für das matrixbasierte Komplexitätsmanagement in der PE in Kapitel 3.3 zu finden.

Allerdings hat die Diskussion des derzeitigen Stands der Forschung und Technik offenbart, dass aus Sicht der Wissenschaft als auch aus Sicht der industriellen Praxis noch erheblicher **Handlungsbedarf bzgl. der Weiterentwicklung bestehender Ansätze als auch der Entwicklung neuer Ansätze und IT-Werkzeuge** besteht (vgl. Kapitel 4.1, v. a. auch [88]). Aus akademischer Sicht sind Verbesserungspotentiale sowohl bzgl. der matrixbasierten Modellierung, graphbasierten Visualisierung als auch der mathematischen Analyse und Bewertung aufgezeigt worden (vgl. Kapitel 4.1.1), wobei bzgl. der Erstellung von DSMs auf Basis bereits bestehender domänenspezifischer Modelle (z. B. CAD-Modelle) und der anschließenden Verknüpfung der Elemente verschiedener Domänen mittels DMMs noch besonders viel Verbesserungspotenzial vorliegt. Dies steht auch im Einklang mit den Bedürfnissen der Wirtschaft (vgl. Kapitel 4.1.2), welche eine deutliche Aufwandsreduzierung (v. a. bei der Modellerstellung) und eine flexible Adaptier- und Erweiterbarkeit von Methoden und IT-Werkzeugen fordert. Daher lässt sich schlussfolgern, dass aus wissenschaftlicher als auch aus wirtschaftlicher Sicht der Komplexitätsingenieur für seine immer wichtiger werdende Aufgabe des holistischen Komplexitätsmanagements mit Methoden und IT-Werkzeugen besser zu befähigen ist, matrixbasierte Modelle komplexer Systeme zu erstellen, graphbasiert zu visualisieren und darauf aufbauend mathematisch zu analysieren und zu bewerten.

Daher werden in Kapitel 4.2.1 die **Forschungsfragen und Ziele der Arbeit** skizziert. Anschließend werden die Anforderungen an die zu erforschende Methodik und das zu entwickelnde Konzept in Kapitel 4.2.2 erläutert.

4.2.1 Forschungsfragen und Ziele

Aus den dargestellten Herausforderungen und in Folge dieser Zielsetzung lassen sich u. a. die folgenden, übergeordneten **Fragestellungen** ableiten:

- Wie können die Elemente und deren Relationen in und zwischen den Domänen Anforderungen, Produkt, Prozess, Wissen, Organisation und Kosten geeignet modelliert, analysiert, visualisiert und bewertet werden, um beim holistischen Komplexitätsmanagement in der PE zu unterstützen?
- Wie können die Änderungen und deren Auswirkungen auf bestimmte Systemelemente in Folge der Modifikation eines bestimmten Systemelements geeignet modelliert, analysiert, visualisiert und bewertet werden, um bei einer zielorientierten Entscheidungsfindung zur Erfüllung von Spezifikationen sowie zur Realisierung von Kosten zu unterstützen?

Das sich daraus ergebende **wissenschaftliche Ziel** dieser Arbeit ist ein Konzept zum besseren, holistischen Management von Komplexität, um Entwickler mit Methoden zur Modellierung, Analyse, Visualisierung und Bewertung von komplexen Systemen zu unterstützen. Dazu wird insb. erforscht, wie sich unterschiedliche Modelle aus den Domänen Produkt, Prozess, Wissen und Organisation verknüpfen (z. B. durch matrixbasierte Ansätze), analysieren (z. B. durch mathematische Algorithmen), visualisieren (z. B. durch auf Graphen basierende Layouts) und bewerten (z. B. durch multikriterielle Bewertungsmethoden) lassen. Der wissenschaftliche Anspruch ist die Entwicklung einer Methodik mit jeweils geeigneten Methoden zur Modellierung, Analyse, Visualisierung und Bewertung, welche Entwickler bei der systematischen Komplexitätsbeherrschung in den unterschiedlichen, integrierten Domänen der PE als auch bei der systematischen Entscheidungsfindung in den verschiedenen Entwicklungsphasen unterstützt. Dabei dient das prototypisch implementierte Assistenzsystem zur Unterstützung der Anwender bei der Nutzung dieser Methoden.

Die **wirtschaftliche Zielsetzung** bzw. anwendungsbezogene Zielsetzung dieser Arbeit ist eine Methodik zum Komplexitätsmanagement, die sich auf die jeweilige Entwicklungssituation flexibel, kontextspezifisch und zielgerichtet adaptieren lässt. Da die Unternehmen durch VUCA geprägt werden, muss die entwickelte Methodik i. S. v. VUCA-Prime (s. Anhang A.2) Entwickler bei der zielorientierten Vorgehensweise (z. B. Kaskadieren von Zielen), bei der Erarbeitung eines systematischen Verständnisses (z. B. Verstehen von Ursache- und Wirkungsbeziehungen), bei der zweckmäßigen Transparenzschaffung (z. B. Modellieren von Systemelementen) und bei

der agilen Anpassung an sich verändernde Rahmenbedingungen (z. B. Ändern von internen und externen Elementen) unterstützen [23], [24]. Dadurch soll die Komplexität in der PE verringert und beherrscht werden, was zu einer zentralen Herausforderung für Unternehmen geworden ist.

4.2.2 Anforderungen an das zu entwickelnde Konzept

Zur Beantwortung dieser Forschungsfragen und Erreichung dieser Ziele (s. Kapitel 4.2.1) **wird in der vorliegenden Arbeit ein Beitrag geleistet, indem eine Methodik zur Modellierung, Visualisierung, Analyse und Bewertung von v. a. komplexen Systemen erforscht und entwickelt wird.** Dazu ist aufbauend auf dem in dieser Arbeit erkenntnistheoretisch erforschten, tiefgreifenden Verständnis von zu modellierenden, komplexen Systemen in der PE (vgl. Kapitel 2 und Anhang A.5, A.6, A.7, A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15, A.16) ein umfassendes Konzept für das holistische Komplexitätsmanagement in der PE zu konzipieren. Zur Demonstration und Nutzung sowie für die effiziente Anwendung der Methodik ist zudem auch ein Fachkonzept für ein Assistenzsystem zu entwickeln und dieses zu implementieren (vgl. [169]).

Aus den in Kapitel 4.1 erläuterten Handlungsbedarfen lassen sich die folgenden **Anforderungen an die Methodik und das Assistenzsystem zum holistischen Komplexitätsmanagement** in der PE ableiten. Dabei werden zuerst zentrale, funktionale Anforderungen aus den vier Handlungsfeldern aus Wissenschaftssicht zusammengefasst und danach um nicht-funktionale Anforderungen ergänzt.

Anforderungen aus dem Handlungsfeld Modellierung von DSMs

Wie die Diskussion des Stands der Forschung und Technik in Kapitel 4.1 gezeigt hat, unterstützen weder kommerzielle noch akademische Softwarewerkzeuge bei der (teil-)automatisierten Akquisition von Elementen und Relationen im Rahmen der Erstellung matrixbasierter Modelle – was insb. aufgrund von Zeit- und Kostengründen aus Sicht der Wirtschaft wichtig ist. Daher lassen sich u. a. folgende **Anforderungen an die Lösung** ableiten:

- Die Lösung muss Funktionalitäten zur Navigation entlang der drei Dimensionen des Modellraums (d. h. Konkretisierung, Zerlegung, Variation, s. Bild 14) aufweisen (vgl. Systemanalyse und -synthese)
- Die Lösung muss geeignete Möglichkeiten zur Manipulation der Modelle (v. a. von Elementen, Relationen und deren Werte) bieten
- Die Lösung muss skalierbar sein, d. h. v. a. für die unterschiedlichen Komplexitätsstufen anwendbar sein (s. Anhang A.6)

- Die Lösung muss flexibel sein, d. h. für alle domänenspezifischen Modelle (s. Kapitel 3.1) adaptierbar sein
- Die Lösung muss IT-Schnittstellen für einen (teil-)automatisierten Import von domänenspezifischen Modellen bereitstellen
- Die Lösung muss zudem Funktionen für eine (teil-)automatisierte Modellierung der Elemente sowie deren Relationen bieten

Anforderungen aus dem Handlungsfeld Modellierung von DMMs

Aufbauend auf den computerunterstützt erstellten DSMs für die diversen Domänen sind Methoden zu entwickeln, die auch bei der Modellierung der bestehenden Relationen zwischen Elementen unterschiedlicher, domänenspezifischer Modelle unterstützen. Dabei besteht oft die Herausforderung, dass die Relationen zwischen Elementen unterschiedlicher Domänen nicht automatisiert in das Assistenzsystem geladen werden können. Allerdings entsteht erst durch die Erstellung dieser DMMs ein holistisches Modell der IPE in Form einer MDM. Infolgedessen sind für die Modellierung dieser domänenübergreifenden Relationen in den DMMs entsprechende, computerunterstützte Funktionalitäten zu entwickeln. Dadurch ergeben sich u. a. folgende **Anforderungen an die Methodik**:

- Die Lösung muss sowohl matrix- als auch graphbasierte Funktionen zur Modellierung der unterschiedlichen Relationen zwischen den Elementen verschiedener Domänen zur Verfügung stellen
- Die Lösung muss Methoden für eine systematische Modellierung der Relationen zwischen Elementen verschiedener Domänen zur Verfügung stellen, um Aufwand und Fehler zu minimieren
- Die Lösung muss Konzepte für die Modellierung von objektiven und subjektiven sowie quantifizierten und qualitativen Relationen zur Verfügung stellen
- Die Lösung muss Funktionen zur matrix- als auch graphbasierten Manipulation von DMMs zu jedem Zeitpunkt im PEP bereitstellen

Anforderungen aus dem Handlungsfeld Visualisierung

Während v. a. zur Modellerstellung und teils auch zur Modellanalyse und -bewertung matrixbasierte Methoden zweckdienlich sind, unterstützen die Methoden für die graph(en)basierten Visualisierung v. a. bei der Modellnutzung und -kommunikation (vgl. Anhang A.19). Daher ergeben sich u. a. folgende **Anforderungen an die Lösung**:

- Die Lösung muss Funktionen zur matrix- und graph(en)basierten Visualisierung als auch zur Transformation bereitstellen

- Die Lösung muss Visualisierungsfunktionalitäten zur Transparenz und Nachvollziehbarkeit der Relationen zwischen den Elementen in einem sowie zwischen mehreren Systemmodell(en) ermöglichen (d. h. intra- und inter-Domänen-Relationen)
- Die Lösung muss geeignete Visualisierungsmöglichkeiten für die Darstellung des strukturellen Aufbaus (z. B. von Clustern) bieten
- Die Lösung muss differenzierte Visualisierungsoptionen zur geeigneten Darstellung u. a. der diversen DSMs (s. Kapitel 3.2.3) bieten
- Die Lösung muss verschiedene Graphen-Layouts (s. Kapitel 3.3.2) zur kontextsensitiven und zweckmäßigen Visualisierung bieten
- Die Lösung muss Exportmöglichkeiten zur Visualisierung mittels geeigneter Visualisierungssoftwaresysteme bereitstellen
- Die Lösung soll die Auswirkung(en) in Folge von Modifikationen eines Elements i. S. einer Änderungsfortpflanzung mit Hilfe einer geeigneten Visualisierung unterstützen (z. B. durch Filtern von Elementen, selektives Anzeigen zusammenhängender Elemente)
- Die Lösung soll Visualisierungsmöglichkeiten zur Darstellung der Art der Relationen (z. B. gerichtet/ungerichtet, komplementär/konfliktär) zur Verfügung stellen

Anforderungen aus dem Handlungsfeld Analyse und Bewertung

Die Untersuchung der Softwarelösungen (Kapitel 3.3.4) hat verdeutlicht, dass IT-Werkzeuge zur matrixbasierten Analyse nicht oder nur begrenzt neben DSMs auch DMMs und MDMs analysieren können. Zudem sind in den meisten Softwaresystemen lediglich einige wenige Analyseverfahren für Matrizen implementiert. Aus diesem Handlungsbedarf lassen sich u. a. folgende **Anforderungen an die Lösung** ableiten:

- Die Lösung muss Analyseverfahren zur Durchführung individueller Analysen bestimmter Domänen zur Verfügung stellen
- Die Lösung muss Analyseverfahren zur Durchführung holistischer, domänenübergreifender Analysen zur Verfügung stellen
- Die Lösung muss Exportmöglichkeiten für die weitere Analyse, Optimierung und Bewertung bereitstellen
- Die Lösung soll möglichst viele gängige Analyseverfahren (s. Kapitel 3.3.3) für matrixbasierte Modelle für die Anwender bereitstellen
- Die Lösung soll Möglichkeiten zur Analyse von Relationen, die mit mathematischen Funktionen modelliert sind, bieten
- Die Lösung soll Methoden zur Unterstützung der multikriteriellen Bewertung bieten (z. B. für die Manipulation von Elementen)

Weitere, nicht-funktionale Anforderungen

Neben den funktionalen Anforderungen sind auch weitere Anforderungen an die Methodik und deren Implementierung in ein IT-Werkzeug zu berücksichtigen. Dazu gehören u. a. diese **Anforderungen an die Lösung**:

- Die Lösung muss ein benutzerfreundliches Graphical User Interface (GUI) zur intuitiven Bedienung und effizienten Nutzung bieten
- Die Lösung muss stabil und zuverlässig funktionieren
- Die Lösung muss frei verfügbar sein (i. S. v. kostengünstig) und einfach in die IT-Architektur von Unternehmen integrierbar sein
- Die Lösung muss einen modularen Aufbau zur einfachen Erweiterbarkeit um weitere Methoden (z. B. SIPOC, RASCI) als auch um Softwaremodule (z. B. Excel-Module, Python-Bibliotheken) haben
- Die Lösung muss passende Schnittstellen zu anderen IT-Werkzeugen zum Import und Export von Modellen zur Verfügung stellen
- Die Lösung muss gängige Datei- und auch Datenformate verwenden und geeignete Notationsformen bereitstellen
- Die Lösung soll sich an einem Referenzmodell und Metamodell (s. Anhang A.15) orientieren, welche als Basis für ein einheitliches und gemeinsames Verständnis der konzeptuellen Modellierung dienen

5 Konzept für ein Assistenzsystem zum Komplexitätsmanagement in der PE

„Wir können gar nicht anders erfolgreich denken und handeln als methodisch.“

KLAUS EHRENSPIEL (geb. 1930)

Durch das Zitat von EHRENSPIEL, einem der bekanntesten Wissenschaftler der methodischen und integrierten PE, wird die Notwendigkeit einer methodischen Vorgehensweise insb. aufgrund der Beschränktheit des menschlichen Gedächtnisses und Handelns im Hinblick auf die steigende Komplexität (in) der PE deutlich [148]. Auch wenn EHRENSPIEL unter dem Begriff Methodik die gesamte Bandbreite von implizit im Unbewussten ablaufenden Methoden über Vorgehens- und Analysemethoden bis hin zu computerunterstützten bzw. implementierten Methoden subsumiert, steht im Fokus dieses Kapitels die **Methodik zum holistischen, matrixbasierten Komplexitätsmanagement in der IPE**.

Die in diesem Kapitel **vorgestellte Methodik ist der Kern dieser Arbeit**. Die Erforschung der erforderlichen Grundlagen für diese Methodik sowie deren darauf aufbauende, prototypische Implementierung in ein Assistenzsystem erfolgte gemäß dem V-Modell®XT, dem deutschen Referenzmodell zur Entwicklung von IT-Systemen (s. Bild 44 für eine vereinfachte Darstellung, s. für Details [16]). Die Bezeichnung V-Modell®XT leitet sich aus dem V-förmigen Schema des Vorgehensmodells und dem Zusatz eXtreme Tailoring (XT) ab (d. h. flexibel an Projektspezifika adaptierbar). Das V-Modell®XT beruht auf den vier Prinzipien „Spezifikation und Zerlegung“ (vgl. Analyse), „Realisierung und Integration“ (vgl. Synthese), „Verifikation und Validierung“ und „Iterationen und Inkremente“. Dabei ist der Schritt der Anforderungsfestlegung bereits in Kapitel 4.2.2 auf Grundlage des abgeleiteten Handlungsbedarfs in Kapitel 4.1 dargelegt. Für die Vorstellung der Methodik mit ihrem assistierenden IT-Werkzeug wird zuerst in Kapitel 5.1 auf Basis des Gesamtkonzepts der Entwurf des Gesamtsystems mit der Systemarchitektur erläutert, woraus auch die Integration der einzelnen Teilsysteme deutlich wird. Darauf aufbauend werden die Entwürfe der einzelnen Teilsysteme bzw. Module mit ihrer Implementierung bzgl. des Handlungsfelds Modellierung von DSMs in Kapitel 5.2, des Handlungsfelds Modellierung von DMMs in Kapitel 5.3, des Handlungsfelds Visualisierung in Kapitel 5.4 und des Handlungsfelds Analyse und Bewertung in Kapitel 5.5 und Kapitel 5.6 erläutert.

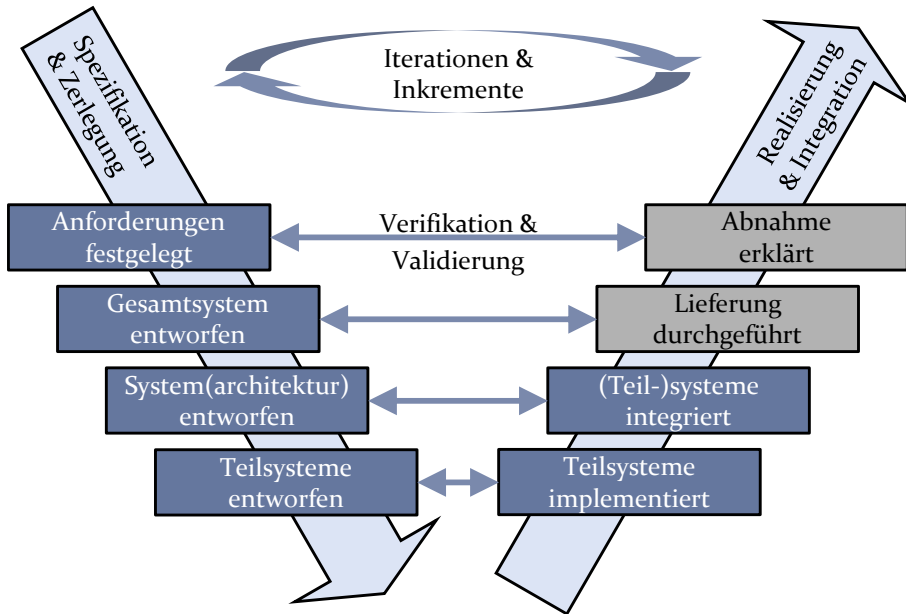


Bild 44: Vereinfachte Darstellung des V-Modell®XT Entwicklungsprozess (in Anlehnung an [16])

5.1 Gesamtkonzept des Assistenzsystems

In diesem Kapitel wird die grundlegende **Methodik mit dem IT-Werkzeug zum holistischen Komplexitätsmanagement** in der integrierten PE überblicksartig erläutert. Dazu wird in Kapitel 5.1.1 zuerst das generelle Vorgehen zum Management komplexer Systeme skizziert, bei dem der Komplexitätsingenieur (vgl. Kapitel 1.4) durch das Assistenzsystem unterstützt wird. Im Anschluss steht der prinzipielle Ablauf der Modellierung in Form der MDM bestehend aus mehreren DSMs und DMMs sowie der darauf aufbauenden Analysephase im Fokus des Kapitels 5.1.2. Abschließend erfolgt in Kapitel 5.1.3 die Vorstellung der IT-Architektur des Assistenzsystems mit den relevanten Modulen und Schnittstellen.

5.1.1 Vorgehen beim Management komplexer Systeme

Da die vorhandenen Vorgehensweisen zum matrixbasierten Management komplexer Systeme entweder zu wenig konkret (z. B. Vorgehen nach EPINGER & BROWNING [161] und nach RINKEVICH & SAMSON [453]) oder zu spezifisch (z. B. nach LINDEMANN ET AL. [355]) sind (s. für Details Anhang A.21), wird auf diesen aufbauend ein neues **agiles, adaptives und analytisches Vorgehen** in Bild 45 vorgestellt.

Dieses Vorgehen zum matrixbasierten Management komplexer Systeme dient zur Erfüllung der in Kapitel 1.3 formulierten Ziele sowie der in Kapitel 4.2.2 aus dem Handlungsbedarf abgeleiteten Anforderungen und stellt somit den wissenschaftlichen Beitrag dieser Arbeit dar. Aus diesem Grund wird in Anlehnung an den DMAIC-Zyklus (engl. für define, measure, analyze, improve and control) aus der Six Sigma Methodik (s. u. a. [386]) das agile, adaptierbare und analytische **Vorgehen für das matrixbasierte Management komplexer Systeme** in die folgenden fünf prinzipiellen Phasen „Definiere“, „Modelliere“, „Analysiere“, „Verbessere“ und „Verifiziere“ gegliedert, wobei dies nur die grundsätzliche Reihenfolge darstellt (s. schwarze Pfeile in Bild 45). Dabei entsprechen die Modellierungsphase der Synthese und die Analysephase der Analyse gemäß dem Problemlösungsprozess der VDI-Richtlinie 2221 (s. Kapitel 3.1.2). Aufgrund der VUCA-Rahmenbedingungen (s. Kapitel 1) muss der Komplexitätsingenieur i. S. eines agilen Komplexitätsmanagements flexibel und oft initiativ, proaktiv und antizipativ agieren und somit zwischen den Phasen hin- und herspringen können (s. dunkelblaue Person Bild 45). Außerdem ist das Modell durch den Komplexitätsingenieur in Folge einer veränderten Zielsetzung anhand gewonnener Erkenntnisse i. S. eines adaptiven Komplexitätsmanagements z. B. auf weitere (oder weniger) notwendige Domänen oder Komplexitätsstufen (s. Kapitel 2.2.2) einfach anzupassen bzw. zu skalieren. Ansonsten ist der Komplexitätsingenieur i. S. eines analytischen Komplexitätsmanagements bspw. im Hinblick auf die Modellierung bei der Navigation durch das Systemmodell bzgl. der drei Dimensionen (vgl. Kapitel 2.3.2) und bei der Bestimmung des zweckmäßigen Detaillierungsgrades von Elementen und deren Relationen (vgl. Ansätze in Kapitel 3.3.1) zu unterstützen.

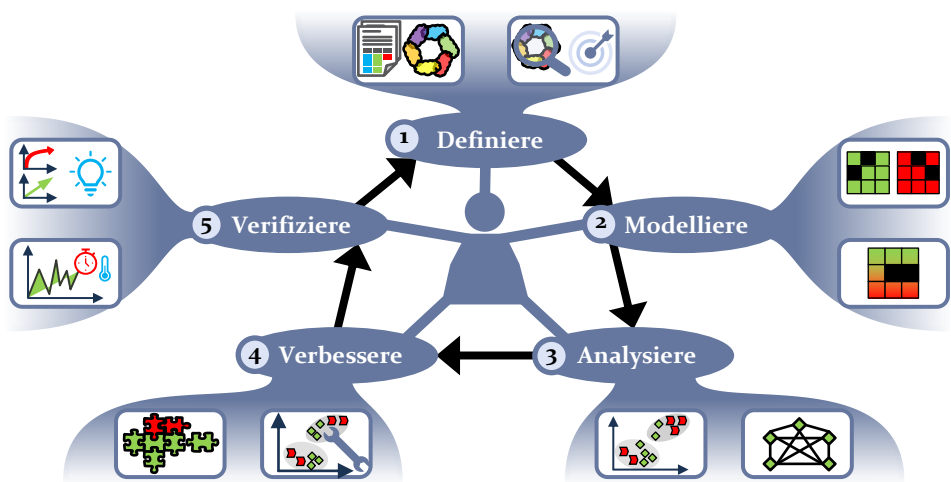


Bild 45: Vorgehen zum matrixbasierten Management komplexer Systeme

Bei dem in der vorliegenden Arbeit angeführtem **agilen, adaptiven und analytischen Vorgehen** wird zwar ein sequenzielles Vorgehen vorgeschlagen (s. Bild 45), aber auch andere Vorgehensweisen werden sowohl methodisch als auch computerunterstützt ermöglicht (im Gegensatz zu existierenden Vorgehensweisen wie in Anhang A.21 skizziert). Die fünf Phasen werden nachfolgend in ihrer generellen Reihenfolge kurz erläutert:

- **Definiere:** Neben der Definition der Aufgaben- und/oder Problemstellung ist die Bestimmung des Zwecks und Ziels im Fokus. Hierbei kann bspw. die Definition des Anforderungsraums (vgl. Kapitel 2.3.2) i. S. v. der anforderungsorientierten Modellierung (s. Anhang A.16) zweckmäßig sein, um alle relevanten Spezifikationen (z. B. Marktanforderungen, Gesetze, Normen, s. unternehmensexterne Komplexität in Kapitel 1.1) für die Zieldefinition zu beachten. Dabei sind oft die durch Target Costing (s. Anhang A.16) abgeleiteten Zielkosten z. B. für bestimmte Produktelemente besonders wichtig.
- **Modelliere:** Aufbauend auf der Definitionsphase sind die relevanten Domänen (s. Kapitel 2.3.1) sowie der notwendige Detaillierungsgrad und ggf. Hierarchisierung (s. Kapitel 2.3.2) zu bestimmen. Danach sind die entsprechenden DSMs mit ihren Elementen zusammen mit deren Relationen sowie die DMMs für die diversen inter-Domänen-Relationen zu modellieren und in eine MDM zu integrieren.
- **Analysiere:** In dieser Phase stehen v. a. die graphbasierten Visualisierungsansätze (s. Kapitel 3.3.2) zum besseren Verständnis und zur Kommunikation sowie die mathematischen Analyseansätze (s. Kapitel 3.3.3) zur multikriteriellen Bewertung zur Verfügung.
- **Verbessere:** In dieser Phase besteht die Möglichkeit zur Manipulation und Modifikation von Elementen und Relationen, um durch diese Änderungen im matrix- und/oder graphbasierten Modell mögliche Verbesserungsmaßnahmen abzuleiten.
- **Verifiziere:** Am Ende sind die in Folge der Realisierung von Verbesserungsmaßnahmen erzielten Ergebnisse und Erkenntnisse zu verifizieren und zu validieren. Dadurch kann sich der Bedarf nach der Definition einer weiteren Aufgaben-/Problemstellung ergeben.

Ein **zweckmäßiges Abweichen von der grundsätzlichen Reihenfolge** ist z. B. denkbar, wenn sich in der Analysephase durch die Kommunikation mit Domänenexperten zeigt, dass eine Verifizierung der modellierten DSM durch (Fach-)Experten sinnvoll ist. Zudem kann sich bspw. auch in der Verbesserungsphase herausstellen, dass bestimmte Elemente und/oder Relationen für eine zweckdienliche Manipulation zuerst detaillierter zu modellieren sind.

5.1.2 Prinzipieller Ablauf der Modellierung und Analyse

Auf Basis der tiefgreifenden Auseinandersetzung mit dem Management von Komplexität in PE wurde vom Autor erkannt, dass die Realisierung der Zielsetzung dieser Arbeit am besten durch eine effektive, effiziente und zielgerichtete Modellierung komplexer Systeme erfolgen kann (vgl. Kapitel 2). Da alle domänenspezifischen Modelle (vgl. Kapitel 3.1) aus Elementen und Relationen bestehen, sind die matrixbasierten Ansätze für deren Modellierung (s. Kapitel 3.2.3) besonders gut geeignet. Da bzgl. der bereits bestehenden Ansätze für das matrixbasierte Komplexitätsmanagement (s. Kapitel 3.3) noch großer Handlungsbedarf besteht (s. Kapitel 4) und die Modellierung matrixbasierter Modelle für die darauf aufbauende graphbasierte Visualisierung (s. Kapitel 3.3.2) sowie mathematische Analyse und Bewertung (s. Kapitel 3.3.3) entscheidend ist, wird zuerst der grundlegende Aufbau und Ablauf der **Methodik zur Modellierung und Analyse komplexer Systeme** vorgestellt. So liegt der Fokus dieser Arbeit auf der Vorstellung der erforschten, wissenschaftlichen Lösung für die Gestaltung der zwei Phasen „Modelliere“ und „Analysiere“.

In Bild 46 ist eine **Konkretisierung der Phase Modelliere** aus Bild 45 dargestellt. Diese Phase besteht aus vier Schritten, welche zwar in Bild 46 in ihrer optimalen Abfolge angeordnet sind aber auch explizit Iterationen ermöglichen. Aufbauend auf der Definitionsphase sind in einem ersten Schritt alle Systemelemente und -relationen aus den domänenspezifischen Modellen (vgl. Kapitel 3.1), die für die verfolgte Zielsetzung relevant sind, zu akquirieren. Dazu sind alle domänenspezifischen IT-Systeme (s. Bild 43), in welchen die erforderlichen Modelle zu finden sind, im Rahmen der Modellierung des im Fokus stehenden, komplexen Systems zu berücksichtigen. Dabei kann es auch sein, dass die noch benötigten aber nicht digital bzw. formal vorliegenden Modelle manuell ergänzt werden müssen. In dem in Bild 46 dargestellten beispielhaften Ablauf sind die drei Domänen Anforderung, Produkt und Prozess von Interesse. Daher erfolgt im zweiten Schritt die Erstellung der DSMs für diese drei Domänen. Dazu werden für diese drei Domänen die Elemente mit deren Relationen in jeweils einer DSM modelliert. In Schritt drei sind alle im Hinblick auf die Zielsetzung notwendigen DMMs zu erstellen. Dafür müssen die bestehenden Relationen zwischen den Elementen von je zwei verschiedenen Domänen modelliert werden, wie z. B. zum einen zwischen den Anforderungs- und Produktelementen und zum anderen zwischen den Produkt- und Prozesselementen. Im abschließenden vierten Schritt sind die drei DSMs und die zwei DMMs in eine MDM zu integrieren. Die vier weiteren, möglichen DMMs (bei drei DSMs) sind in Bild 46 transparent bzw. ausgegraut dargestellt.

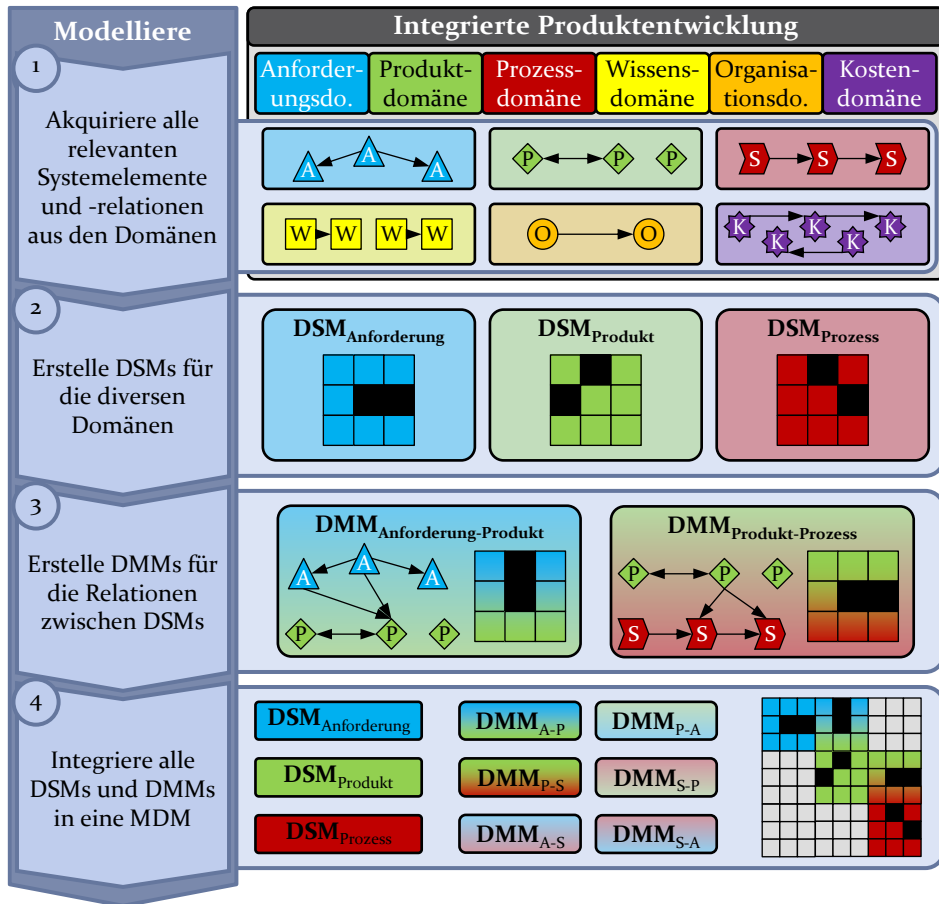


Bild 46: Methodik bzgl. der Modellierung komplexer Systeme

Auf die Modellierungsphase folgt grundsätzlich die Analysephase, die auch im Fokus dieser Arbeit steht. Daher wird ebenfalls die **Konkretisierung der Phase „Analysiere“** in Bild 47 skizziert. In dieser Phase sind kontext-bezogen sowohl Visualisierungsmethoden zweckmäßig als auch Analysemethoden zielgerichtet einzusetzen. Dabei wird insb. in dieser Phase das agile, adaptive und analytische Vorgehen (s. Kapitel 5.1.1) durch eine kontinuierliche Bewertung der Ergebnisse und Erkenntnisse in Folge der Visualisierungs- und Analysemethoden gesteuert. Im Hinblick auf die Visualisierungsmethoden lässt sich die matrixbasierte von den vielfältigen graphbasierten Visualisierungsansätzen differenzieren (s. Bild 47 und Kapitel 3.3.2). Bei den Analysemethoden kann neben den Methoden für Elemente und Relationen zwischen domänenspezifischen und domänenübergreifenden Methoden unterschieden werden (s. Bild 47 und Kapitel 3.3.3).

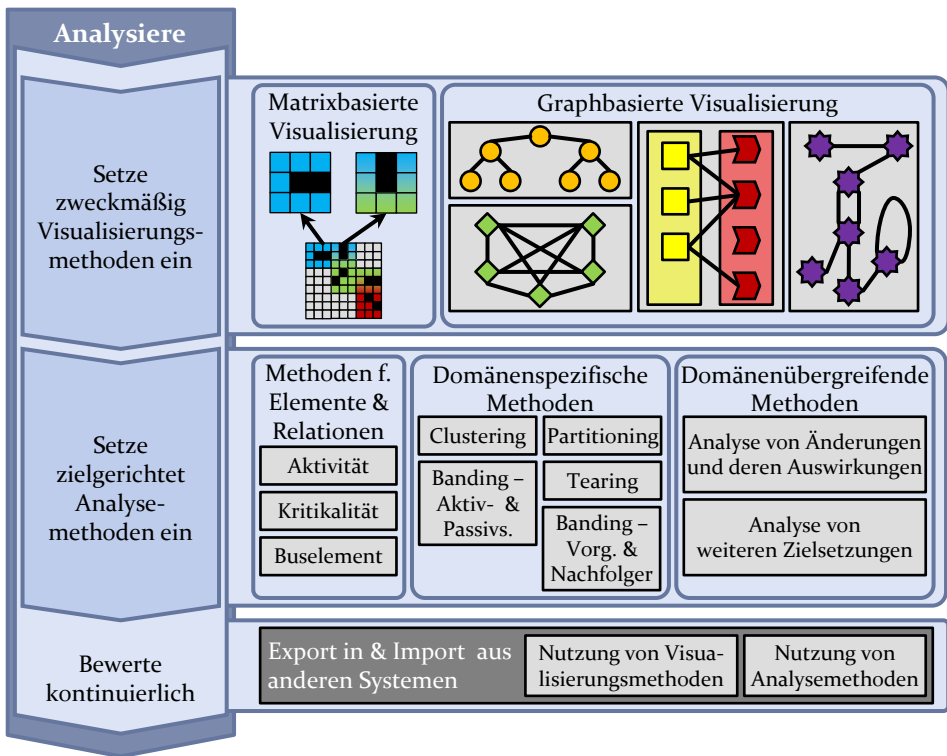


Bild 47: Methodik bzgl. der Analyse, Visualisierung und Bewertung komplexer Systeme

Dieser **prinzipielle Ablauf der Modellierungs- und Analysephase** ist der Kern des Vorgehens für das matrixbasierte Management komplexer Systeme (s. Kapitel 5.1.1) und steht somit im Fokus dieser Arbeit. Dabei ist zu betonen, dass ein wissenschaftlicher Beitrag dieser Arbeit darin besteht, dass das methodische Vorgehen durch den Komplexitätsingenieur flexibel adaptiert werden kann. So kann das Vorgehen z. B. an eine geänderte Zielsetzung, die individuelle Entwicklungssituation und die vielfältigen Nutzerbedürfnisse angepasst werden.

Daher lässt sich schlussfolgern, dass Komplexitätsingenieure durch diese **Methodik beim holistischen Komplexitätsmanagement in der IPE** in vielfältiger Weise unterstützt werden. Dies liegt v. a. daran, dass mittels der Methodik dem Komplexitätsingenieur kontextbezogen jeweils geeignete Methoden für eine effektive und effiziente Modellierung, Analyse, Visualisierung und Bewertung von komplexen Systemen für eine zielgerichtete Entscheidungsfindung bereitstellt werden. Dadurch wird sowohl die Systemanalyse als auch -synthese in der PE mit Hilfe dieser neuen Methodik unterstützt.

5.1.3 IT-Architektur und Module des Assistenzsystems

Nachfolgend werden durch die Erläuterung der generellen **Architektur des Assistenzsystems mit dessen Modulen die Besonderheiten** bzw. charakteristischen Eigenschaften deutlich. Auch durch die im Folgenden verwendeten Bezeichnung des Assistenzsystems AS₄MAVEN² wird v. a. die modulare, flexible, erweiterbare und skalierbare Architektur ausgedrückt. Dabei steht in diesem Akronym zum einen „AS₄“ für **agiles, adaptives und analytisches Assistenzsystem** sowie zum anderen „MAVEN²“ für die englischen Begriffe Modelling, Analysis, Visualisation, Evaluation, Navigation und Negotiation. Zudem ist auch das Wort „Maven“ zur Charakterisierung des Assistenzsystems passend, da mit Maven oft ein Fachexperte für eine bestimmte Domäne bezeichnet wird, der – ähnlich zum Komplexitätsingenieur – relevantes Wissen agil an andere Beteiligte vermittelt. Durch die Ergänzung von „N²“ (d. h. engl. Navigation und Negotiation) wird zudem darauf aufmerksam gemacht, dass das im Rahmen der Forschungsarbeiten des Autors dieser Arbeit entwickelte Assistenzsystem eine Lösung darstellt, die zum einen auch für die Navigation durch Modelle, Analysen und Visualisierungen von komplexen Systemen diverser Domänen dient und zum anderen die Kommunikation und Diskussion der Ergebnisse sowie die Beurteilung und Verhandlung der weiteren Verbesserungsmaßnahmen mit den vom Komplexitätsingenieur zu involvierenden bzw. verantwortlichen Stakeholdern gut unterstützt.

Die **Architektur des Assistenzsystems AS₄MAVEN² mit den zentralen Modulen** ist in Bild 48 vereinfacht und überblicksartig dargestellt. Darin sind ausgewählte Module des Assistenzsystems hellblau hinterlegt. Die grau hinterlegten Elemente gehören zwar nicht zum Assistenzsystem, aber stehen mit diesem in Interaktion. Die vier Hauptmodule sind dunkelblau eingefärbt. Dabei lassen sich die drei Module matrixbasierte Modellierung, graphbasierte Visualisierung und mathematische Analyse über eine graphische Benutzeroberfläche (engl. Graphical User Interface, GUI) bedienen. Diese greifen u. a. mittels Methoden zur Transformation, Integration und Manipulation auf die Wissensbasis zu, welche insb. die Systemelemente und deren Relationen der relevanten Domänen in Form von DSMs sowie Relationen zwischen Elementen verschiedener Domänen in DMMs enthält.

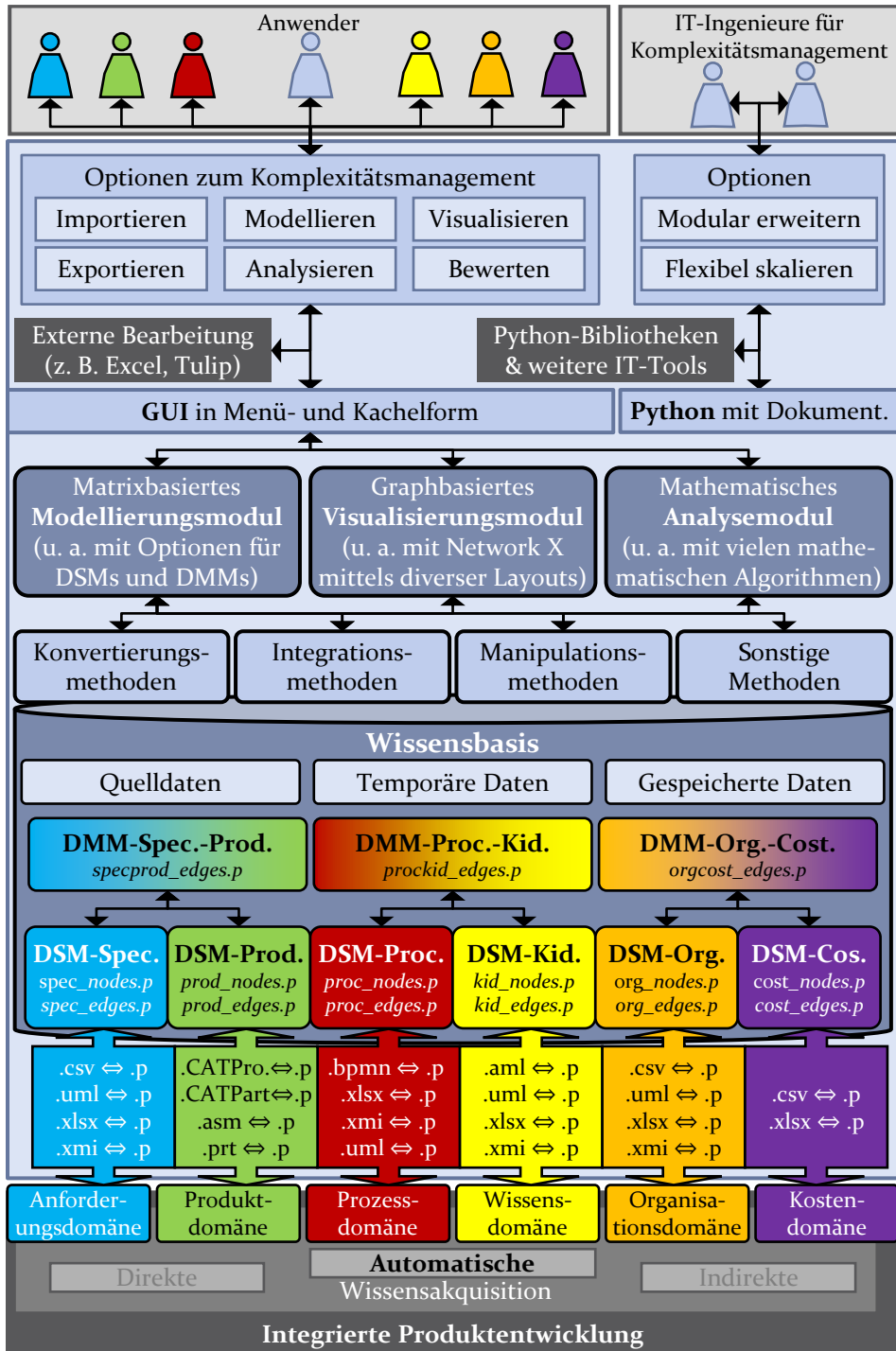


Bild 48: Vereinfachte, schematische Architektur des Assistenzsystems AS4MAVEN²

Die Nutzer des Assistenzsystems sind einerseits diverse Anwender, wie v. a. der Komplexitätsingenieur, und andererseits der sog. IT-Ingenieur für Komplexitätsmanagement bzw. Administrator. Den Anwendern werden mit der **GUI vielfältige Optionen zum Komplexitätsmanagement zur Verfügung** gestellt. In Bild 49 ist eine beispielhafte GUI mit einer Auswahl an möglichen Funktionalitäten in Menü- und Kachelform dargestellt. In Bild 49 besteht die GUI aus einer MDM mit vier DSMs, wobei im dargestellten Beispiel die DSM-Organisation vollständig modelliert wurde (in Grün hinterlegt). Die generell bereit gestellten Funktionalitäten können entweder aus dem Menü aufgerufen oder den Kacheln hinterlegt werden. In Bild 49 sind einige Konvertierungs- und Importfunktionalitäten zu finden, die z. T. in dieser Arbeit erläutert und auch modular erweitert werden.

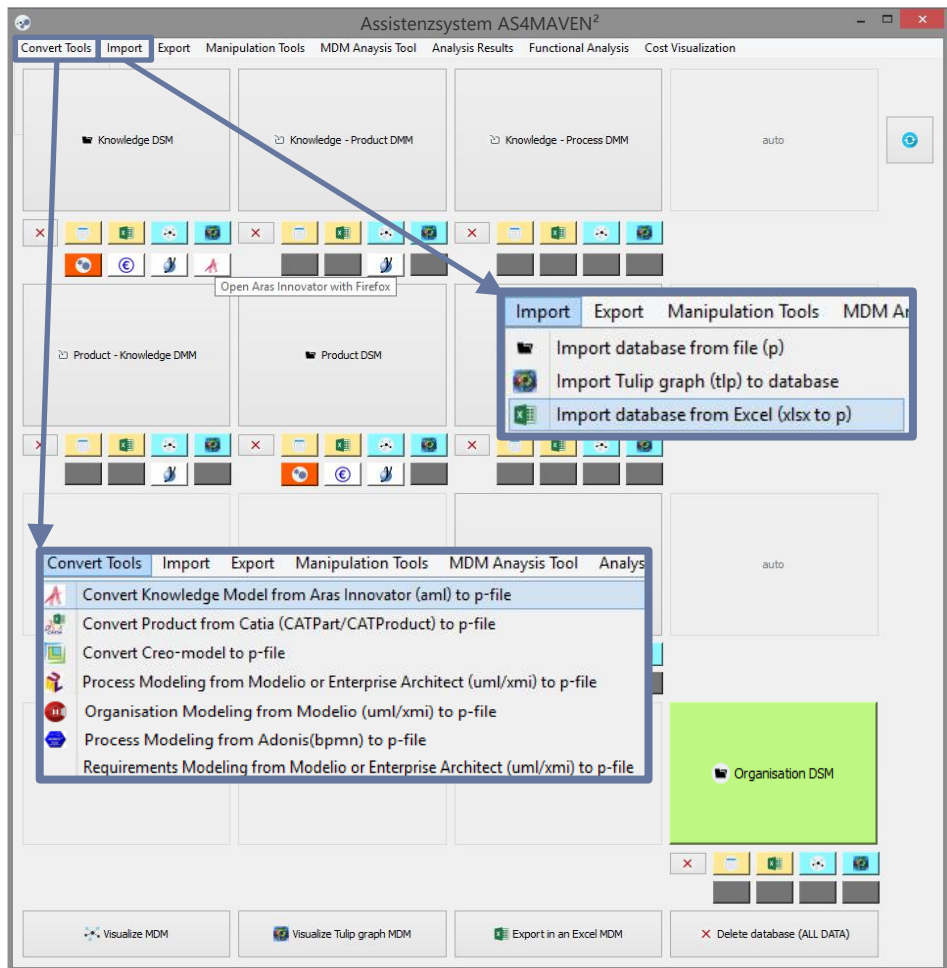


Bild 49: Übersicht der Benutzeroberfläche mit ausgewählten Funktionalitäten

Zudem wird auch eine externe Bearbeitung (d. h. mit Import und Export) ermöglicht, um z. B. diverse Tätigkeiten pragmatisch in Excel zu realisieren oder die graphischen Visualisierungsmöglichkeiten von Tulip nutzen zu können. Auch dem sog. IT-Ingenieur für Komplexitätsmanagement stehen durch die objektorientierte Programmierung mit der verbreiteten Programmiersprache Python, den zahlreichen Python-Bibliotheken sowie der einfachen Anpassung von vorhandenen IT-Schnittstellen zu weiteren IT-Werkzeugen viele **Optionen zur modularen Erweiterung und flexiblen Skalierung des Assistenzsystems** zur Verfügung. Dabei sind v. a. auch die detaillierte Dokumentation bzw. Kommentierung des Codes in englischer Sprache hilfreich. Daher sei auch an dieser Stelle darauf aufmerksam gemacht, dass die verwendeten Begriffe und Abkürzungen in dieser Arbeit z. T. von denen in der Implementierung abweichen (z. B. für die Elemente der Wissensdomäne „W“ vs. „kid“ für knowlegde, information and data).

Die **drei Hauptmodule des Assistenzsystems** sind in Bild 48 dunkelblau koloriert. Davon können die drei Module matrixbasierte Modellierung, graphbasierte Visualisierung und mathematische Analyse mit Hilfe der GUI von den Anwendern genutzt werden. Diese Module sind über Methoden für die Konvertierung, Integration, Manipulation oder direkt mit der Wissensbasis des Assistenzsystems verknüpft. Der zentrale Inhalt der Wissensbasis sind die matrixbasierten Modelle der komplexen Systeme aus den relevanten Domänen. Dazu sind die jeweils erforderlichen Elemente und deren Relationen aus der integrierten PE über Methoden der Wissensakquisition (s. Anhang A.16) in DSMs zu transformieren. Aufgrund der Zielsetzung (s. Kapitel 1.3) und des Handlungsbedarfs (s. Kapitel 4) der Arbeit steht die automatische Akquisition im Fokus, welche durch direkte und indirekte Wissensakquisition problemlos ergänzt werden kann. In Bild 48 sind einige bereits im Assistenzsystem implementierte Schnittstellen zwischen ausgewählten domänenspezifischen IT-Systemen (vgl. Bild 43) in Form von möglichen Im- und Exporten von Dateiformaten dargestellt. Als Beispiel sei auf die zwei Dateiformate „*asm*“ und „*prt*“ der 3D-CAD-Software PTC Creo hingewiesen, welche in die DSM-Produkt (d. h. bestehend aus „*prod_nodes.p*“ und „*prod_edges.p*“) transformiert werden. Neben den für die Zielsetzung benötigten DSMs umfasst die Wissensbasis die erforderlichen DMMs (in Bild 48 sind drei DMMs beispielhaft dargestellt), welche mit Unterstützung des Assistenzsystems zu modellieren sind.

Details zur IT-Architektur des Assistenzsystem sind im Anhang A.26 zu finden. In den nachfolgenden Kapiteln wird die Modellierung von DSMs (Kapitel 5.2), DMMs (Kapitel 5.3) sowie deren Visualisierung (Kapitel 5.4), Analyse (Kapitel 5.5) und Manipulation (Kapitel 5.6) erläutert.

Fazit zum Gesamtkonzept des Assistenzsystems

Das in diesem Kapitel vorgestellte Konzept für ein auf Grundlage von wissenschaftlichen Erkenntnissen entwickelte Assistenzsystem zum Komplexitätsmanagement in der PE bezieht sich unmittelbar auf die Zielsetzung der Arbeit (s. Kapitel 1.3). Dieses entwickelte Konzept und dessen Implementierung in ein Assistenzsystem zum holistischen Komplexitätsmanagement in der PE unterstützen den Komplexitätssingenieur, indem kontextbezogen jeweils geeignete Methoden für die fünf generellen Phasen „Definiere“, „Modelliere“, „Analysiere“, „Verbessere“ und „Verifiziere“ zur Verfügung gestellt werden. Der Fokus dieser Arbeit liegt dabei insb. auf der Modellierung und Analyse. Durch die Visualisierung und Bewertung von komplexen Systemen wird zudem eine zielgerichtete Entscheidungsfindung in der IPE ermöglicht. Im Vergleich zu bereits existierenden Konzepten ist dieses spezifischer, adaptiver und ermöglicht es, das methodische Vorgehen flexibel und iterativ an unterschiedliche Komplexitätsbedingungen anzupassen. Zusätzliche Effizienz- und Effektivitätsgewinne werden durch eine benutzerfreundliche GUI, eine flexibel erweiterbare IT-Architektur sowie zahlreiche spezifische Module und IT-Schnittstellen erzielt.

5.2 Automatisierte Modellierung von DSMs

Für die **Modellierung von DSMs auf Basis der bereits bestehenden, domänenspezifischen Modelle** sind folgende Softwareprogramme, die in Unternehmen breite Anwendung finden, vom Autor auf Basis einer Aufwand-Nutzen-Abschätzung ausgewählt worden. Somit sind vom Autor und in von ihm betreuten studentischen Arbeiten IT-Schnittstellen zwischen folgenden Softwareprogrammen und dem Assistenzsystem zur computerunterstützten Modellierung von DSM auf wissenschaftlicher Grundlage konzeptuell entwickelt und implementiert worden:

- Anforderungsmodellierung: Enterprise Architect, Modelio und IBM Rational Doors
- Produktmodellierung (v. a. Produktmerkmale): Catia und Creo
- Prozessmodellierung: Adonis und Modelio
- Wissensmodellierung: Aras Innovator, (Netzwerk-)Laufwerk
- Organisationsmodellierung: Modelio und MS Excel
- Kostenmodellierung: MS Excel

Mit Blick auf die industrielle Praxis wird eine Kosten- bzw. Aufwand- und Nutzen-Abschätzungen für die Priorisierung von zusätzlichen, noch zu entwickelnden IT-Schnittstellen empfohlen. Dazu sind für **potenzielle IT-Schnittstellen für die möglichst automatisierte Akquisition aller notwendigen Systemelemente mit deren Relationen** aus den unternehmensspezifischen IT-Systemen für die relevanten Domänen (s. Bild 43) deren Nutzen und Aufwände abzuschätzen. Für die Schätzung des Nutzens der computerunterstützten Modellierung sind einerseits der geringe Zeitaufwand und die niedrige Fehleranfälligkeit sowie andererseits gewonnene Ergebnisse und Erkenntnisse aus deren Analyse, Visualisierung und Bewertung zu berücksichtigen (s. Kapitel 4). Die Aufwände bestehen v. a. in der Entwicklung eines Fachkonzepts für eine IT-Schnittstelle und dessen Implementierung (s. Kapitel 5) und in der Erweiterung des Assistenzsystems.

Da in dieser Arbeit nicht für alle unterschiedlichen Systemelemente und deren Relationen, die für die integrierte PE relevant sind, **Konzepte für eine computerunterstützte bzw. (teil-)automatisierte Akquisition** vorgestellt und prototypisch implementiert werden können, wird die computerunterstützte Modellierung von DSMs zum einen von Produktmodellen in Kapitel 5.2.1 und zum anderen von Prozessmodellen in Kapitel 5.2.2 kurz erläutert. Diese Konzepte lassen sich auch für die computerunterstützte Modellierung sowohl für andere Elemente in diesen beiden Domänen als auch für andere Domänen einfach adaptieren, wie bspw. auf die von LAUER identifizierten 57 verschiedenen Dokumententypen (wie z. B. CAD-Modell, Berechnungsbericht, Lastenheft) der IPE [343].

5.2.1 Produktmodellierung

Eine wissenschaftliche Erkenntnis dieser Arbeit ist es, dass **Produktmerkmale, welche in CAD-Modellen zu finden sind**, besonders nützlich und geeignet für die matrixbasierte Produktmodellierung sind. Dies liegt daran, dass zum einen durch die Merkmale die Eigenschaften des Produkts realisiert werden (vgl. CPM-Ansatz in Kapitel 3.1.1) und zum anderen der Dokumenttyp „CAD-Modell“ – im Gegensatz z. B. zum Dokumenttyp „Konzeptheft“ oder „Gestaltstudie“ (s. weitere Beispiele in [343]) – in formal-strukturierter Art und Weise in parametrischen CAD-Modellen vorliegt.

Daher wird eine Methode vorgestellt, mit der **computerunterstützt relevante Elemente und deren Relationen aus dem CAD-Modell für die matrixbasierte Produktmodellierung** genutzt werden können. Diese Methode wird zwar am Beispiel eines CAD-Modells in Creo Parametric dargestellt, lässt sich aber aufgrund des adaptiven Fachkonzepts einfach auf

andere CAD-Systeme anpassen. Dies wird u. a. dadurch erreicht, indem die zwei Submodule „Datenakquisition“ und „Datenverarbeitung“ in MS Excel bzw. das Assistenzsystem gekapselt sind, so dass für andere CAD-Systeme nur das Datenauslesen leicht anzupassen ist. Daher wurde diese Methode bereits für die mit bzw. in Catia erstellten CAD-Modelle adaptiert (vgl. u. a. [S21], [S51]), um die erforderlichen Informationen für die automatisierte Modellierung der DSM-Produkt zu nutzen. Dadurch werden die geforderte Anpassbarkeit und Erweiterbarkeit des Assistenzsystems ermöglicht. Grundlagen zu Creo sind bspw. in [311] und [108] zu finden.

Die für die matrixbasierte Produktmodellierung relevanten Bestandteile von Creo sind insb. die **Parameter und Beziehungen in den CAD-Modellen**. Parametereigenschaften sind u. a. Besitzer(typ), Modell, Name, Typ (i. S. v. Datentyp, wie z. B. integer, double, string, boolean), Wert und Einheit. Mit Beziehungen (i. S. v. Relationen) werden Abhängigkeiten zwischen Parametern (i. S. v. Elementen) modelliert. Nach VAJNA ET AL. lassen sich arithmetische (v. a. Formeln), logische (v. a. Regeln) und geometrische Beziehungsarten unterscheiden [570]. Diese sind jeweils mit einer kurzen Erläuterung und Beispielen in Bild 50 dargestellt.

Die **relevanten Datenobjekte** – Parameter und Beziehungen – sind an verschiedenen Stellen im CAD-Modell zu finden. So werden Parameter in diversen Ebenen des CAD-Modells definiert (z. B. Teil, Konstruktionselement, Fläche, Kante, Material). Dabei lassen sich sowohl die von Konstrukteuren modellierten als auch die von Creo erzeugten Parameter auslesen. Die Definition von Beziehungen findet u. a. auf folgenden CAD-Modell-Ebenen statt: Teil, Konstruktionselement, Schnitt und Muster. Für die Akquisition der relevanten Datenobjekte und der somit notwendigen Navigation durch das CAD-Modell wurden in dieser Arbeit geeignete, sog. smarte Algorithmen für das Assistenzsystem entwickelt und implementiert.

Die **bidirektionale Schnittstelle**, die zur automatisieren Transformation der relevanten Elemente und Relationen aus dem CAD-Modell und in das Assistenzsystem (und umgekehrt) notwendig ist, wird mit Hilfe des Visual Basic Application Programming Interface (VB API) realisiert. Diese Anwendung beruht auf der Component Object Model (COM) Schnittstelle, mit welcher der Zugriff auf (Daten-)Objekte möglich ist, und zwar unabhängig von der Programmiersprache, mit der diese Objekte implementiert sind (s. für Details [242]). Durch die Verwendung des Python-Moduls *win32com.client* wird die COM-Schnittstelle vom Assistenzsystem genutzt und dadurch die bidirektionale Schnittstelle zwischen dem CAD- und Assistenzsystem realisiert (s. für Details [440]).

Arithmetische Beziehungen (explizite Restriktionen)	Als Formeln bezeichnet sind grundlegende mathematische Funktionen, wie <i>Addition</i> , <i>Subtraktion</i> , <i>Wurzel</i> - und <i>Winkelfunktionen</i> .	Rohrbeispiel $d_i = d_a - 2 \times s$ s. Anhang A.3 für Beispiele
Logische Beziehungen (explizite Restriktionen)	Als Regeln bezeichnet, wie z. B. <i>größer/kleiner als</i> , <i>UND</i> , <i>ODER</i> können mit Bedingungsoperationen (<i>if-then-else</i> -Konstrukten) verwendet werden.	Rohrbeispiel <i>if</i> $s < 2$ <i>then</i> Werkstoff = Edelstahl <i>else</i> Werkstoff = Baustahl
Geometrische Beziehungen (implizite Restriktionen)	Beziehungen sind z. B. Angaben zur geometrischen Positionierung von Elementen, wie <i>parallel</i> , <i>vertikal</i> <i>zusammenfallend</i> .	Rohrbeispiel Zylindrizität (Zylinderform) Konzentrizität (Mittelpunkte)

Bild 50: Beziehungsarten in CAD-Modellen (mit Inhalten nach [570])

Das prinzipielle **Vorgehen zur matrixbasierten Produktmodellierung** lässt sich in eine automatisierte Akquisitions- und Verarbeitungsphase gliedern (d. h. Konkretisierung von Phase 1 und 2 in Bild 46). Diese Phasen bestehen aus den im Bild 51 dargestellten Schritten, die z. T. iterativ zu durchlaufen sind (d. h. bis zu einem bestimmten Abbruchkriterium „max“).

Zuerst ist zwischen dem Assistenz- und dem CAD-System über die zuvor beschriebene bidirektionale Schnittstelle eine Verbindung herzustellen. Danach sind in der **Akquisitionsphase** alle relevanten Parameter mit deren jeweiligen Eigenschaften aus dem CAD-Modell zu akquirieren und im temporären Speicher des Assistenzsystems (vgl. in Bild 48 *prod_nodes.p*) zwischenspeichern. Anschließend sind auch die relevanten Beziehungen mit den für die DSM-Modellierung weiteren erforderlichen Daten in analoger Art und Weise zwischenspeichern (vgl. in Bild 48 *prod_edges.p*).

Nach dieser Akquisitionsphase erfolgt die **Verarbeitungsphase** im Assistenzsystem, in der die Modellierung der DSM-Produkt abgeschlossen wird. Hierfür besteht die Möglichkeit, die akquirierten Daten entweder in MS Excel oder direkt im Assistenzsystem in Form einer DSM zu repräsentieren. Dazu sind die akquirierten Parameter und Beziehungen in Matrixform nach bestimmten Kriterien, die nachfolgend genauer erläutert werden, in das matrixbasierte Visualisierungsmodul des Assistenzsystems oder in MS Excel zu schreiben. Abschließend ist DSM für eine bessere Lesbarkeit und Interpretierbarkeit geeignet zu formatieren, wie bspw. durch Farbgebung und Rahmen zur Strukturierung sowie Symbole bzw. Abkürzungen in den Matrixzellen zur Beschreibung der Relationen.

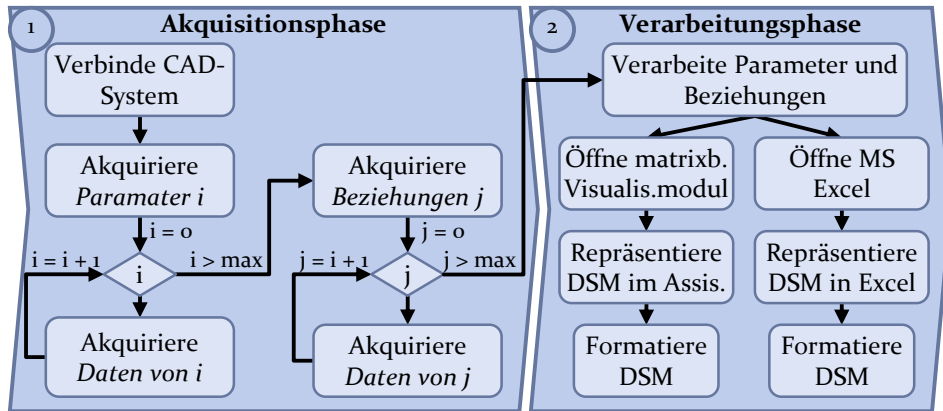


Bild 51: Automatisiertes Vorgehen zur matrixbasierten Produktmodellierung

Akquisitionsphase

Ein CAD-Modell besteht generell aus Bauelementen, welche als Baugruppen (assembly bzw. *.asm*) und Einzelteile (part bzw. *.prt*) bezeichnet werden (s. Bild 52). Die Anordnung der Bauelemente zueinander (z. B. parallel, zentriert, zusammenfallend) erfolgt durch die Definition von Bedingungen bzw. Restriktionen (engl. constraints), wofür oft Geometrielemente (z. B. Koordinationssystem, Ebene, Fläche, Achse) verwendet werden. Durch die implementierte, automatisierte IT-Schnittstelle können **sowohl alle Bauelemente als auch alle Beziehungsarten zwischen diesen akquiriert** werden (s. Bild 52). Dadurch gehen in Folge der Transformation vom CAD- zum DSM-Modell kaum Informationen verloren, so dass zum einen aus der DSM-Produkt das CAD-Modell erstellt werden kann und zum anderen die Nutzer der DSM einen vollständigen Überblick über alle relevanten Bauelemente mit allen deren Relationen, inkl. deren Arten, bekommen.

Auf Ebene der Einzelteile können **sowohl alle Parameter als auch die notwendigen Beziehungsarten durch die IT-Schnittstelle akquiriert** werden (s. Bild 52). Somit lassen sich neben den vom CAD-Konstrukteur manuell modellierten Parametern auch die vom CAD-System automatisch erstellten Parameter automatisiert akquirieren. Neben den Merkmals-Parametern werden auch für die einzelnen Bauelemente die zwei im CAD-Modell zu findenden Eigenschafts-Parameter, die Masse und das Volumen, ausgelesen. Im Hinblick auf die in Bild 50 dargestellten Beziehungsarten lassen sich mit Hilfe der IT-Schnittstelle nur explizite Restriktionen, also arithmetische und logische Beziehungen, in das Assistenzsystem transformieren. Dies liegt v. a. daran, dass geometrische Beziehungen insb. im Zusammenhang zu anderen Maßangaben zu betrachten sind und daher für die matrixbasierte Modellierung wenig bzw. nicht geeignet sind. Zur

Identifikation von arithmetischen Beziehungen wird die linke und rechte Seite der jeweiligen Gleichung auf Übereinstimmung mit den zuvor gefundenen Parametern geprüft. Zur Identifikation von logischen Beziehungen werden die auszuführenden Zeilen wie Formeln behandelt und dadurch kann eine bedingte Beziehung zwischen Ausführung und Bedingung erkannt werden. Neben den beiden expliziten Restriktionen werden für die Modellierung der DSM-Produkt auch indirekte Beziehungen zwischen Bauelementen identifiziert und akquiriert, die vom Anwender bei Bedarf jederzeit auch ausgeblendet werden können.

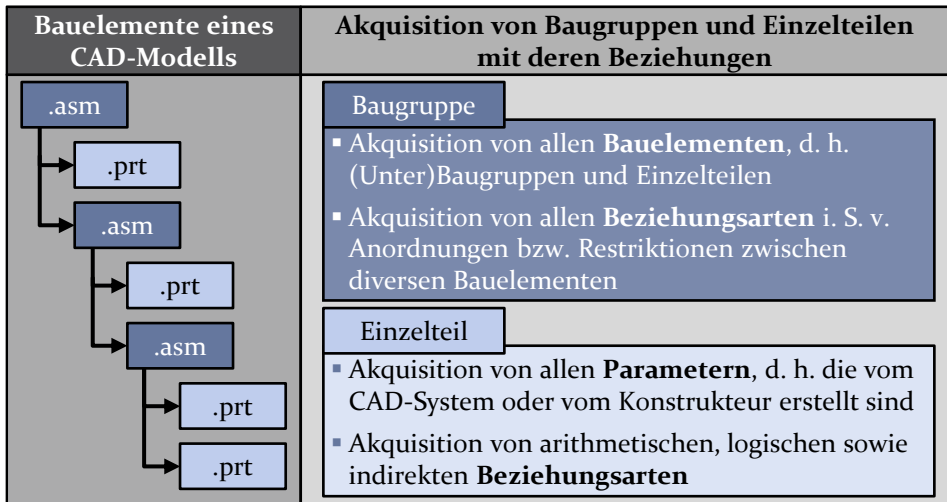


Bild 52: Akquisition von Baugruppen, Einzelteilen und deren Beziehungen

Verarbeitungsphase

Analog zum CPM-Ansatz nach WEBER (s. Kapitel 3.1.1) werden in dieser Methode Bauelemente als Merkmale aufgefasst, da vom Konstrukteur auch z. B. Einzelteile zu Baugruppen oder Unterbaugruppen zu Baugruppen mit Hilfe bestimmter Beziehungen zusammengesetzt werden können. So ist der in Bild 52 links dargestellte Strukturbaum in Bild 53 links als matrixbasiertes Produktmodell repräsentiert. Diese Repräsentation der Baugruppen ist das Ergebnis eines Herauszoomens aus der „darunter liegenden“ Repräsentation auf Einzelteilebene (s. Bild 53 rechts) – ähnlich dem HDDSM-Ansatz in [554]. Im Gegensatz zum HDDSM-Ansatz werden bei dieser Methode alle im CAD-Modell bestehenden Beziehungen zwischen den jeweiligen Bauelementen einer Baugruppe (z. B. *prt3* und *prt4* von *asm3*) in der sog. *DSM_{Constraints}* **automatisiert in der jeweiligen Matrixzelle dargestellt**. Daher enthält die *DSM_{Constraints}* nur die Beziehungen zwischen den beiden Einzelteilen (d. h. unabhängig vom Geometrieelement zur

Definition der Beziehung). Da sowohl die Art als auch die Anzahl der bestehenden Beziehungen zwischen Baugruppen für die DSM-Modellierung relevant sind, werden auf Baugruppenebene – im Gegensatz zur Einzelteilebene – alle aus dem CAD-Modell akquirierten Beziehungsarten (d. h. auch Dopplungen) in der entsprechenden Zelle der $DSM_{Constraints}$ dargestellt. Dagegen ist es aus Anwendersicht auf Einzelteilebene sinnvoll, Beziehungen identischer Art nur einmal in die Matrixzelle einzutragen. Damit kann es zwischen zwei Elementen eines Einzelteils maximal vier Beziehungen geben (d. h. eine direkte, eine bedingte und die jeweils indirekte Form). Auf die $DSM_{Constraints}$ folgen die DSMs der Einzelteile entlang der Diagonalen. Für die bessere Lesbarkeit und Übersichtlichkeit werden die einzelnen DSMs danach formatiert (z. B. mittels Rahmen, Farben). So wird sofort ersichtlich, dass die Baugruppe *asm2* das Einzelteil *prt2* sowie die Unterbaugruppe *asm3*, welche *prt3* und *prt4* enthält, umfasst (s. Bild 53 links).

	asm1	prt1	asm2	prt2	asm3	prt3	prt4
asm1							
prt1							
asm2							
prt2							
asm3							
prt3							
prt4							

		DSMCon		prt3			prt4	
		prt3	prt4	M1	M2	E1	M3	E2
DSMCon	prt3							
	prt4							
prt3	M1							
	M2							
	E1							
prt4	M3							
	E2							

Bild 53: Repräsentation von Baugruppen (links) und Einzelteilen (rechts) in der DSM

Indirekte Beziehungen sind zwar nicht direkt modelliert, können aber das Resultat einer Kombination aus mind. zwei Beziehungen sein. So stellen die zwei Formeln zur Berechnung der Masse ($m = V * \rho$) und des Volumens ($V = l * A$) des Rohrs ein anschauliches Beispiel dar (s. Anhang A.4 für Details). Da sich durch die Modifikation des Parameters Länge l (oder der Querschnittsfläche A) auch der Wert des Volumens V ändert, welcher wiederum die Masse m beeinflusst, liegt eine indirekte Beziehung zwischen der Länge l (und auch A) sowie der Masse m vor. Nach dieser Logik werden auf Basis der im temporären Speicher akquirierten Relationen (s. Akquisitionsphase) mit Hilfe der implementierten Methode nicht nur indirekte Beziehungen 1. Grades, sondern auch indirekte Beziehungen (beliebig) höherer Grade ermittelt und in der DSM dargestellt.

Datenstruktur der Parameter und Beziehungen auf Einzelteilebene

Alle Parameter des CAD-Modells werden jeweils zusammen mit allen für die DSM-Modellierung relevanten Informationen mittels eines *key* (dt. Schlüssel) in einem *dictionary* in der Wissensbasis des Assistenzsystems gespeichert. Somit umfasst jedes **dictionary für einen Parameter** generell die folgenden *keys* mit den dazugehörigen Informationen:

- *name*: Mit *name* wird der Parametername erfasst, der entweder eine automatisch durch das CAD-System vergebene Bezeichnung (z. B. „d2“) oder eine durch CAD-Konstrukteure vergebene Benennung (z. B. „Außendurchmessers des Rohrs A“) haben kann
- *type*: Mit *type* (sofern vorhanden) wird die Art des Parameters (z. B. „linear“, „radial“, „diameter“ und „angular“) erfasst, der nur für Bemessungen und u. a. für das Auslesen der Einheit erforderlich ist
- *value*: Mit *value* wird der Wert des Parameters (z. B. „50.0 mm“, „true“ und „v_oi“) erfasst, wobei die möglichen Datentypen *string*, *integer*, *boolean* und *double* sind
- *unit*: Die *unit* (dt. Einheit) wird in Creo global festgelegt, wobei der voreingestellte Standard „Millimeter Newton Second (mmNs)“ ist; Auf Basis des Standardeinheitensystems und der zuvor ausgelesenen *types* werden mit Hilfe einer Funktion des Assistenzsystems die entsprechenden Einheiten für jeden Parameter erfasst
- *ID*: Die *IDs* dienen zur eindeutigen Identifikation von Baugruppen, Einzelteilen und Parametern, um diesen deren Parametern und/oder Beziehungen richtig zuordnen zu können; da eine KE-ID (z. B. KE-ID „23“, vgl. Bild 54) auf verschiedenen Ebenen des CAD-Modells i. d. R. mehrmals vorkommen kann und somit durch eine KE-ID allein keine eindeutige Identifikation möglich ist, wird eine Kombination von KE-IDs erforderlich; daher werden mit Hilfe einer Funktion des Assistenzsystems die durch Creo automatisch vergebenen KE-IDs entsprechend ihres Pfads durch die Baumstruktur des CAD-Modells top-down kombiniert, so dass alle Baugruppen, Einzelteile und Parameter eindeutig identifiziert werden können (z. B. „I-127-I-23“, vgl. Bild 54)
- *path*: Der *path* (dt. Pfad) dient dazu, dass Anwender Baugruppen und Einzelteile im CAD-Modell schnell finden können; der Pfad wird auf Basis der *ID* (d. h. der Kombination von KE-IDs) erstellt (z. B. „getriebe.asm/antriebswelle.asm/rillenkugellager.asm/innenring“, vgl. Bild 54)
- *elementtype*: Mit *elementtype* wird die Art des Parameters erfasst, wobei die entsprechende Funktion des Assistenzsystems folgende

vier Parameterarten unterscheidet: „c“ (für characteristic, Merkmal), „p“ (für property, Eigenschaft), „prt“ (für part, Einzelteil) und „asm“ (für assembly, Baugruppe)

- *index*: Mit dem *index* wird jedem Dictionary ein eindeutiger integer zugewiesen, so dass zielgerichtetes Zuordnen von Beziehungen und zielgerichtetes Zugreifen auf bestimmte Dictionaries möglich ist

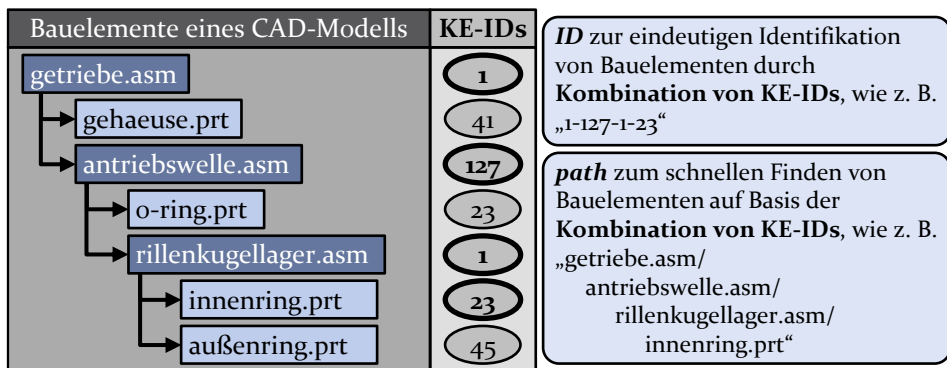


Bild 54: Schematische Darstellung eines CAD-Modell-Strukturbaums mit KE-IDs

Die **Zusammenfassung der dictionaries für ein Einzelteil erfolgt in je einer list** (dt. Liste). Die Liste eines Einzelteils bzw. die Listen von mehreren Einzelteilen, die zu einer Baugruppe gehören, werden in einer weiteren Liste dieser Baugruppe gespeichert. Dadurch können vielfach ineinander verschachtelte Baugruppen eindeutig in der Wissensbasis des Assistenzsystems erfasst werden. So ist bspw. in Bild 55 z. B. die Struktur einer Baugruppe, die aus zwei Einzelteilen besteht, dargestellt.

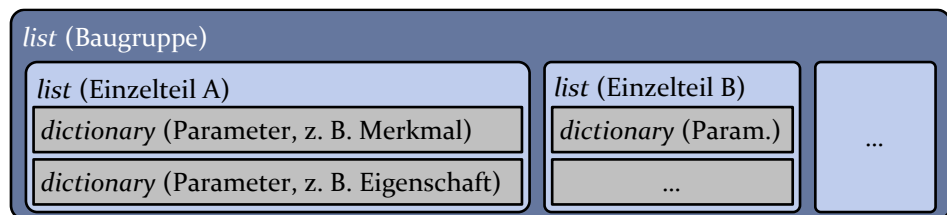


Bild 55: Datenstruktur auf Einzelteilebene in der Wissensbasis des Assistenzsystems

Die Beziehungen zwischen Parametern werden – analog zur Speicherstruktur der Parameter – in *dictionaries* in die Wissensbasis des Assistenzsystems gespeichert sowie auch mittels *lists* strukturiert. Jedes **dictionary für eine Beziehung** hat generell folgende *keys* und Informationen:

- *row*: Mit *row* (dt. Zeile) wird jeweils die Zeilennummer (d. h. eindeutige Indizes) der beeinflussenden Parameter erfasst

- *column*: Mit *column* (dt. Spalte) wird jeweils die Spaltennummer (d. h. eindeutige Indizes) der beeinflussenden Parameter erfasst
- *elementtype*: Mit *elementtype* wird zwischen Bauelementen auf Baugruppen- und Einzelteilebene differenziert, so dass die beiden *values* (dt. Werte) „*assembly*“ (für Baugruppe) und „*part*“ (für Einzelteil) sind; davon hängt bei der Repräsentation der DSM ab, ob identische Beziehungstypen zwischen Bauelementen auf Baugruppenebene mehrmals oder zwischen Einzelteilen auf Einzelteilebene nur einmal dargestellt werden
- *type*: Mit *type* wird die Art der Beziehung erfasst, wie z. B. ein „X“ für eine direkte und ein „Y“ für eine bedingte Beziehung sowie eine Abkürzung der Beziehungsart bei (Lage-)Bedingungen und ein Einklammern des Eintrags bei allen indirekten Beziehungen

Datenstruktur der Bauelemente auf Baugruppenebene

Im Hinblick auf die Baugruppenebene müssen nicht nur die Parameter und Beziehungen eines jeden Einzelteils – wie zuvor bereits beschrieben – in die Wissensbasis des Assistenzsystem gespeichert werden, sondern auch zusätzlich jeweils die Beziehungen zwischen den einzelnen Bauelementen. Analog zur Datenstruktur der Parameter und Beziehungen auf Einzelteilebene werden auch **alle Bauelemente auf Baugruppenebene erfasst** und jeweils zusammen mit allen für die DSM-Modellierung relevanten Informationen mittels derselben *keys* wie auch bei den Parametern in einem *dictionary* in der Wissensbasis des Assistenzsystem gespeichert.

Analog zur Einzelteilebene werden auch die Struktur der Bauelemente und Beziehungen mit Hilfe von *lists* strukturiert. Dabei wird aber zusätzlich berücksichtigt, dass z. B. beim Hinzufügen einer weiteren Unterbaugruppe **Beziehungen zwischen allen bereits bestehenden Bauelementen zu erfassen sind**. Daher ist die Listenstruktur aus Bild 55 für die Daten- und Speicherstruktur auf Einzelteilebene für die Daten- und Speicherstruktur auf Baugruppenebene entsprechend erweitert (s. Bild 56). Dazu wird zusätzlich zur Listenstruktur aus Bild 55 auf jeder Ebene ein *dictionary* eingegliedert, welches die entsprechenden Informationen auf der jeweiligen Baugruppenebene enthält. So sind bspw. alle Bauelemente der Unterbaugruppe in Bild 56 in der *list* (Unterbaugruppe) zusammengefasst (z. B. die Einzelteile A und B), welche auf das *dictionary* (Unterbaugruppe) folgt.

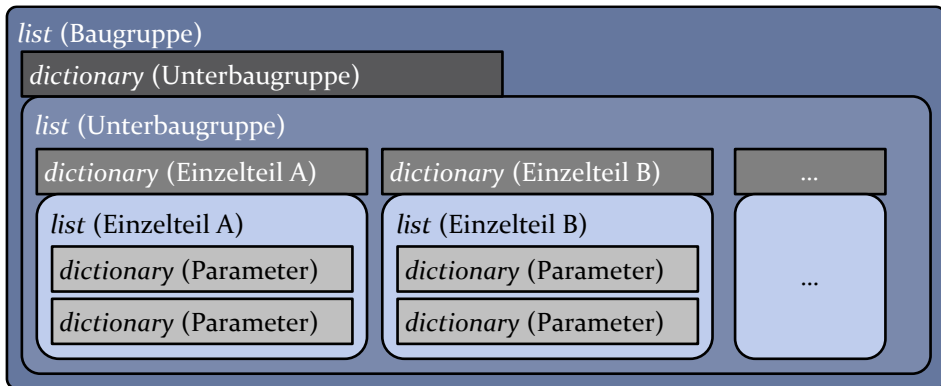


Bild 56: Datenstruktur auf Baugruppenebene in der Wissensbasis des Assistenzsystems

5.2.2 Prozessmodellierung

Eine wissenschaftliche Erkenntnis dieser Arbeit in Bezug auf die Prozessdomäne ist, dass die Elemente in Prozessmodellen insb. durch Phasen und Prozessschritte als auch durch Ereignisse, Zustände und Prozessressourcen repräsentiert werden. Diese Prozesselemente können durch Angabe von z. B. (Durchlauf-)Zeit und Art (attributive Daten) genauer spezifiziert werden. Die **Relationen dienen der Verknüpfung der Prozesselemente** und können durch Angabe von z. B. Richtung und Art der Relation konkretisiert werden. Dementsprechend sind die Konnektoren, d. h. logische Operatoren zur Verknüpfung von Relationen (i. S. v. Operationen der Boole'schen Algebra), in Prozessmodellen entscheidend. Dabei lassen sich aufgrund der vielen prozessorientierten Modellierungsansätze (s. Kapitel 3.1.2) auch für die Erstellung von Prozessmodellen in der industriellen Praxis eine Vielzahl von Modellierungssprachen und Modellierungssoftware finden (s. Anhang A.23).

Daher kann die Methode für die **computerunterstützte Modellierung der matrixbasierten Prozessmodellierung auf Basis der relevanten Elemente und Relationen aus den Prozessmodellen** auf alle üblichen Modellierungssprachen sowie Modellierungssoftware adaptiv angepasst werden. Dies wird – analog dem Vorgehen zur Produktmodellierung in Kapitel 5.2.1 – realisiert. Nachfolgend wird die Methode am Beispiel der Modellierungssprache Business Process Model and Notation (BPMN) und der Modellierungssoftware ADONIS vorgestellt, da diese stark verbreitet sind (s. Anhang A.23). Die BPMN ist – ähnlich wie die UML und die EPK – eine weit verbreitete, international standardisierte Modellierungssprache. Die Software ADONIS dient mit jeweils einer kommerziellen und kostenlosen

Version u. a. zur Modellierung, Analyse und Dokumentation von Unternehmensprozessen. Diese Methode ist neben ADONIS z. B. auch für die in Modelio erstellte Prozessmodelle adaptiert (vgl. [S4], [S20]), um die notwendigen Informationen für die automatisierte Modellierung der DSM-Prozess zu nutzen. Dadurch ist sichergestellt, dass das Assistenzsystem im Hinblick auf Modellierungssprachen und dazugehörige Softwaresysteme der Prozessdomäne einfach angepasst werden kann.

Eine wissenschaftliche Erkenntnis der Forschungsarbeiten des Autors ist daher, dass die für die matrixbasierte Prozessmodellierung relevanten Bestandteile bspw. des BPMN-Prozessmodelles in ADONIS v. a. die **Prozesselemente und deren Relationen in den Prozessmodellen** sind (vgl. Anhang A.23). Somit sind aus BPMN-Prozessmodellen insb. die Flussobjekte (d. h. Aktivitäten, Gateways, Ereignisse) als die relevanten Prozesselemente und die verbindenden Objekte (d. h. Sequenzfluss, Nachrichtenfluss, Assoziation) als die relevanten Prozessrelationen automatisiert zu akquirieren und zu verarbeiten (s. Anhang A.23). Dabei ist die automatisierte Akquisitions- und Verarbeitungsphase zur matrixbasierten Prozessmodellierung dem in Bild 51 skizzierten Vorgehen zur Produktmodellierung sehr ähnlich. So dienen analog z. B. *.prt*- und *.asm*-Dateien zur automatisierten Produktmodellierung bspw. die mit ADONIS modellierten *.bpmn*-Dateien als Ausgangsbasis für die Akquisition zur automatisierten Modellierung der DSM-Prozess. Dabei sind die *.bpmn*-Dateien durch die Extensible Markup Language (XML), einer sowohl von Menschen als auch von Rechnern lesbaren Auszeichnungssprache zur Modellierung hierarchisch strukturierter Daten in Form einer Textdatei, encodiert. Daher wird nachfolgend die Methodik und deren Implementierung zur automatisierten Modellierung der DSM-Prozess nur verkürzt erläutert. Dabei werden – im Vergleich zu den im Stand der Forschung und Technik dargestellten Ansätzen – die Besonderheiten sowie die Weiterentwicklungen erläutert, wie bspw. die automatisierte Modellierung logischer und hierarchischer Relationen.

Automatisierte Modellierung logischer Prozessrelationen

Analog zu den Produktelementen werden auch die Prozesselemente mit Hilfe einer bidirektionalen Schnittstelle zwischen dem Assistenzsystem und ADONIS automatisiert akquiriert und in die Wissensbasis des Assistenzsystems gespeichert (vgl. *proc_nodes.p* in Bild 48). Dabei enthält generell jedes **dictionary für ein Flussobjekt** aus dem BPMN-Modell die folgenden *keys* mit jeweils dazugehörigen Informationen: *ID* (Identifikationsnummer), *index* (i. S. v. laufende Nummer), *name* (Bezeichnung), *Level1* (Hierarchieebenen 1), *Level2* (Hierarchieebenen 2), *value* (Wert, z. B.

Bearbeitungszeit), *unit* (Einheit, z. B. Bearbeitungszeit in Stunden) und *type* (Art des Flussobjekts, s. Anhang A.23). Die generellen *types* des Flussobjekts sind die Aktivität (d. h. Prozessschritt i. e. S.), die drei Ereignisse Start-, Zwischen- und Endereignis sowie die drei Gateways, also paralleles (d. h. *AND*), inklusives (d. h. *OR*) und exklusives (d. h. *XOR*) Gateway.

In ähnlicher Weise wie die Beziehungen für die Produktmodellierung (vgl. Kapitel 5.2.1) werden auch die verbindenden Objekte (d. h. Sequenzfluss, Nachrichtenfluss, Assoziation) automatisiert akquiriert und in *dictionaries* der Wissensbasis des Assistenzsystems gespeichert sowie auch mittels *lists* strukturiert (vgl. *proc_edges.p* in Bild 48). Dabei umfasst jedes ***dictionary für ein verbindendes Objekt*** generell folgende *keys* und Informationen: *ID* (Identifikationsnummer), *row* (Quellprozessschritt), *column* (Zielprozessschritt), *type* (Konnektor-Symbol bzw. Verknüpfungszeichen).

In den bisher existierenden Ansätzen werden die Relationen in der DSM-Prozess i. d. R. lediglich durch ein „X“ oder Ziffern symbolisiert, so dass die drei logischen Relationen *AND*, *OR* und *XOR* nicht explizit modelliert oder zumindest nicht im Rahmen einer automatisierten Prozessmodellierung berücksichtigt werden. Allerdings sind die drei **logischen Relationen und infolgedessen die types der verbindenden Objekte der BPMN für die Prozessmodelle essenziell**, weil diese den logischen Ablauf des Prozesses beschreiben. Daher ist im Assistenzsystem eine neu entwickelte Methodik zur Akquisition und Verarbeitung von logischen Relationen bzw. Operatoren implementiert, welche die ursprünglich mittels der BPMN modellierten, verbindenden Objekte und Gateways eindeutig und anschaulich in das matrixbasierte Produktmodell transformiert. Dafür stellt diese Methodik eine Symbol- als auch eine Textdarstellung bereit.

Bei der **Symboldarstellung** sind die verschiedenen *types* der verbindenden Objekte zwischen den involvierten Flussobjekten, d. h. zwischen den beeinflussenden Aktivitäten und beeinflussten Gateways (und umgekehrt), in der jeweiligen Matrixzelle der DSM-Prozess mit verschiedenen Symbolen dargestellt. So ist für die Symbolisierung des parallelen Gateways „*AND*“ ein Pluszeichen „+“ (für i. S. v. „und zugleich“) besonders gut geeignet. Mit dem parallelen Gateway wird ein paralleler, oft gleichzeitiger Prozessablauf von mind. zwei Prozessschritten (vgl. Simultaneous Engineering) modelliert, wobei alle mit „+“ gekennzeichneten Prozesse auszuführen sind (s. Bild 57). Das inklusive Gateway „*OR*“ wird mit einem „O“ (i. S. v. „oder/und zudem auch“) symbolisiert. Mit dieser logischen Relation werden viele Optionen für den Prozessablauf ermöglicht, von denen mind. ein oder beliebig viele Prozessschritt(e) abzuschließen sind. Im Gegensatz dazu kann beim

exklusiven Gateway „XOR“ nur ein einziger der folgenden Prozessschritte durchlaufen werden. Grund hierfür ist, dass mit XOR gewöhnlich eine Bedingung verknüpft ist, die nur eine der folgenden Prozessschritte erfüllt und daher nur dieser ausgeführt wird. Kennzeichnend dafür wird ein Häkchen „✓“ (i. S. v. Bedingung erfüllt) vor dem Prozessschritt gesetzt. Dagegen wird vor dem bzw. den Prozessschritt(en), welche die Bedingung nicht erfüllen und somit nicht ausgeführt werden, ein „X“ (i. S. v. Bedingung nicht erfüllt) modelliert. Dabei kann über die Einstellungen im Assistenzsystem auch ein anderes Symbol gewählt werden, um Verwechslungen mit anderen Relationen, welche durch ein „x“ (also mit dem Kleinbuchstaben) symbolisiert sind, zu vermeiden.

Anstelle der zuvor skizzierten Symbole werden bei der **Textdarstellung** Texte bzw. Begriffe zur Modellierung der logischen Relationen verwendet. So wird beim parallelen Gateway „AND“ der Text „UND“, beim inklusiven Gateway „OR“ das Wort „ODER“ und beim exklusiven Gateway „XOR“ der Begriff „ENTWEDER“ in den entsprechenden Matrixzellen angezeigt.

Automatisierte Modellierung hierarchischer Prozessrelationen

Analog zu Produkthierarchien der Produktdomäne bestehen auch in der Prozessdomäne hierarchische Prozessmodelle, deren Erstellung mit Hilfe vieler prozessorientierter Modellierungsansätze unterstützt werden (vgl. z. B. FORFLOW-Prozessmodell in Kapitel 3.1.2). Die **Methode für die automatisierte Modellierung hierarchischer Prozessrelationen** beruht z. T. auf der Systemhierarchiemodellierung des HDDSM-Ansatzes in Bild 30. So wird analog zu diesem Absatz die *Aktivität 2* (kurz „Aktiv. 2“) in Bild 57 durch Herunterbrechen auf eine niedrigere Hierarchieebene in die zwei Teilaktivitäten *Aktivität 2.1* und *Aktivität 2.2* sowie das *Start-* und *Endereignis* des Unterprozesses (d. h. der *Aktivität 2*) detailliert.

Die Methode orientiert sich am **Konzept der Prozesshierarchieebenen**, also der Gruppierung in sich möglichst gekapselter Prozesselemente, wie z. B. *Aktivität 2.1* und *Aktivität 2.2* in Bild 57. So bildet die höchste Ebene des BPMN-Prozessmodells die grundsätzliche Struktur der DSM-Prozess. Falls der Prozess auf dieser Ebene Unterprozesse besitzt (z. B. *Aktivität 2*), werden diese im BPMN-Prozessmodell durch ein Plussymbol „+“ (s. Unterprozess in Bild 152 in Anhang A.23) dargestellt. Das Ergebnis der Methode für die automatisierte Modellierung hierarchischer Prozessrelationen, die im Assistenzsystem implementiert ist, sind die beiden modellierten DSMs im nachfolgenden Bild 57. Die den beiden DSMs zugrunde liegenden zwei BPMN-Prozessmodelle sind in Bild 57 oben schematisch dargestellt.

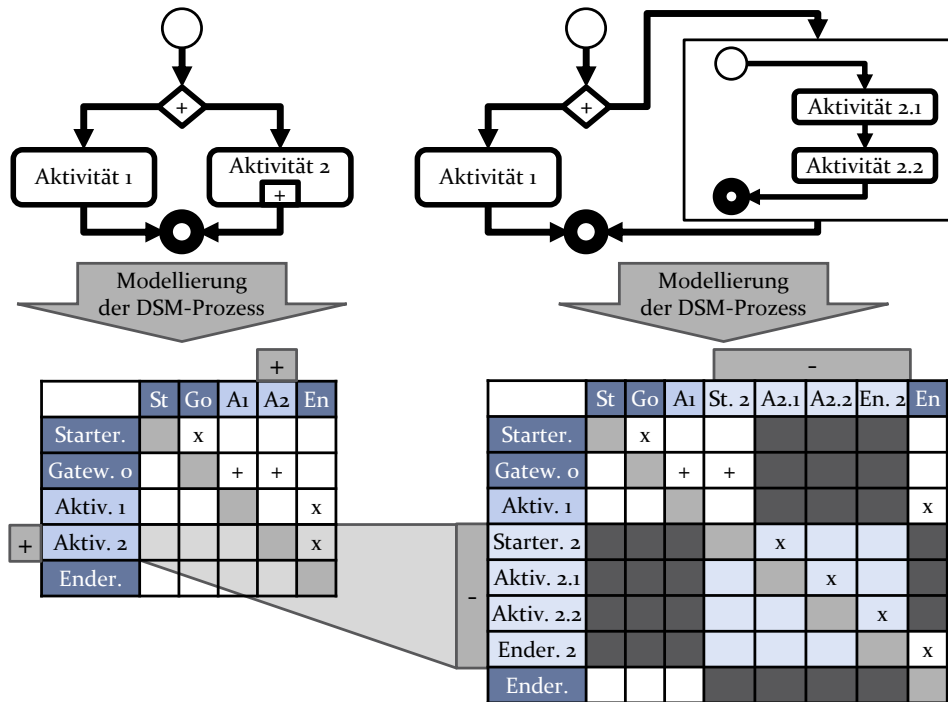


Bild 57: Veranschaulichung der Modellierung hierarchischer Prozessrelationen

Generell werden in der **DSM des Unterprozesses** nur die Relationen innerhalb dieser Ebene angegeben, wobei das Start- und Endereignis des Unterprozesses die beiden Relationen zur übergeordneten Prozessebene sind. Diese Logik findet sich auch in den graphischen Modellierungsansichten der BPMN-Softwarelösungen wieder. Gemäß dieser Logik können die in Bild 57 in dunkelgrauer Farbe eingefärbten Matrixzellen keine Relationen enthalten. Analog zur Repräsentation von Baugruppen (s. Bild 53) lassen sich auch Unterprozesse farblich voneinander abgrenzen. So ist z. B. in Bild 57 der Unterprozess *Aktivität 2* mit hellblauer Farbe hinterlegt.

Fazit zur automatisierten Modellierung von DSMs

In diesem Kapitel wurden auf Basis wissenschaftlicher Erkenntnisse neuartige Konzepte zur computerunterstützten Modellierung von DSMs auf Grundlage der bereits bestehenden, domänenspezifischen Modelle beschrieben. Inwiefern diese Konzepte die automatisierte Akquisition und Verarbeitung der erforderlichen Systemelemente erlauben, wurde detailliert am Beispiel von Produkt- und Prozessmodellen erläutert. Die hier vorgestellten Konzepte erfüllen die in Kapitel 4.2 angeführten Anforderungen aus Sicht der Wissenschaft und Wirtschaft. So sind sie beispielsweise zeitlich effizient einsetzbar, zeichnen sich durch eine geringe Fehleranfälligkeit aus und berücksichtigen die Ergebnisse aus vorangegangenen Phasen der Analyse, Visualisierung und Bewertung. Zudem eignen sich diese Konzepte auch für alle weiteren, digitalen Modelle der relevanten Domänen in der PE.

5.3 Computerunterstützte Modellierung von DMMs

Mit den DMMs wird grundsätzlich das Ziel verfolgt, alle Elemente aus zwei unterschiedlichen DSMs bzw. Domänen, welche miteinander in Beziehung stehen, durch geeignete Relationen zu verbinden (s. Bild 46). So sind bspw. die Anforderungen mit den Produkteigenschaften und deren Merkmalen mit einer DMM zu verknüpfen. In diesem Abschnitt werden verschiedene, im Rahmen der Forschungsarbeiten des Autors dieser Arbeit erforschte, neue **Methoden für die Modellierung von DMMs** vorgestellt.

Eine Methode besteht darin, sich vom Assistenzsystem auf Grundlage der angrenzenden DSMs eine zunächst leere DMM automatisiert erzeugen zu lassen. Diese DMM kann vom Komplexitätsingenieur (ggf. zusammen mit Domänenexperten) entweder **direkt im Assistenzsystem** oder aufgrund der bidirektionalen IT-Schnittstelle in MS Excel ausgefüllt werden.

Eine zweite Methode zielt darauf ab, mit Hilfe des sog. **smarten Mappings und Checks von Worten bzw. Bezeichnungen potenzielle Relationen** zu erkennen. Die Methode beruht auf der Identifizierung aller Elemente unterschiedlicher Domänen, die potenziell miteinander in Beziehung stehen (könnten). So wird bspw. durch diese Methode erkannt, dass die Produktelemente „Rohrinnendurchmesser“, „Rohraußendurchmesser“ und

„Rohrlänge“ mit dem Prozesselement „3D-CAD Modell des Rohrs modellieren“ potenziell eine Relation haben könnten. Dies liegt daran, dass die Methode zunächst alle Bezeichnungen der Elemente in den entsprechenden IT-Systemen (z. B. zur CAD- und Prozessmodellierung) in geeigneter Art und Weise anpasst (z. B. „Rohrlänge“ in „Rohr“ und Länge bzw. „3D-CAD Modell des Rohrs modellieren“ in „3D-CAD-Modell“ und „Rohr“). Durch das Erkennen gleichartiger oder ausreichend ähnlicher Bezeichnungen von Elementen in unterschiedlichen Domänen (z. B. „Rohr“) werden in die DMM potenzielle Relationen an entsprechender Stelle eingetragen. Anschließend werden diese potenziellen Relationen dem Komplexitätssingenieur mit Hilfe einer GUI schrittweise angezeigt, so dass alle potenziellen Relationen in der entsprechend vorausgefüllten DMM bestätigt und bei Bedarf konkretisiert oder verworfen werden können.

In den folgenden beiden Kapiteln werden zwei neu entwickelte Methoden näher vorgestellt, die bei der **Modellierung und Manipulation sowohl von domänenspezifischen als auch domänenübergreifenden Relationen** signifikant unterstützen. So wird in Kapitel 5.3.1 eine Methode zur automatischen und graphbasierten Computerunterstützung mit dem Assistenzsystem und dem yEd-Grafikeditor dargestellt. Anschließend wird der Domain Merger in Kapitel 5.3.2 erläutert, der als Modul des Assistenzsystems v. a. bei der Modellierung und Manipulation von DMMs assistiert.

5.3.1 Grafische Modellierung

Das grundlegende Vorgehen zur Erstellung von DMMs für die Relationen zwischen DSMs – siehe Phase 3 in der übergeordneten Methodik für die Modellierung komplexer Systeme in Bild 46 – wird in Bild 58 in fünf Phasen gegliedert und überblicksartig dargestellt. Mit diesem, in dieser Arbeit auf wissenschaftlichen Grundlagen (vgl. u. a. A.18 und A.19) erarbeiteten, Vorgehen stellt das entwickelte Assistenzsystem Anwendern eine **grafische, benutzerfreundliche Unterstützung zur Modellierung von DMMs** zur Verfügung. In dieser methodischen Vorgehensweise werden die ersten zwei Phasen durch das Assistenzsystem und die Phase 4 durch den kostenlosen yEd Graph Editor (yEd) unterstützt, welcher über eine bidirektionale Schnittstelle in das Assistenzsystem integriert ist (vgl. Phase 3 und Phase 5 in Bild 58).

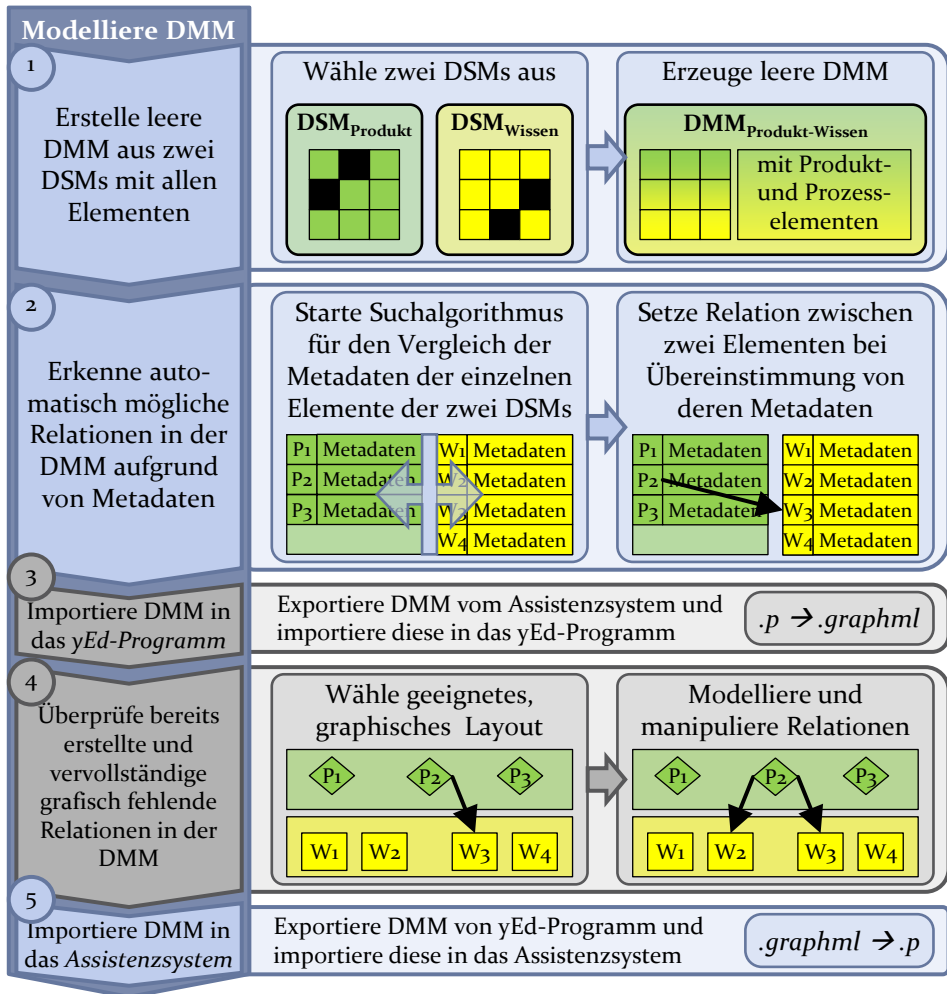


Bild 58: Computerunterstütztes Vorgehen zur graphbasierten Modellierung von DMMs

Leere DMM aus zwei DSMs mit allen Elementen

Nach der Modellierung aller für eine Aufgabenstellung relevanten DSMs sind jeweils die Relationen zwischen den Elementen der zwei DSMs mit Hilfe der entsprechenden DMM zu modellieren. Dazu sind wie in Bild 58 skizziert die zwei DSMs auszuwählen (z. B. $DSM_{Produkt}$ und DSM_{Wissen}), für die eine DMM erzeugt werden soll, die zwar alle Elemente der **zwei DSMs in der Form $DMM_{Produkt-Wissen}$, aber noch keine Relation** enthält. Dafür steht im Assistenzsystem eine Funktion bereit, die die Elemente mit relevanten Metadaten in ein *dictionary* für die zu erstellende $DMM_{Produkt-Wissen}$ integriert. Dabei wird sichergestellt, dass zum einen die Elemente in der DMM auch den jeweiligen DSMs zugeordnet bleiben und zum anderen in

der DMM zwischen Ausgangs- und Zielelementen differenziert wird (z. B. die Produktelemente als Ausgangselemente beeinflussen die Prozesselemente als Zielelemente in Bild 58 und nicht umgekehrt).

Im Rahmen von Anwendungsbeispielen (vgl. verschiedene, automatisiert erstellte DSM_{Produkt} in Kapitel 6.1 und Anhang A.22) hat sich bei der sowohl automatischen als auch manuellen Modellierung von Relationen gezeigt, dass eine aussagekräftige und einheitliche **Bezeichnung der Elemente in den Anwendungssystemen** und damit in den DSMs essenziell ist, um diese effektiv und effizient in DMMs in Beziehung zueinander setzen zu können. Da dies in der industriellen Praxis schwer realisierbar ist und dies auch bspw. bei einigen Merkmalen zu Dubletten (i. S. v. redundanten Bezeichnungen, z. B. kommt das Merkmal „Länge“ oft in vielen Einzelteilen vor) führen kann, besteht im Assistenzsystem die Option ausschließlich die eindeutigen IDs von Elementen zu verwenden. Durch die IDs sind zwar die Elemente eindeutig vom Assistenzsystem identifizierbar, jedoch nicht vom Anwender den Elementen zuordenbar und damit nutzbar. Daher werden im Assistenzsystem als auch in yEd in einem kontextspezifischen Fenster zu jeder ID die Bezeichnung und optional weitere Metadaten des Elements angezeigt (z. B. zum Merkmal „Länge“ auch dessen .prt-Zugehörigkeit).

Relationen aufgrund der Übereinstimmung von Metadaten

Ähnlich wie bei der zuvor skizzierten Methode „smart Mapping und Check“ von Worten bzw. Bezeichnungen werden in der zweiten Phase **automatisch potenzielle Relationen zwischen Elementen der DMM erkannt und gesetzt** (s. Bild 58). Dazu wird nach Erstellung der DMM (z. B. leere $DMM_{\text{Produkt-Wissen}}$) ein neu entwickelter Suchalgorithmus gestartet, welcher automatisch potenzielle Relationen in der DMM setzt. Dieser Algorithmus beruht v. a. darauf, dass gewisse Produktelemente bestimmte Arten von Wissensselementen i. d. R. beeinflussen, wie bspw. geometrische Merkmale die Wissensselemente z. B. „CAD-Dokument“ und „Simulationsdokument“. So haben bspw. die Merkmale des Einzelteils Rohr (z. B. *Rohrlänge*) i. d. R. einen Einfluss auf die entsprechende CAD-Datei (z. B. *rohr.prt*). Dies liegt daran, dass sich bei Erstellung oder Modifikation des Parameters im CAD-System auch die entsprechende CAD-Datei ändert.

Die prinzipielle **Funktionsweise des Suchalgorithmus** wird am Beispiel der Modellierung einer Relation zwischen einem Produkt- und Wissensselement nachfolgend erläutert. Der Algorithmus extrahiert mittels einer *for*-Schleife alle relevanten Metadaten der einzelnen Elemente der DSM_{Produkt} und speichert die gewonnenen Informationen in strukturierter Form ab. So werden z. B. beim Produktelement „Länge“ u. a. die zwei Arten von

Metadaten Einzelteil (z. B. mit der Information *rohr.prt*) und Baugruppe (z. B. mit der Information *antriebseinheit.asm*) vom Suchalgorithmus gefunden. Somit „weiß“ der Algorithmus, dass das Merkmal „Länge“ zum Einzelteil *Rohr* und dieses wiederum ein Teil der Baugruppe *Antriebseinheit* ist. Diese und weitere durch den Suchalgorithmus gefundene Metadaten dienen als Basis für den Vergleich der Metadaten der Produktelemente mit den Metadaten der Wissens Elemente (s. Bild 58). Falls bspw. die Bezeichnung eines Wissens Elements in den Metadaten der Produktelemente gefunden ist, wird automatisch zwischen beiden Elementen eine Relation gesetzt (z. B. zwischen Produktelement *P2* und Wissens Element *W3* in Bild 58). Somit wird der Komplexitätsingenieur bei der Modellierung von DMMs durch das automatische Erkennen und Setzen von potenziellen Relationen unterstützt.

Import der DMM in das yEd-Programm

Die in der ersten und zweiten Phase automatisiert modellierten Elemente und Relationen der DMM (s. Bild 58 z. B. $DMM_{\text{Produkt-Wissen}}$) werden in der Wissensbasis des Assistenzsystems (vgl. *prodkid_edges.p* in Bild 48, d. h. in der DMM-Prod.-Kid) gespeichert. Diese DMM wird in der dritten Phase (s. Bild 58) aus dem Assistenzsystem in Form einer *graphml*-Datei exportiert, so dass die **DMM in das yEd-Programm importiert** werden kann. Dabei werden mit Hilfe der im Assistenzsystem implementierten Funktionen alle Elemente und Relationen zusammen mit den jeweiligen, relevanten Metadaten mit Hilfe der *graphml*-Datei an das yEd-Programm übergeben. Der Import von DMMs in das yEd-Programm ist für alle in dieser Arbeit angeführten Domänen möglich. Zudem ist aufgrund der objektorientierten Programmierung der Import **auf weitere Domänen einfach skalierbar**.

Grafische Überprüfung und Vervollständigung von Relationen

Für die grafische **Überprüfung und Vervollständigung von Relationen ist der kostenlose yEd Graph Editor** in das Assistenzsystems integriert. Mit der Visualisierungssoftware yEd lassen sich eine Vielzahl verschiedener Diagrammarten effizient modellieren, manipulieren und darstellen, wie bspw. Prozessdiagramme (z. B. BPMN-, ERM/EPK- und UML-Diagramme), Organigramme und Ontologien. Diese Diagramme werden in yEd im *xml*-basierten GraphML-Format gespeichert.

Nach dem Import der *graphml*-Datei in yEd ist ein **geeignetes Graphlayout zur visuellen Darstellung der DMM** zu wählen. Hierfür kann z. B. ein bipartiter Graph (s. Bild 58) verwendet werden (s. Kapitel 3.3.2 für weitere Graphen und Layouts). Mit Hilfe des bipartiten Graphs werden bspw.

die Produkt- und Wissens Elemente gruppiert dargestellt und zur besseren Unterscheidung in zwei unterschiedlichen Farben eingefärbt. Zudem kann mit Hilfe des hierarchischen Layouts die Überprüfung der Relationen auf Richtigkeit und Vollständigkeit unterstützt werden. Dadurch kann der Anwender in yEd die zuvor automatisch erstellten Relationen effizient überprüfen und nach Bedarf bearbeiten als auch neue Relationen intuitiv hinzufügen. Die Modellierung und Manipulation von Relationen erfolgt durch Anklicken des Ausgangselements (z. B. Produktelement P_2) und des Zielelements (z. B. Wissens element W_2 , vgl. Bild 58).

Dabei gewährleisten diverse Funktionen des Assistenzsystems, dass die vom Anwender in yEd möglicherweise (falsch) erstellten neuen Elemente, modellierten Relationen zwischen Elementen derselben Domäne und gesetzte Relationen von Ziel- zu Ausgangselementen im Rahmen der DMM-Modellierung ignoriert werden. Somit bleiben die möglicherweise aus Versehen oder die fälschlicherweise erstellten Elemente und Relationen unberücksichtigt. Dadurch wird die geforderte **Minimierung des Aufwands und der Fehleranfälligkeit** realisiert (vgl. Kapitel 4.2.2).

Import der DMM in das Assistenzsystem

Abschließend wird die modellierte DMM in yEd als eine *.graphml*-Datei gespeichert und in das Assistenzsystem importiert. Da sich diese *graphml*-Dateien gut zur Realisierung der **bidirektionalen Schnittstelle zwischen yEd und dem Assistenzsystem** eignen, wird yEd für die Modellierung von Relationen in den DMMs mit Hilfe der vorgestellten, graphbasierten Methode in das Assistenzsystem integriert.

5.3.2 Domain Merger

Eine vierte Methode zur (teil-)automatisierten Modellierung von DMMs stellt der sog. Domain Merger als weiteres Modul des Assistenzsystems dar. Der Domain Merger ist ein Teil des matrixbasierten Modellierungsmoduls und umfasst v. a. Integrations- und Manipulationsmethoden (s. Bild 48). Der Aufruf des Moduls Domain Merger erfolgt über die Menüleiste der GUI unter Manipulation Tools (s. Bild 49). Der **Domain Merger unterstützt über eine assistierende GUI bei der Modellierung der domänenübergreifenden Relationen** (s. Bild 59). Die hierfür vom Domain Merger bereitgestellten Funktionalitäten werden nachfolgend kurz erläutert.

Grundsätzlich werden über die **zweigeteilte GUI des Domain Mergers** auf der linken Seite alle Ausgangselemente und auf der rechten Seite alle Zielelemente in jeweils einer Auswahlliste angezeigt (s. Bild 59). Somit

beeinflussen im Sinne einer Parents-Child-Beziehung die Ausgangselemente der linken Seite die Zielelemente der rechten Seite. Die zwei Auswahllisten lassen sich jeweils mit Hilfe eines Domänenfilters (z. B. in Bild 59 „Process“ und „Organisation“) nach den Elementen der selektierten Domänen filtern. Mit Hilfe der zwei Schaltflächen „>“ und „<“ erfolgt das Hinzufügen bzw. das Entfernen sowohl der Ausgangselemente in die Liste „Parent nodes“ als auch der Zielelemente in die Liste „Child nodes“ aus den zwei zuvor gefilterten Auswahllisten (z. B. „S14: Festlegung Hauptabmessung“ und „O4: Konstrukteur Lagerung“). Mit dem Button „clear“ besteht auch die Option alle hinzugefügten Ausgangs- bzw. Zielelemente zu löschen.

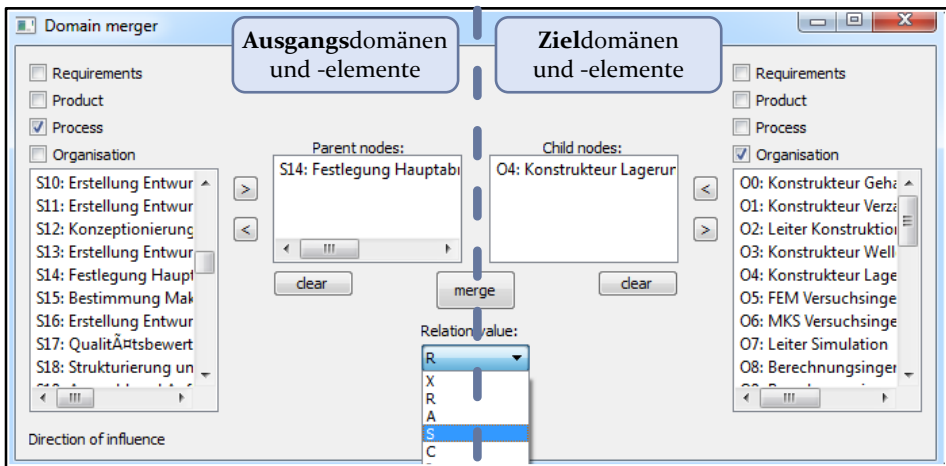


Bild 59: Übersicht der GUI des Moduls Domain Merger mit ausgewählten Funktionalitäten

Grundsätzlich können mittels des Buttons „merge“ zwischen mindestens einem Ausgangselement in der Liste „Parent nodes“ und mindestens einem Zielelement in die Liste „Child nodes“ Relationen mit dem identischen Wert (s. selektierter Wert im Drop-down-Menü „Relation value“ in Bild 59) erzeugt und in die Wissensbasis des Assistenzsystems gespeichert werden. Allerdings ist es i. d. R. aufgrund einer geringeren Fehleranfälligkeit sinnvoller **mit der merge-Funktion schrittweise je eine Relation zwischen einem Ausgangs- und einem Zielelement** zu modellieren.

Für die **Modellierung der Art der Relationen mit der merge-Funktion** stehen im Drop-down-Menü „Relation value“ verschiedene Auswahlmöglichkeiten zur Verfügung (s. Bild 59). Dabei werden vom Assistenzsystem je nach Selektion der Ausgangs- und Zielelemente bzw. deren Domänen nur die zulässigen Arten von Relationen im Drop-down-Menü angezeigt. So sind bspw. in Folge der Selektion „Process“ als Ausgangsdomäne und „Organisation“ als Zieldomäne neben der Standardrelationenart „X“

aufgrund des Kontextbezugs nur die RASCI-Notationen als „*Relation value*“ für den Komplexitätsingenieur kontextbezogen verfügbar. Dabei dient das RASCI-Modell (d. h. Responsible, Accountable, Support, Consulted und Informed) als Erweiterung des verbreiteten RACI-Modells Rollen mit verschiedenen Prozessschritten zu assoziieren. Darüber hinaus sind im Drop-down-Menü „*Relation value*“ je nach Kontext eine Vielzahl weiterer numerischer und alphanumerischer Notationen verfügbar. So stehen z. B. bei Selektion von „*Process*“ auf beiden Seiten die Werte „AND“, „OR“ und „XOR“ zur Verfügung. Zudem ist bei Bedarf auch das Anlegen zusätzlicher Werte im Drop-down-Menü „*Relation value*“ möglich.

Die Modellierung der definierten Relation(en) zwischen den i. d. R. zwei Elementen erfolgt mit „*merge*“. Im Rahmen der merge-Funktion und vor Speicherung der modellierten Relationen werden noch einige Tests durchlaufen, wie z. B. die **Prüfung auf bereits bestehende Relationen** zwischen den (zwei) beteiligten Elementen. Falls zwischen den Elementen bereits Relationen bestehen, werden diese dem Komplexitätsingenieur über ein Auswahlfenster mit folgenden Entscheidungsoptionen angezeigt:

- Bestehende Relation(en) durch neue Relation(en) ersetzen („Yes / No“),
- Neue Relation(en) zusätzlich zu vorhandener/en Relation(en) erstellen („Yes / No“),
- Bestehende Relation(en) beibehalten und neue Relation(en) verwerfen („Yes / No“),
- Bestehende und neue Relation(en) verwerfen („Yes / No“).

Fazit zur computerunterstützten Modellierung von DMMs

In diesem Kapitel wurden auf Basis wissenschaftlicher Erkenntnisse vier miteinander kombinierbare Methoden für die computerunterstützte Modellierung von DMMs entwickelt und vorgestellt. So wird der Anwender u. a. durch das automatische Erkennen potenzieller Relationen aufgrund von Metadaten sowie durch eine grafische, benutzerfreundliche Unterstützung mittels des yEd-Grafikeditors bei der Erstellung von DMMs unterstützt. Zudem wird auch mit dem Modul Domain Merger des Assistenzsystems die Modellierung und Manipulation von DMMs erheblich erleichtert. Insgesamt wird dadurch eine besonders effiziente Erstellung von DMMs realisiert, welche für die domänenübergreifenden, matrixbasierten Modelle erforderlich ist.

5.4 Graphbasierte Visualisierung und Kommunikation

Neben der Definitions- und Modellierungsphase des in Bild 45 skizzierten Vorgehens zum matrixbasierten Management komplexer Systeme steht im Folgenden die wissenschaftliche Entwicklung von Konzepten für die Phase der Analyse im Fokus. Die Analysephase lässt sich gemäß Bild 47 in die Anwendung von zweckmäßigen Visualisierungs- sowie von zielgerichteten Analysemethoden gliedern. Da für die Analyse und Bewertung komplexer Modelle sowie v. a. für deren Kommunikation die diversen graphbasierten Repräsentationsformen im Vergleich zu matrixbasierten Darstellungen erheblich besser geeignet sind (s. für Details Anhang A.19), stehen in diesem Kapitel **computerunterstützte Visualisierungsmethoden zur Visualisierung, Kommunikation und Bewertung komplexer Systeme** im Fokus. Dazu werden in Kapitel 5.4.1 die internen Visualisierungsmöglichkeiten des Assistenzsystems erläutert. In Kapitel 5.4.2 werden externe IT-Werkzeuge zur Darstellung von matrixbasierten Modellen beleuchtet, die in das Visualisierungsmodul des Assistenzsystems integriert sind.

Die zur **Visualisierung von matrixbasierten Modellen** vom Autor als positiv bewerteten graphbasierten Visualisierungsansätze sind v. a. gerichtete Graphen, Wurzelbäume, freie Bäume und Multigraphen (s. Kapitel 3.3.2). Bei der Verwendung dieser und weiterer Arten von Graphen zeigt sich insb. bei Modellen mit vielen Elementen und Relationen die große Herausforderung einer übersichtlichen und benutzerfreundlichen graphbasierten Visualisierung – was jedoch bei allen Visualisierungstools (s. Kapitel 3.3.4) eine Schwierigkeit darstellt. Aus diesem Grund stellt das Assistenzsystem dem Entwickler folgende Optionen generell zur Verfügung:

- Vielfältige Visualisierungsoptionen mittels v. a. planaren, orthogonalen, hierarchischen, kreisförmigen und kräfteorientierten Layouts von Graphen (vgl. Kapitel 3.3.2) sowie Kombinationen davon
- Kontextspezifische Visualisierungsmöglichkeiten in Form der für die jeweilige Zielsetzung zweckmäßigen Darstellungen von allein einer DSM oder DMM oder mehreren, kombinierten DSMs, DMMs oder der gesamten MDM (s. in der GUI in Bild 49 die hellblau eingefärbten Buttons unter der jeweiligen Matrix bzw. den Matrizen)
- Computerunterstützte Visualisierungsmöglichkeiten mit Hilfe einer schrittweisen, kontextabhängigen Benutzerführung, die sich je nach Aktion des Benutzers adaptiv anpasst (z. B. s. Bild 62)
- Interne als auch externe Visualisierungsmöglichkeiten mit Hilfe des Visualisierungsmoduls des Assistenzsystems sowie der Möglichkeit der Integration von externen Visualisierungstools (s. Bild 48)

5.4.1 Visualisierungsmodul

Das **generelle Vorgehen zur Visualisierung matrixbasierter Modelle mit dem graphbasierten Visualisierungsmodul des Assistenzsystems** gliedert sich in die in Bild 60 dargestellten drei Phasen. Mit dem Vorgehen, das sowohl auf die jeweils individuellen Bedürfnisse als auch auf die jeweils spezifischen Zielsetzungen flexibel angepasst werden kann, unterstützt das Assistenzsystem bei der Visualisierung und somit Analyse, Bewertung und Kommunikation von Modellen komplexer Systeme. Da es zum einen aufgrund der großen Vielzahl und Vielfalt von Elementen und deren Relationen und zum anderen aufgrund der verschiedenen Aufgaben- und Problemstellungen des Komplexitätsingenieurs nicht möglich und v. a. sinnvoll ist, eine einzige Visualisierungsmethode oder auch Graphdarstellung zu optimieren, wird vom Autor die Zielsetzung verfolgt, möglichst kontextspezifisch geeignete Visualisierungsmöglichkeiten mit vielen nützlichen Optionen computerunterstützt zur Verfügung zu stellen.

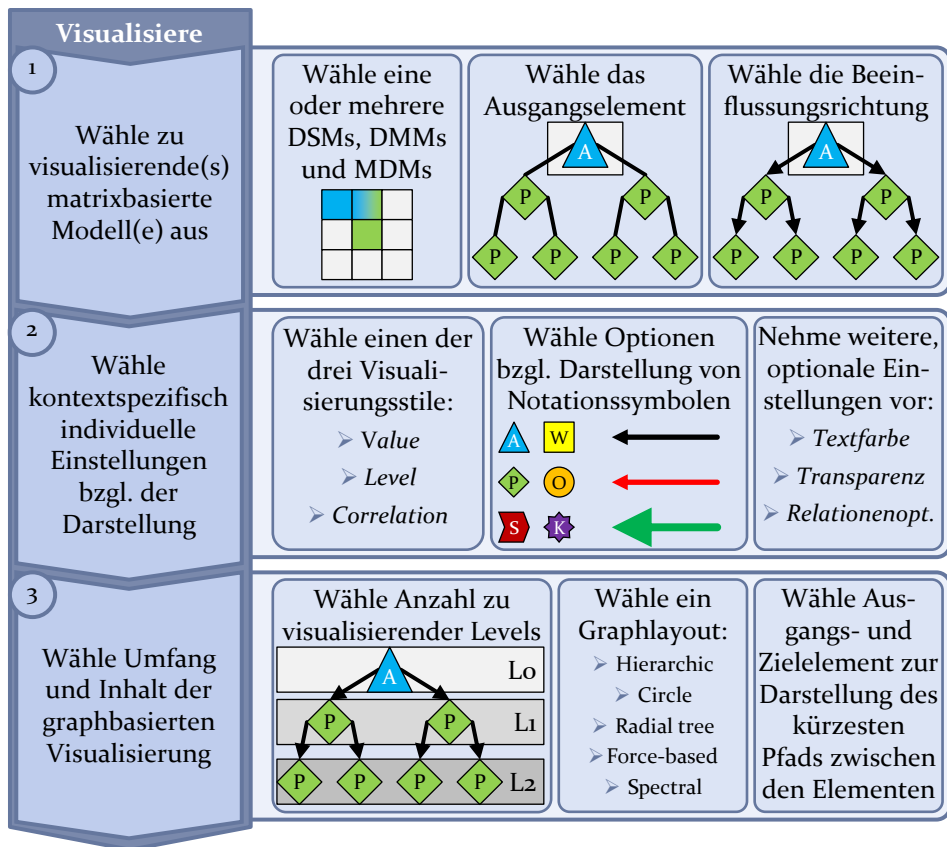


Bild 60: Computerunterstütztes Vorgehen zur graphbasierten Visualisierung

Nach der **Auswahl der zu visualisierenden Matrix oder Matrizen** (z. B. $DSM_{Anforderung}$, $DSM_{Produkt}$, $DMM_{Anforderung-Produkt}$ in Bild 6o) über die GUI des Assistenzsystems ist durch den Anwender zu entscheiden, ob das matrixbasierte Modell entweder mit dem graphbasierten Visualisierungsmodul oder über eine Schnittstelle in einem externen Visualisierungsprogramm (s. Kapitel 5.4.2) dargestellt werden soll. Sofern das interne Visualisierungsmodul des Assistenzsystems gewählt wird, erfolgt der Abruf der relevanten Daten zu den relevanten Elementen und deren Relationen aus der Wissensbasis (z. B. *ID*, *name*, *type* für Parameter aus der Produktdomäne, s. Kapitel 5.2.1). Während die eindeutige Identifizierung der Elemente und Relationen im Assistenzsystem intern über die jeweiligen IDs erfolgt, kann der Komplexitätsingenieur für die Visualisierung z. B. auch (Kurz-)Bezeichnung oder Typ der Elemente und Relationen wählen (s. Bild 61). Danach ist je nach Zielsetzung der Visualisierung ein Ausgangselement (i. S. eines Wurzelknotens) aus den selektierten Elementen zu bestimmen (z. B. das Anforderungselement A_3 wie in Bild 61). Zudem ist eine der zwei möglichen Beeinflussungsrichtungen des ausgewählten Ausgangselements festzulegen. Dabei werden je nach Festlegung der Beeinflussungsrichtung (i. S. v. Zeile beeinflusst Spalte vs. Spalte wird von Zeile beeinflusst) entweder die Relationen ausgehend vom Ausgangselement dargestellt (wie die in Bild 61 orange eingefärbten Relationen ausgehend von Anforderung A_3) oder die Relationen der Elemente, die das Ausgangselement beeinflussen.

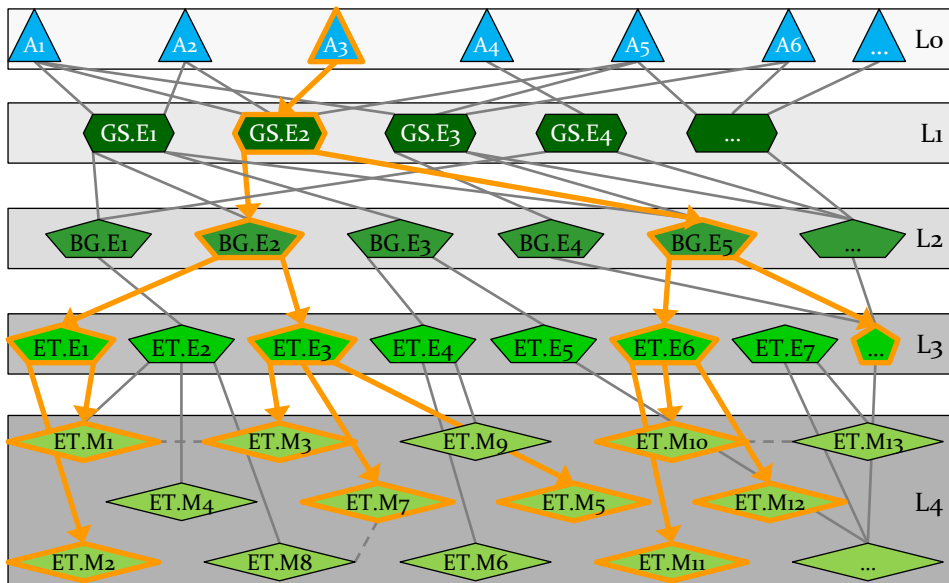


Bild 61: Wahl des Ausgangselements, der Beeinflussungsrichtung und Anzahl der Levels sowie Darstellung von verschiedenen Pfaden

Je nachdem welche matrixbasierten Modelle für welche Zielstellung visualisiert werden, sind vom Komplexitätsingenieur **in der zweiten Phase kontextspezifisch individuelle Einstellungen für die Darstellung vorzunehmen**. So stehen dem Anwender drei Visualisierungsstile, eine Vielzahl an Optionen zur Darstellung der Notationssymbole für v. a. Elemente und Relationen sowie zahlreiche optionale Einstellungen (z. B. Textfarbe, Transparenz und Relationen-Optionen) zur Verfügung (s. Bild 60).

Die gegenwärtig im Assistenzsystem **vorhandenen Visualisierungsstile** sind *Value*, *Level* und *Correlation* (s. Bild 62). Diese dienen dem Anwender zur Darstellung komplexer Modelle bei unterschiedlichen Zielsetzungen (s. für Beispiele v. a. [S23]).

Mit dem **Visualisierungsstil Value** werden die Relationen nach deren Wertigkeit der Wechselwirkung bzw. Stärke der Beeinflussung (i. S. v. Gewichtung) eingefärbt, wobei die bspw. in Bild 62 dargestellte Skala *Values* von +5 für die größte, positive Wechselwirkung (dunkelgrün) bis -5 für die größte, negative Wechselwirkung (dunkelrot) aufweist. Für eine bessere Unterscheidung lassen sich alle Elemente mit einer positiven (negativen) Relation zum Ausgangselement auch rechtsseitig (linksseitig) anordnen. Zudem dient auch die Distanz als Differenzierungsoption. Dabei werden Elemente mit einer geringen (großen) Wechselwirkung in kleiner (großer) Distanz voneinander abgebildet. Als eine weitere Option ist das Selektieren bestimmter Relationen möglich, wie bspw. der Auswahl von nur besonders starken, negativen Wechselwirkungen zwischen Elementen. Sofern z. B. die Wertigkeit zwischen zwei Elementen nicht bestimmt werden kann, wird die Relation mit einem „x“ symbolisiert.

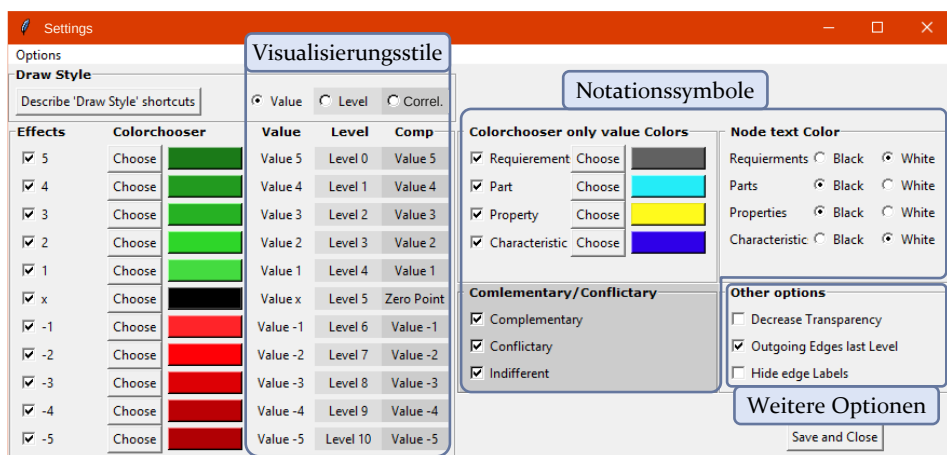


Bild 62: Einstellungen bzgl. Visualisierungsstil, Notationssymbol und weiterer Optionen

Mit dem **Visualisierungsstil *Level*** werden anstatt der *values* die *levels* (i. S. v. hierarchischen Ebenen) unterschiedlich eingefärbt. So können ausgehend vom ausgewählten Ausgangselement mit dessen Relationen auf *Level* o Elemente in Abhängigkeit ihrer Zugehörigkeit zu einem *Level* in verschiedenen Farben eingefärbt werden (s. z. B. Bild 61). Die mit Hilfe des Visualisierungsstils *Level* dargestellten Graphen können bspw. durch die beiden weiteren Optionen *Decrease transparency* und *Hide edge Labels* übersichtlicher gestaltet werden. So werden zum einen Elemente mit deren Relationen mit steigender Levelnummer transparenter und zum anderen Bezeichnungen von Relationen ausgeblendet.

Der **Visualisierungsstil *Correlation*** dient insb. für die Darstellung von komplementären, konkurrierenden (bzw. konfliktären) und indifferenten (bzw. neutralen) Zielbeziehungen. Daher ist dieser Visualisierungsstil nur für bestimmte DSMs und DMMs geeignet, wie bspw. der $\text{DSM}_{\text{Anforderung}}$ und der $\text{DMM}_{\text{Anforderung-Produkt}}$ (s. für weitere Beispiele [P32]). So lassen sich mit diesen beiden Matrizen die unterschiedlichen Zielbeziehungen zwischen den Anforderungen und den Produkteigenschaften darstellen. Dabei erfolgt die Färbung der Relationen nach der Art und Wertigkeit der Wechselwirkung (vgl. Ähnlichkeit zum Stil *Value*), und zwar von +5 für die stärkste komplementäre bis -5 für die stärkste konkurrierende Zielbeziehung. Zudem erfolgt auch die Abbildung indifferenter Elemente (mittels einer anderen Farbgebung), die in keiner Beziehung zum gewählten Ausgangselement stehen. Ein wesentlicher Unterschied zu den beiden zuvor erläuterten Visualisierungsstilen ist, dass beim Stil *Correlation* lediglich die direkten Relationen anstatt des ganzen Netzwerks dargestellt werden.

Neben diesen drei Visualisierungsstilen stehen zur kontextspezifischen, nutzergerechten Darstellung der matrixbasierten Modelle auch diverse **Optionen zur Darstellung von Notationssymbolen sowie weiteren Einstellungen** zur Verfügung. So ist unabhängig vom selektierten Stil die Abwahl von Relationen mit bestimmten *Values* bzw. *Levels* möglich, um aufgrund der dadurch geringeren Anzahl an Elementen und Relationen die Übersichtlichkeit zu erhöhen. Dadurch kann sich der Anwender bspw. auch nur besonders starke Beziehungen zwischen Elementen oder allein die Relationen des Ausgangselements bis zum zweiten Grad (bzw. Ebene) darstellen lassen. Für eine bessere und bedarfsgerechte Darstellung dienen auch die verfügbaren Optionen, wie z. B. zur Farb- und Symbolauswahl für Elemente sowie Art und Stärke von Linien und Pfeilen für Relationen. Ferner kann zur besseren Lesbarkeit auch die Textfarbe und -größe eingestellt werden.

In der dritten Phase erfolgt die **Auswahl des Umfangs und Inhalts der graphbasierten Visualisierung**. Aufgrund der Vielzahl an Elementen und Relationen von matrixbasierten Modellen komplexer Systeme dienen Darstellungen des Gesamtsystems nur für die grobe Orientierung und der Identifizierung relevanter Elemente. Daher ist es oft sinnvoll, die Anzahl zu visualisierender Ebenen ausgehend vom Ausgangselement zu wählen, wobei die maximale Zahl an Systemebenen kontextspezifisch vom Assistenzsystem angezeigt wird. Zudem besteht die Möglichkeit zur Darstellung des kürzesten Pfads zwischen ausgewählten Ausgangs- und Zielelementen.

Das graphbasierte Visualisierungsmodul stellt für die **ziel- und bedarfsgerechte Darstellung verschiedene Graphlayouts** bereit (s. Bild 60 und für Grundlagen Kapitel 3.3.2). So können durch das hierarchische Layout wie in Bild 61 gezeigt die (Pfade der) Relationen vom Ausgangselement in *Level 0* (z. B. Anforderung *A3*) über die Gesamtsystemeigenschaften in *Level 1* (z. B. *GS.E2*), Baugruppeneigenschaften in *Level 2* (z. B. *BG.E2*, *BG.E5*) und Einzelteileigenschaften in *Level 3* (z. B. *ET.E1*, *ET.E3*, *ET.E6*) bis zu den Einzelteilmerkmalen in *Level 4* (z. B. *ET.M1*, *ET.M2*, *ET.M3*) systematisch abgebildet werden. Dabei werden mittels des hierarchischen Layouts auch Relationen in einer Ebene (z. B. zwischen *ET.M1* und *ET.M3*) sowie direkte und indirekte Relationen nachvollziehbar. Beim kreisförmigen Layout ist die Größe des Kreises von der Anzahl an Elementen abhängig. Diese Anordnung ist insb. für einen ersten Überblick hilfreich, da sich die Relationen trotz vieler Überschneidungen aufgrund der gleichmäßigen Verteilung der Elemente gut nachvollziehen lassen. Durch das Radiale Baum-Layout werden alle Elemente in Abhängigkeit von ihrer Ebene in konzentrischen Kreisen um das Ausgangselement herum positioniert, wodurch im Vergleich zum hierarchischen Layout die Relationen innerhalb einer Ebene übersichtlicher dargestellt werden. Das vierte implementierte Graphlayout ist ein kräftebasiertes Layout nach dem Fruchterman-Reingold-Algorithmus. Demgemäß liegen diejenigen Elemente umso näher zusammen, je mehr Relationen diese untereinander haben und je stärker diese sind. Das spektrale Graphlayout dient v. a. der Visualisierung von besonders eng miteinander in Beziehung stehenden Elementen. Daher sind spektrale Layouts insb. zur Analyse und Bewertung von Clustern von Vorteil.

Das im Rahmen der Forschungsarbeiten des Autors implementierte **graphbasierte Visualisierungsmodul des Assistenzsystems** (s. Bild 48) nutzt v. a. die Python-Bibliotheken *NetworkX* und *Matplotlib* (s. für Details Anhang A.26). Dabei ist von Vorteil, dass diese und weitere verwendete Bibliotheken fortlaufend weiterentwickelt werden, so dass auch zukünftig mit Verbesserungen der Visualisierungsmethoden zu rechnen ist.

5.4.2 Visualisierungsprogramme

Neben dem graphbasierten Visualisierungsmodul stehen dem Komplexitätsingenieur auch **externe Visualisierungsprogramme zur Visualisierung matrixbasierter Modelle** zur Verfügung. Aufgrund der vielfältigen Exportmöglichkeiten des Assistenzsystems (vgl. Anhang A.26) lassen sich die matrixbasierten Modelle auch in andere IT-Werkzeuge exportieren, um in diesen mit deren Funktionalitäten grafisch dargestellt werden zu können. Dafür eignen sich zwar prinzipiell auch die Visualisierungsfunktionalitäten von CAM, Loomio und Soley (s. Kapitel 3.3.4), aber auf Grundlage der Forschungsarbeiten des Autors sind diese nur bedingt und stark eingeschränkt geeignet (s. für detailliertere Erläuterungen [P27], [P34]).

Stattdessen werden auf Basis der Erkenntnisse des Autors die verbreiteten Visualisierungswerkzeuge, wie z. B. Gephi, Graphviz und Tulip, aufgrund ihres größeren Funktionsumfangs und ihrer besseren Eignung zur Visualisierung bevorzugt. Dies wird in diesem Kapitel am Beispiel des **externen Visualisierungsprogramms Tulip** skizziert. Die Autoren von Tulip verstehen unter dieser wissenschaftlichen Visualisierungssoftware ein Framework zur Informationsvisualisierung, das zur Analyse und Visualisierung von relationalen Daten dient [25]. Tulip ist für diese Arbeit und als integratives Modul des Assistenzsystems (mittels bidirektionaler Schnittstelle) v. a. aus den folgenden zwei Gründen besonders zweckmäßig: Erstens wurde auf Basis der Tulip-Python-Bibliothek (d. h. Vielzahl von Tulip-Python-Paketen) fast der gesamte Inhalt der Tulip C++ API in Python und somit im Assistenzsystem verfügbar gemacht (s. Anhang A.24); zweitens stellt Tulip besonders viele und geeignete Funktionalitäten zur graphbasierten Visualisierung, Analyse und Manipulation von Modellen komplexer Systeme bereit. So sind in Tulip über 40 verschiedene Graphlayouts verfügbar, wovon v. a. die sechs hierarchischen und zwölf stärkebasierten Layouts für diese Arbeit hilfreich sind. Im Gegensatz zu anderen Visualisierungssystemen kann Tulip auch Adjazenzmatrizen darstellen. Ein weiterer Vorteil von Tulip ist, dass der kürzeste Pfad zwischen zwei selektierten Elementen einfach ermittelt und grafisch abgebildet werden kann.

Tulip stellt neben einer Vielzahl von Visualisierungsmöglichkeiten auch viele **Funktionalitäten zur Erstellung, Manipulation und Analyse von Graphen** zur Verfügung (s. für Details [25], [488]). So kann z. B. die Farbgebung von Elementen und deren Relationen sowohl manuell als auch computerunterstützt auf Basis zuvor definierter Kriterien erfolgen. Zudem stehen verschiedene Funktionen zum Suchen und Filtern von Elementen und Relationen zur Verfügung, die auch nutzerspezifisch angepasst werden

können. Zudem können zur besseren Übersichtlichkeit auch nur die Elemente mit direkten Relationen des selektierten Elements angezeigt und alle anderen Elemente und Relationen des Modells transparent gemacht werden. Außerdem stellt Tulip auch Möglichkeiten zur hierarchischen Navigation in Graphen bereit. So lässt sich auch in Tulip analog zur Systemhierarchiemodellierung in Bild 30 in Kapitel 3.3.1 ein Graph in einzelne Subgraphen (i. S. v. Clustern) z. B. für eine bessere Übersichtlichkeit aufteilen (vgl. auch Vorgehensweise zur systematischen Zerlegung von DSMs in Submatrizen in Bild 31). Dabei können diese Subgraphen mit denselben Funktionalitäten wie Graphen visualisiert und manipuliert werden. Dazu stehen diverse Funktionen zum Hinzufügen, Modifizieren und Löschen von Elementen und Relationen bereit.

Das **Vorgehen zur graphbasierten Visualisierung mittels Tulip** ist ähnlich zu den drei Phasen des in Bild 60 skizzierten Vorgehens zur graphbasierten Visualisierung mittels des internen Visualisierungsmoduls des Assistenzsystems, wobei sich einzelne Teilschritte unterscheiden. Tulip wird aus der GUI des Assistenzsystems geöffnet (vgl. Bild 49). Dazu stehen zur Visualisierung einer Matrix (z. B. $\text{DSM}_{\text{Produkt}}$) unter jeder DSM und DMM in der GUI des Assistenzsystems (vgl. Bild 49) hellblau eingefärbte Tulip-Buttons zur Verfügung. Ferner können auch mehrere Matrizen sowie die gesamte MDM (z. B. mit dem Button *Visualize Tulip graph MDM*) in Tulip über das implementierte Modul „*SHOW_TLP*“ (s. Anhang A.24) geöffnet werden. Dabei werden die zu den jeweiligen Elementen und Relationen in der Wissensbasis des Assistenzsystems gespeicherten Informationen bzw. Attribute (z. B. *ID*, *Bezeichnung*, *Wert*, vgl. *dictionary* für einen Parameter in Kapitel 5.2.1) als auch die Einstellungen bspw. bzgl. der Darstellung von Notationssymbolen und Graphlayouts (s. Bild 62) durch die implementierte, integrative Schnittstelle übernommen. In Tulip kann der Anwender im Vergleich zum graphbasierten Visualisierungsmodul des Assistenzsystems mehr Funktionalitäten zur Visualisierung, Analyse und Manipulation von matrixbasierten Modellen komplexer Systems nutzen. Danach erfolgt der Import des (manipulierten) Modells von Tulip in das Assistenzsystem über die implementierte Importfunktion (s. in Bild 49 *Import Tulip graph (tlp) to database*). Da Tulip deutlich mehr Funktionen und Einstellungen als das Assistenzsystem bereitstellt, können nur diejenigen vom Assistenzsystem unterstützten übernommen werden. Bei Bedarf sind daher Modifikationen und Erweiterungen des Assistenzsystems oder von Tulip von Entwicklern und Anwendern vorzunehmen, was auf Basis bestehender Python-Bibliotheken und der somit sehr guten Integration zwischen Tulip und dem Assistenzsystem relativ einfach möglich ist (s. Anhang A.24).

Fazit zur graphbasierten Visualisierung und Kommunikation

Computerunterstützte Visualisierungsmethoden dienen v. a. zur zielgerichteten Visualisierung, Kommunikation und Bewertung komplexer Systeme und sind daher entsprechend den Erkenntnissen dieser Arbeit für das holistische Komplexitätsmanagement in der PE von besonderer Bedeutung. Das in diesem Kapitel vorgestellte graphbasierte Visualisierungsmodul des Assistenzsystems stellt mittels eines computerunterstützten Vorgehens zur graphbasierten Visualisierung verschiedene Visualisierungsstile und Graphlayouts für die ziel- und bedarfsgerechte Visualisierung bereit. Zudem ist das Visualisierungsmodul sowohl durch weitere Python-Bibliotheken als auch mit externen Visualisierungsprogrammen wie Tulip flexibel erweiterbar. Somit stellt das vorgestellte Assistenzsystem umfangreiche Methoden und Funktionalitäten zur Visualisierung und Kommunikation von Modellen komplexer Systeme zur Verfügung und zeichnet sich daher durch eine besondere Benutzerfreundlichkeit aus.

5.5 Mathematische Analyse und Bewertung

Die Analysephase besteht neben der Teilphase *Visualisierung und Kommunikation* auch aus der Teilphase *Analyse und Bewertung* der matrixbasiert erstellten Modelle von komplexen Systemen (vgl. Bild 45). Diese **computerunterstützten Methoden zur Analyse und Bewertung komplexer Systeme** stehen im Fokus dieses Kapitels. So sind die im Assistenzsystem implementierten mathematischen Algorithmen zur Analyse von Matrizen Inhalt von Kapitel 5.5.1. Das zur Integration der Anforderungs- bzw. Spezifikationsdomäne neu entwickelte *Requirements Analysis Module* unterstützt bei der multikriteriellen Bewertung der Anforderungserfüllung und ist Gegenstand von Kapitel 5.5.2. In Kapitel 5.5.3 erfolgt die Integration der Kosten mit dem *Cost Analysis Module*, das durch die Berücksichtigung von Kosten das holistische Komplexitätsmanagement komplementiert.

5.5.1 Analysemodul

In der Literatur lassen sich eine Vielzahl von mathematischen Algorithmen finden, die prinzipiell zur **Analyse von matrixbasierten Modellen** geeignet sind (s. Kapitel 3.3.3). Allerdings sind nach Erkenntnissen des Autors viele der Analysemethoden nicht oder nur ungenügend in Softwaresysteme

für das matrixbasierte Komplexitätsmanagement implementiert (s. Kapitel 3.3.4 und die betreuten studentischen Arbeiten des Autors [S9], [S54], [S8]). Der sich daraus ergebende und in Kapitel 4.2.2 skizzierte Handlungsbedarf umfasst daher die Bereitstellung gängiger Analyseverfahren zur Durchführung sowohl individueller Analysen bestimmter Domänen als auch holistischer, domänenübergreifender Analysen in einem Assistenzsystem.

Daher sind vom Autor dieser Arbeit **ausgewählte Analysemethoden im Assistenzsystem** implementiert worden, die sich wie in Bild 63 in die drei Kategorien Analysemethoden für Elemente und Relationen, individuelle, domänenspezifische Analysemethoden sowie holistische, domänenübergreifende Analysemethoden unterteilen lassen (s. Bild 38). Im Rahmen der Implementierung dieser Analyseverfahren lag der Fokus zwar auf der Produkt- und Prozessdomäne, aber die implementierten Analysemethoden sind auch für alle anderen Domänen nutzbar. Bis auf *Clustering via CAM* (d. h. eine bidirektionale Schnittstelle zum CAM) sind alle in Bild 63 dargestellten Analysemethoden im Analysemodul des Assistenzsystems implementiert (s. Anhang A.25), wobei sich dieses bei Bedarf aufgrund des modularen Aufbaus um weitere Analysemethoden ergänzen lässt (s. v. a. Kapitel 3.3.3 und Anhang A.20, z. B. k-Means-Algorithmus).

Analysemethoden für Elemente und Relationen			
▪ Aktivsumme	▪ Aktivität	▪ Startelement	▪ Transitelement
▪ Passivsumme	▪ Kritikalität	▪ Endelement	▪ Bus
Individuelle, domänenspezifische Analysemethoden			
Methoden zur Analyse komponenten- und organisationsbasierter DSMs (Fokus: Produktanalyse)		Methoden zur Analyse aktivitäten- und parameterbasierter DSMs (Fokus: Prozessanalyse)	
Clustering via CAM	Banding – Aktiv- und Passivsumme	Partitioning	Banding – Vorgänger und Nachfolger
Clustering mit ZAKARIAN-Algorithmus		Tearing	
Holistische, domänenübergreifende Analysemethoden			
Methoden zur Analyse von Änderungen und deren Auswirkungen		Methoden zur Analyse von weiteren Zielsetzungen	
Vorwärtsgerichtete Analyse		Partitioning mit Änderungsanfälligkeit	
Rückwärtsgerichtete Analyse		Banding einer MDM	
Change Prediction Index (CPI)		Geschlossene Kreise	

Bild 63: Übersicht über ausgewählte Analysemethoden des Assistenzsystems

Die **Auswahl der domänenspezifischen Analysemethoden**, welche (weiter-)entwickelt und in das Analysemodul implementiert wurden (s. Bild 63), erfolgte auf Grundlage deren Nützlichkeit und Zweckmäßigkeit für die beiden grundsätzlichen Kategorien von DSMs. So ist für die Analyse komponenten- und organisationsbasierter DSMs (z. B. Produktmodelle) das *Clustering* besonders hilfreich, um bspw. geeignete Module für modulare Produktstrukturen zu identifizieren. Dazu stehen dem Komplexitätsingenieur über eine bidirektionale Schnittstelle das *Clustering via CAM* als auch das im Analysemodul implementierte *Clustering mit dem ZAKARIAN-Algorithmus* zur Verfügung. Durch die IT-Schnittstelle zu dem geläufigsten wissenschaftlichen IT-Werkzeug zum matrixbasierten Komplexitätsmanagement wird die in Kapitel 4.2.2 formulierte Anforderung nach einer Exportmöglichkeit für weitere Analysen in andere Softwaresysteme erfüllt. Die Entscheidung für die (Weiter-)Entwicklung und Implementierung des *Clustering mit dem ZAKARIAN-Algorithmus* [629] (s. Anhang A.20 und Anhang A.25) lässt sich u. a. damit begründen, dass mittels des ZAKARIAN-Algorithmus – im Gegensatz zu vielen anderen Clusteranalysen (z. B. auch der in CAM implementiert ist) – neben binäre auch numerische Matrizen geclustert werden können. Zudem können mit dem ZAKARIAN-Algorithmus im Vergleich zu vielen anderen Clusterverfahren sowohl $n \times n$ (d. h. DSMs) als auch $n \times m$ (d. h. DMMs und MDMs) analysiert werden. Außerdem wird mit Hilfe des Matrixplots des *Banding – Aktiv- und Passivsumme* dargestellt, welche Elemente welche anderen Elemente wie stark beeinflussen (oder von diesen beeinflusst werden). Dagegen ist für die Analyse aktivitäten- und parameterbasierter DSMs (z. B. von Prozessmodellen) das *Partitioning* am wichtigsten, da dieser Algorithmus die Prozessschritte in einen Ablauf mit möglichst wenigen Rücksprüngen anordnet. Der wiederum darauf aufbauende *Tearing-Algorithmus* dient zur weiteren Reduzierung noch bestehender, vermeidbarer Iterationen. Zudem werden mittels der Methode *Banding – Vorgänger und Nachfolger* sowohl die direkten als auch alle nachfolgenden und vorhergehenden Elemente (z. B. Prozesse) gezeigt.

Auch die **Auswahl der domänenübergreifenden Analysemethoden** erfolgte auf Basis deren Nützlichkeit und Zweckmäßigkeit für die Analyse. Dabei stand zum einen die Analyse von Änderungen und deren Auswirkungen und zum anderen die Eignung für spezifische Zielsetzungen des Komplexitätsmanagements im Fokus (s. Bild 63). So stehen für das Änderungsmanagement die *vorwärtsgerichtete Analyse*, die *rückwärtsgerichtete Analyse* und der *Change Prediction Index (CPI)* bereit (s. Kapitel 3.3.3 und Anhang A.20 und Anhang A.25), wobei der CPI auf der Methode *Banding – Aktiv- und Passivsumme* aufbaut. Mit Hilfe des CPI-Werts der Elemente

kann der Komplexitätsingenieur z. B. erkennen, welche Elemente (oder sich auch aus Elementen zusammensetzende Einzelteile und Baugruppen) aufgrund ihrer großen Auswirkung auf ein System als kritisch zu bewerten sind. Auf Basis dieser Erkenntnisse lassen sich Systeme an diesen Stellen bspw. flexibler gestalten, um Änderungsketten möglichst zu minimieren und die Anzahl möglicher Änderungsweitergaben zu reduzieren. So sind v. a. die Verstärkerelemente, die eine hohe Aktiv- im relativen Vergleich zur Passivsumme aufweisen, zu reduzieren, da diese die Systemflexibilität aufgrund ihrer hohen Änderungsverbreitung signifikant einschränken. Zu diesem Zweck eignen sich Änderungen absorbierende (Puffer-)Elemente, die die weitere Änderungsausbreitung reduzieren oder sogar verhindern (s. Anhang A.20 und auch [542] für detaillierte Erläuterungen). Zudem sind im Analysemodul für holistische Analysen der MDM, wie bspw. zum Identifizieren von domänenübergreifenden Relationen die zwei Methoden *Partitioning mit Änderungsanfälligkeit* und *Banding von einer MDM* implementiert. Für die Ermittlung von geschlossenen Kreisen von Elementen (auch) aus unterschiedlichen DSMs dient die Methode *geschlossene Kreise*.

5.5.2 Requirements Analysis Module

Mit dem im Folgenden vorgestellten, im Rahmen der Forschungsarbeiten des Autors entwickelten, Modul *Requirements Analysis* werden zur Unterstützung des holistischen Komplexitätsmanagements zwar insb. die zuvor modellierten Anforderungs- mit den Produktelementen in Beziehung gesetzt, aber dieses methodische und computerunterstützte Vorgehen lässt sich auch auf Elemente anderer Domänen erweitern. Dem Entwickler dient das **Modul Requirements Analysis insb. zur Analyse, Visualisierung und Bewertung von Änderungen und deren Auswirkungen** auf bestimmte Systemelemente in Folge der Modifikation eines bestimmten Systemelements. Dadurch unterstützt das Modul bei der zielorientierten Entscheidungsfindung zur Erfüllung von Spezifikationen. Somit stellt das Modul einen wissenschaftlichen Beitrag dar und dient zur Erfüllung der Zielsetzung (s. Kapitel 1.3) und der Handlungsbedarfe (s. Kapitel 4.2.2).

Die steigende Anforderungskomplexität aufgrund von VUCA (s. Kapitel 1) führt v. a. zu häufigeren und komplexeren **Bewertungen der Erfüllung von Zielwerten**, die von Anforderungen vorgegeben werden und durch die Produkteigenschaften der zu entwickelnden Produkte zu erfüllen sind. Die Erfüllung von Zielwerten der Anforderungen sind in der PE wesentliche Kriterien für Bewertungen im PEP (s. Methoden der Entscheidungsfindung von LUFT ET AL. in [P25]). Dabei unterliegen sowohl die Anforderungs- als

auch die Produktelemente Veränderungen über die Zeit. So kann eine Produktänderung dazu führen, dass sich die Zielwerterfüllung bzgl. einiger Anforderungen verbessert und bzgl. anderer verschlechtert. Aus diesem Grund unterstützt das Modul *Requirements Analysis* im Sinne einer anforderungsorientierten Bewertung durch den visuellen Vergleich zwischen Anforderungszielwerten und entsprechenden Produkteigenschaftswerten.

Dazu sind aus der Anforderungsdomäne alle erforderlichen Elemente und Relationen zu akquirieren und als **matrixbasiertes Anforderungsmodell in Form der DSM-Specification** zu speichern (s. Bild 48). Dabei werden ähnlich zu Elementen anderer Domänen auch die Anforderungselemente in Form von Tupeln in der Wissensbasis des Assistenzsystems gespeichert. Jedes *dictionary* für ein Anforderungselement (d. h. *spec_nodes.p* in Bild 48) hat die folgenden *keys* mit den entsprechenden Informationen bzw. Attributen: *ID*, *index*, *name*, *level1*, *level2*, *value*, *unit*, *text*, *type*, *group*, *class*, *interval*, *operator*. Dagegen ist das Tupel für Anforderungsrelationen (d. h. *spec_edges.p*) im Vergleich bspw. zum Tupel für Produktrelationen nur um die Auswahl möglicher Notationssymbole modifiziert. Bei Bedarf können die Tupel und somit die Anforderungsmodellierung aufgrund der Objektorientierung einfach erweitert werden.

Die Definition der *keys* *ID*, *index*, *name*, *level1*, *level2*, *value*, *unit* mit den dazugehörigen Informationen bzw. Attributen für Anforderungselemente ist jeweils analog zu derjenigen für die Produktelemente bzw. Parameter in Kapitel 5.2.1. Zudem stehen im **dictionary für ein Anforderungselement** noch die folgenden *keys* für zusätzliche Zwecke zur Verfügung:

- *text*: Mit *text* wird ein beliebiger Text zur Erläuterung des Anforderungselements erfasst (z. B. zur Konkretisierung der Spezifikation)
- *type*: Mit der Anforderungskategorie *type* wird die Wichtigkeit bzw. Priorisierung der Anforderung erfasst (z. B. Festforderung, tolerierte Forderung, Wunschforderung)
- *group*: Mit der Anforderungskategorie *group* wird der Ursprung bzw. die Quelle der Anforderung erfasst, wie z. B. Markt, Gesetz, Norm, Nutzung
- *class*: Mit der Anforderungskategorie *class* wird die Bedingung bzw. Voraussetzung, unter der das Anforderungselement zu erfüllen ist, erfasst (z. B. Randbedingung, Schnittstelle zu anderen Systemen, funktionale Bedingung, nicht-funktionale Bedingung)
- *interval*: Mit *interval* wird bei quantifizierten Anforderungen der Soll- bzw. Zielwert oder das Soll- bzw. Zielwertintervall erfasst (z. B. untere und obere Intervallgrenze)

- *operator*: Mit *operator* wird bei quantifizierten Anforderungen die Interpretation bzw. Definition des Sollwerts bzw. Sollwertintervalls erfasst (s. Tabelle 1 für Beispiele)

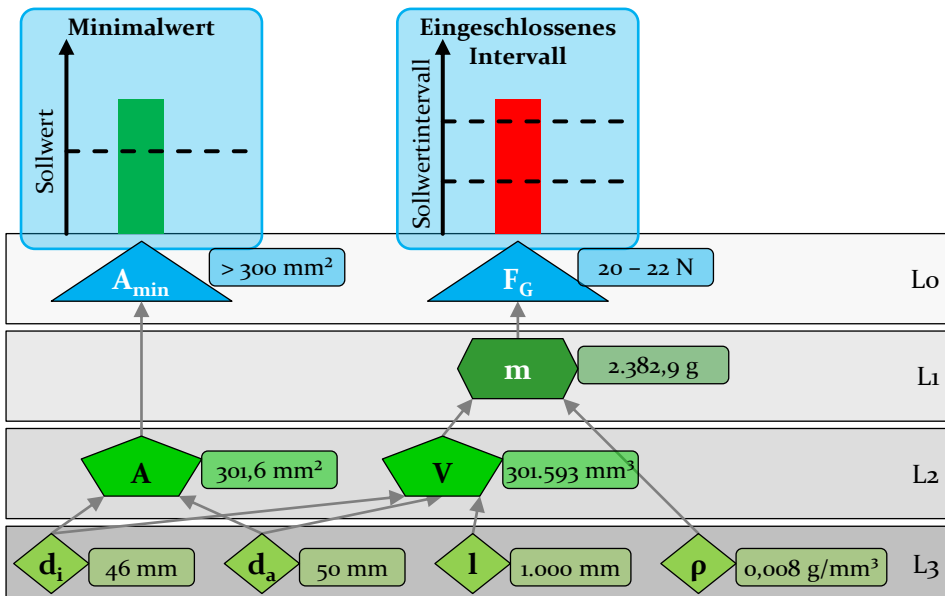
Tabelle 1: Interpretation von Kombinationen von Operator, Sollwert und Sollwertintervall (in Anlehnung an [S29])

Operator	Sollwert	Sollwertintervall
<	Maximalwert : Der zu vergleichende Wert darf nicht größer als der Sollwert sein	-
>	Minimalwert : Der zu vergleichende Wert darf nicht kleiner als der Sollwert sein	-
==	Exakter Wert : Der zu vergleichende Wert muss exakt mit dem Sollwert übereinstimmen	Eingeschlossenes Intervall : Der zu vergleichende Wert soll zwischen den beiden Intervallgrenzen liegen
!=	Ausgeschlossener Wert : Der zu vergleichende Wert darf jeden Wert annehmen, außer den Sollwert	Ausgeschlossenes Intervall : Der zu vergleichende Wert darf nicht zwischen den Intervallgrenzen liegen

Im Hinblick auf die Modellierung von **Relationen zwischen Anforderungselementen** sind die vier Arten komplementär, konfliktär, exkludierend und indifferent voneinander zu differenzieren. Während Elemente, die komplementäre Relationen haben, sich gegenseitig bzgl. der Zielwert-erfüllung unterstützen, behindern sich Elemente bzgl. der Zielwerterfüllung, die konfliktär zueinander in Beziehung stehen. Somit kann zwischen zwei konfliktären Anforderungselementen bspw. eine Priorisierung durch den Komplexitätsingenieur vorgenommen werden. Zwei Anforderungen, die eine exkludierende Relation haben, schließen sich reziprok aus, so dass sich der Komplexitätsingenieur für die Erfüllung einer von beiden entscheiden muss. Dagegen beeinflussen sich indifferente Anforderungen nicht.

Zudem ist die Differenzierung zwischen nicht quantifizierten und quantifizierten **Relationen zwischen Anforderungs- und Produktelementen** (insb. bei Eigenschaften, Merkmalen) hilfreich. Für eine bessere Unterscheidung werden quantifizierte Relationen bspw. mit *q* (engl. *quantified*) und farblich markiert. Bei quantifizierten Relationen ist sicherzustellen, dass die Einheit (*unit*) der Anforderung mit der bspw. in Beziehung stehenden Produkteigenschaft übereinstimmt. Ein Beispiel für eine quantifizierte Relation ist die Relation zwischen der Anforderung *Mindestquerschnittsfläche* von z. B. 300 in der Einheit mm^2 mit der Produkteigenschaft *Querschnittsfläche* von 301,6 in der Einheit mm^2 (s. in Kapitel 1.5 und Bild 64).

Auf Grundlage der quantifizierten Relationen zwischen Anforderungs- und Produktelementen (analog u. a. auch Prozess- und Organisationselemente) erfolgt die **multikriterielle Bewertung der Anforderungserfüllung durch Soll-Ist-Vergleiche**. Dazu wird in Abhängigkeit des definierten Operators und des Sollwerts bzw. Sollwertintervalls (s. Tabelle 1) für jede quantifizierte Relation ein Vergleich zwischen den diskreten Sollwerten bzw. Sollwertintervallen und den mit diesen in Relation stehenden Eigenschafts- oder Merkmalswerten automatisiert durchgeführt. Anschließend erfolgt die visuelle Darstellung der Anforderungserfüllung auf Basis dieser Soll-Ist-Vergleiche (s. Bild 64). Dabei besteht die Möglichkeit der Modifikation der Merkmalswerte mit Hilfe des Assistenzsystem, ohne diese in der Wissensbasis oder in den entsprechenden Modellen (z. B. CAD-Modellen) zu speichern. Dadurch kann analysiert und dargestellt werden, wie sich Merkmalsmodifikationen auf die in Relationen stehenden Eigenschaftswerte und somit auf die Erfüllung der jeweiligen Anforderungen auswirken. So wird mit Hilfe von Bild 64 deutlich, dass die Gewichtskraft $F_G = 23,4 \text{ N}$ (s. Anhang A.4) zur Erfüllung des Sollwertintervalls von 20 bis 22 N am besten durch Reduzierung von der *Länge* l und/oder der *Dichte* ρ realisiert werden kann, da diese A und damit $A_{\min}$ im Gegensatz zu d_i und d_a nicht beeinflussen. Dabei wird die *Gewichtskraft* F_G neben der *Masse* m auch von der ortsabhängigen *Schwerebeschleunigung* g (d. h. einer Umgebungsbedingung), welche in Bild 64 nicht dargestellt ist, beeinflusst.

Bild 64: Bewertung der Anforderungserfüllung mit dem Modul *Requirements Analysis*

Mittels des Moduls *Requirements Analysis* ist die **Auswahl von Elementen über Filter- als auch Suchfunktionalitäten** möglich (s. Bild 65). Durch die Eingabe eines Freitextes werden alle Elemente durchsucht und die Suchergebnisse als Liste dargestellt. Mittels der drei Anforderungskategorien *type*, *group* und *class* lassen sich die Anforderungselemente auch filtern.

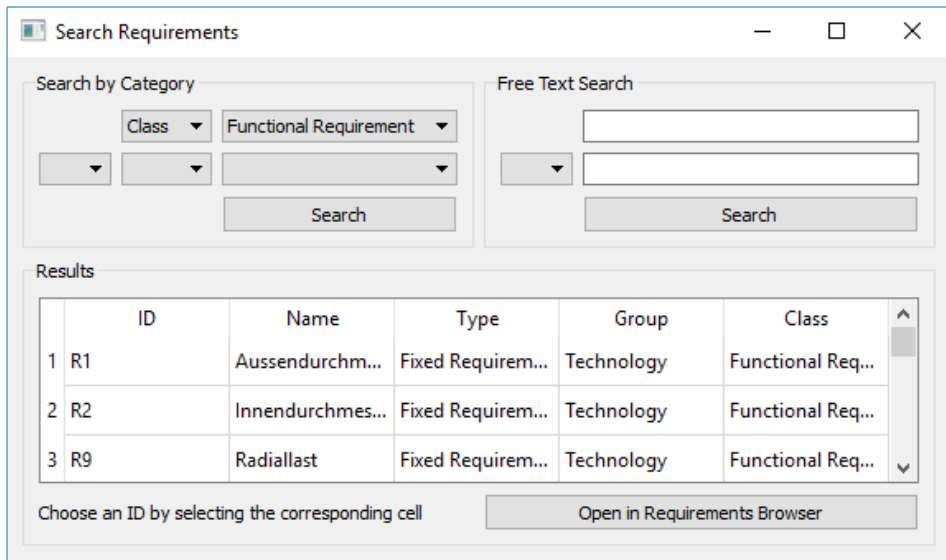


Bild 65: GUI des Moduls *Requirements Analysis* mit Filter- und Suchfunktionalitäten

5.5.3 Cost Analysis Module

Neben der Analyse von Anforderungen mit dem Assistenzsystem durch das Modul *Requirements Analysis* (s. Kapitel 5.5.2) ist auch die Analyse von Kosten im Rahmen des holistischen Komplexitätsmanagement wichtig, da rund zwei Drittel der Kosten in der PE festgelegt werden (s. Anhang A.25). Daher ist ein weiterer wissenschaftlicher Beitrag dieser Arbeit Konzepte zur Kostensenkung vorzuschlagen. Im Folgenden wird die auf Grundlage der Erkenntnisse des Autors das **Konzept des Moduls Cost Analysis** vorgestellt und erläutert. Das Modul *Cost Analysis* dient dem Komplexitätsingenieur zur Analyse und Bewertung von Kosten der diversen Elemente aus den Domänen Produkt, Prozess, Wissen und Organisation. Durch die Integration von Kosten in das Assistenzsystem wird die Kostentransparenz erhöht und Auswirkungen auf Kosten in Folge der Modifikation(en) von Systemelementen werden aufgezeigt. Somit unterstützt das Modul *Cost Analysis* bei der kostenorientierten Entscheidungsfindung und damit bei der Entwicklung gewinnbringender Produkte.

Zur **Integration der Kosten in die matrixbasierte Modellierung** für das holistische Komplexitätsmanagement hat der Autor dieser Arbeit einige, spezifische Konzepte für die diversen Domänen entwickelt. Darin werden grundsätzlich in Abhängigkeit der (geringsten) Kostenallokationsebene für eine bestimmte Zielsetzung die Elemente der relevanten Domänen mit den entsprechenden Kosten verknüpft (s. vereinfachtes, beispielhaftes Schema in Bild 66). So entsteht durch die Zuweisung von Kosten zu jedem Systemelement ein Kostenmodell für die verschiedenen Domänen. Dadurch ergeben sich die Kosten einer Domäne aus der Summe der jeweiligen Kosten der entsprechenden Elemente (z. B. in Bild 66 betragen die Gesamtkosten der Domäne Produkt bzw. DSM_{Produkt} gleich $170 = 50+40+80$).

Kostenmodell (vereinfachte Beispiele)				
	Produktbeispiel 1	Produktbeispiel 2	Anzahl	Kosten
DSM_{Produkt} 	Rohr X	Antriebswelle	1	50
	Rohr Y	Kugellager	4	40
	Rohr Z	Lagerhalbschale	8	80
DSM_{Prozess} 	3D-CAD-Modell modellieren		1	22
	Festigkeitsnachweise durchführen		1	84
	Angebote vom Lieferanten einholen		3	186 €
DSM_{Wissen} 	3D-CAD-Modell		1	5
	Bericht zum Festigkeitsnachweis		1	25
	2D-Zeichnung		1	2
DSM_{Organisation} 	Konstrukteur		2	180
	Berechnungsingenieur		1	110
	Einkäufer		1	80
			Gesamtkosten	864
			Target Costs	800

Bild 66: Stark vereinfachtes Schema zur Integration von Kosten in matrixbasierte Modelle

Dabei sind je nach verfolgter Zielsetzung und bestehenden Unternehmensvorgaben die **Kosten den Domänen und deren Elementen adäquat zuzuweisen** und zu allokalieren. So sind bspw. im Rahmen der Modellierung eines 3D-CAD-Modells für eine Antriebswelle durch einen Konstrukteur eine Vielzahl verschiedener Kostenpositionen potenziell von Interesse (z. B. Lizenzkosten, Lohnkosten, Schulungskosten, Materialstammkosten,

Verbrauchskosten), die zudem den relevanten Elementen bzw. Domänen (z. B. *Antriebswelle*, *3D-CAD-Modell modellieren*, *3D-CAD-Modell*, *Konstrukteur*) auf vielfältiger Art und Weise zugewiesen werden können. Demzufolge sind hierfür klare Vorgaben für eine eindeutige Zuordnung von Kosten zu Elementen und Domänen essenziell, da nur dadurch sichergestellt werden kann, dass sowohl alle Kosten erfasst als auch gemäß den Vorgaben den korrekten Kostenpositionen zugeordnet werden. Dies ist auch eine wesentliche Voraussetzung für den Vergleich der summierten Kosten mit den abgeleiteten Target Costs i. S. v. Target Costing (s. Anhang A.16).

Je nach unternehmensspezifischer Zuordnung von Kosten zu Elementen und damit Domänen und somit gesetzter Zielsetzung bzw. Fokus sind Kostenmodelle geeignet anzupassen. Im Folgenden wird **ein mögliches Kostenmodell konzeptuell am Beispiel der Wissensdomäne** vorgestellt. Weitere Anwendungsbeispiele zum *Cost Analysis Module* sind im Anhang A.25 zu finden (z. B. zur Prozess- und Funktionskostenrechnung). Hierzu sind Kostenmodelle für die anderen Domänen analog zu erstellen.

Die Integration von Kosten in die Wissensdomäne erfolgt generell durch **Zuordnung von Kosten zu Wissenselementen (WI-Elemente)**. Dazu werden die von LUFT in [P6] entwickelten sowie von LUFT & WARTZACK in [P8] konkretisierten und in [P9] kategorisierten *WI-Elemente* als Ergebnis einer Tätigkeit (*Prozess*) einer Person (*Organisation*) für die Entwicklung eines Einzelteils oder einer Baugruppe (*Produkt*) zur Erfüllung von Anforderungen (*Spezifikation*) aufgefasst. Für das Anwendungsbeispiel Rohr sind exemplarisch (explizite) WIs in Bild 67 zu finden, die in Form von Dateien bspw. in PDM-Systemen gespeichert werden.

Die Integration von Kosten in die matrixbasierten Modelle ist zwar einfach, aber die eindeutige **Zuordnung von Kosten zu den Elementen der diversen Domänen** ist in der Praxis häufig schwierig. Daher sind klare Richtlinien zu erstellen, um eine redundanzfreie und klare Kostenzuordnung zu unterstützen. So kann bspw. dem Prozess „*3D-Modell modellieren*“ die jeweils produktive Zeit mit dem jeweils zu verrechnenden Stundensatz eines Konstrukteurs, eines Zeichners und eines Wertanalytikers zugeordnet werden. Zum Wissenselement „*3D-CAD-Modell*“ lassen sich z. B. die Kosten für die Erstellung von Materialstammdaten und deren -pflege sowie Speicherplatzkosten zuordnen. Dem Organisationselement „*Konstrukteur*“ können die auf Prozesse (d. h. Tätigkeiten) nicht zu verrechnenden Kosten, wie bspw. Weiterbildungskosten und auch Personaleinstellungs- und -bindungskosten zugeordnet werden.

Nachfolgend wird auf Grundlage dieser Auffassung vereinfachend davon ausgegangen, dass das Kostenmodell für die Wissensdomäne sich aus den **einzelnen Elementkosten (EK)** zusammensetzt. Für die Ermittlung der EKs stehen dem Komplexitätsingenieur u. a. verschiedene kostenorientierte Modellierungsansätze (s. Anhang A.16) zur Verfügung. Dabei ist die Auswahl eines Ansatzes zur Abschätzung oder Berechnung der Kosten für (ein) WI-Element(e) im Rahmen der Erstellung eines Kostenmodelles u. a. von der Art des WI-Elements, der bestehenden Datengrundlage, der gewünschten Aufwendungen und der zu erreichenden Genauigkeit abhängig. Vereinfacht lassen sich die EK z. B. für ein WI-Element im Allgemeinen gemäß Gleichung (7) berechnen. Dabei entsprechen z. B. den *zeitabhängigen Kosten* das Gehalt eines Mitarbeitenden pro Stunde (z. B. für die Recherche eines expliziten WI-Elements, für das Erlernen eines impliziten WI-Elements) sowie den *zeitunabhängigen Kosten* die einmaligen Kosten für ein WI-Element (z. B. Erwerb von Dokument, Gebühr für eine Schulung).

$$EK = \text{Kosten}_{\text{zeitabhängig}} * \text{Zeiteinheit} + \text{Kosten}_{\text{zeitunabhängig}} \quad (7)$$

Anwendungsbeispiel Rohr		WI 1	WI 2	WI 3	WI 4	WI 5	WI 6	WI 7
WI 1	Kundenanforderungen			1				
WI 2	Gesetze und Normen			1				
WI 3	Spezifikation					1	1	
WI 4	Int. Konstruktionsrichtlinie					1		
WI 5	3D-CAD-Modell						1	1
WI 6	Festigkeitsbericht							1
WI 7	Fertigungszeichnung							

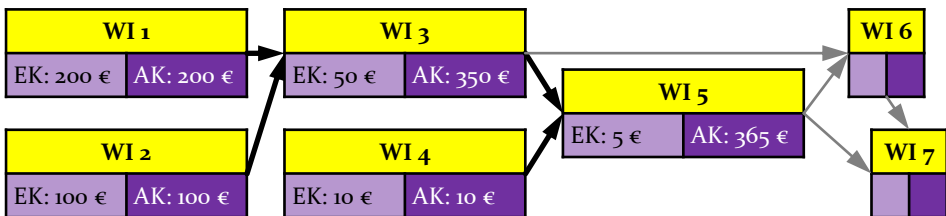


Bild 67: Vereinfachte, beispielhafte Darstellung eines Teilkonzepts für ein Kostenmodell mit Fokus auf die Wissensdomäne

Somit stellen die EKs die Kosten für das jeweilige, einzelne WI-Elemente dar (s. in Bild 67 z. B. Kundenanforderung, Gesetze und Normen, Spezifikation) dar. Die **akkumulierten Kosten (AK)** ergeben sich durch die Addition der EK eines bestimmten WI-Elements mit der Summe der kumulierten EK derjenigen WI-Elemente, die dieses WI-Element beeinflussen.

So berechnen sich zum Beispiel die AK in Höhe von 365 € von WI 5 in Bild 67 durch Addition der EK von 5 € des WI 5 selbst mit allen EK derjenigen WI-Elemente, die das WI 5 direkt als auch indirekt beeinflussen (z. B. WI 1 = 200 €, WI 2 = 100 €, WI 3 = 50 €, WI 4 = 10 €). Daher stellen die AK eines bestimmten WI-Elements die gesamten Kosten dar, die für die Generierung des WI-Elements anfallen. Damit ergeben sich die gesamten Kosten aller WI-Elemente der Wissensdomäne als Summe der EK dieser WI-Elemente.

Demnach kann das **Kostenmodell mit Fokus auf die Wissensdomäne** mit einem *dictionary*, welche mind. die folgenden *keys* enthält, beschrieben werden: *ID* des WI-Elements, *name* des WI-Elements, *Costs_{time-dependent}*, *time unit*, *Costs_{time-independent}*, *Costs_{accumulated}*, *Costs_{target}* sowie einer *list*, die alle IDs der direkt und indirekt beeinflussenden WI-Elemente umfasst. Das *dictionary* ist Teil der Wissensbasis des Assistenzsystems und kann bei Bedarf flexibel erweitert werden. Darüber hinaus können dem Anwender auch die Metadaten dieser WI-Elemente angezeigt werden (z. B. Ersteller, Datum der Erstellung und der Änderung, Dokumententyp und -art, Dateigröße).

Auf Grundlage der modellierten Relationen zwischen den WI-Elementen in der DSM_{Wissen} zusammen mit der Zuordnung von Kosten zu einzelnen WI-Elementen ergeben sich für den Komplexitätsingenieur viele **Nutzenpotentiale für ein kostenorientiertes Komplexitätsmanagement**. So lassen sich mittels der im Assistenzsystem verfügbaren Analyse- und Visualisierungsmethoden direkte und indirekte Relationen zwischen WI-Elementen (und auch Elemente weiterer Domänen) analysieren, visualisieren und bewerten (vgl. Bild 67). So ist bspw. in Bild 67 zu erkennen, dass das WI 5 zwar nur 5 € EK verursacht, aber zur Generierung von WI 5 aufgrund der dafür zuvor benötigten WI-Elemente 360 € erforderlich sind. Zudem wird besser nachvollziehbar und abschätzbar, welche Änderungskosten eine Modifikation eines bestimmten WI-Elements zur Folge haben kann.

Außerdem besteht auf Basis der Metadaten der WI-Elemente (z. B. CAD-Datei als ein Dokumenttyp) die Möglichkeit alle Dokumenttypen zu einer Kostenposition (z. B. für alle CAD-Dateien) zusammenzufassen. Dadurch kann der Komplexitätsingenieur je nach Zielsetzung entscheiden, welcher **Detaillierungsgrad in Bezug auf Kosten** bestimmter Dokumenttypen ein zweckdienliches bzw. optimales Aufwand-Nutzen-Verhältnis hat.

Aufgrund des signifikanten Einflusses der PE auf die Kostenfestlegung (s. Kapitel 1.1) ist das **Target Costing** (s. Anhang A.16) besonders relevant. Daher können im Assistenzsystem den WI-Elementen auch Zielkosten zugewiesen werden. Diese Zielkosten werden den entsprechend geplanten bzw. tatsächlichen Kosten auf allen Detaillierungsebenen gegenübergestellt.

Fazit zur mathematischen Analyse und Bewertung

In diesem Kapitel wurden die entwickelten Methoden zur Analyse und Bewertung matrixbasierter Modelle von komplexen Systemen erläutert. Im Analysemodul des Assistenzsystems sind zum einen individuelle, domänenspezifische Analysemethoden und zum anderen holistische, domänenübergreifende Analysen integriert. Zudem steht für die Analyse und multikriterielle Bewertung der Anforderungserfüllung von komplexen Systemen das Modul Requirements Analysis bereit. Das neu entwickelte Modul Cost Analysis dient zudem der Analyse und Bewertung der Zielkostenenerreichung. Im Vergleich zu bisherigen Analysemethoden, die zudem auch nicht oder nur ungenügend in Softwaresysteme implementiert wurden, lassen sich die in dieser Arbeit erforschten und entwickelten Methoden effizient und effektiv in der IPE einsetzen.

5.6 Holistische Manipulation und Modifikation

Eine weitere, wissenschaftliche Zielsetzung dieser Arbeit ist es, neben den Methoden der Modellierung, Visualisierung und Analyse von Modellen komplexer Systeme auch **mehrere Manipulationsmethoden** zu erforschen (s. Bild 48). Diese Manipulationsmethoden werden im Folgenden vorgestellt und unterstützen Entwickler gemäß dem in Bild 45 dargestellten Vorgehen in der Verbesserungsphase zur Modifikation, Veränderung und Verbesserung von Elementen und deren Relationen. Dazu dienen die zuvor in der Analysephase gewonnen Ergebnisse und Erkenntnisse. Allerdings können die v. a. für die Manipulation von matrixbasierten Modellen im Rahmen der Forschungsarbeiten des Autors dieser Arbeit entwickelten Methoden auch in allen Phasen des Vorgehens zum matrixbasierten Management komplexer Systeme (s. Bild 45) zielgerichtet eingesetzt werden.

Nachfolgend werden **zwei neu entwickelte Manipulationsmethoden** konzeptuell vorgestellt, die auch in das Assistenzsystem implementiert sind. Diese Manipulationsmethoden werden zwar anhand von Beispielen aus bestimmten Domänen dargestellt, sind aber analog für die Elemente und Relationen aller Domänen in ähnlicher Art und Weise anwendbar und bei Bedarf erweiterbar. In Kapitel 5.6.1 wird das *Modifying Module* zur grundsätzlichen Manipulation von Elementen und Relationen skizziert. Danach erfolgt die Darstellung des *Functional Analysis Module* in Kapitel 5.6.2 zur mathematischen bzw. funktionalen Modellierung von Relationen zwischen Elementen am Beispiel der Produktdomäne.

5.6.1 Modifying Module

Nach der automatisierten Modellierung von DSMs (s. Kapitel 5.2) und der computerunterstützten Modellierung von DMMs (s. Kapitel 5.3) kann aus unterschiedlichen Gründen eine **Manipulation von Elementen und deren Relationen** in der Wissensbasis des Assistenzsystems erforderlich sein. Mögliche Gründe sind bspw. Fehler oder Unvollständigkeiten bei der Erstellung der matrixbasierten Modelle sowie die gewünschte Modifikation von Elementen und Relationen in Folge von Erkenntnissen aus der Analyse. Hierfür steht im Menü der GUI des Assistenzsystems unter *Manipulation Tools* (vgl. Bild 49) u. a. das *Modifying Module* (s. Bild 68) zur Verfügung.

Zuerst sind die für die Manipulation relevanten Elemente durch Selektion der Domäne(n) zu wählen (z. B. Domäne *Organisation* in Bild 68). Danach werden diese Elemente aus der Wissensbasis des Assistenzsystems in das *Modifying Module* geladen und **im Bereich A als Ausgangselemente** bzw. *Source Nodes* gelistet. Anschließend kann der Komplexitätsingenieur mit dem Button *New Node* weitere Ausgangselemente erstellen sowie mit dem Button *Delete Node* selektierte Elemente entfernen. Mit dem Button *Modify Node* kann zudem das ausgewählte Element (z. B. *O21 Spezialist CRM*) mit Hilfe von Manipulationsfunktionalitäten modifiziert werden.

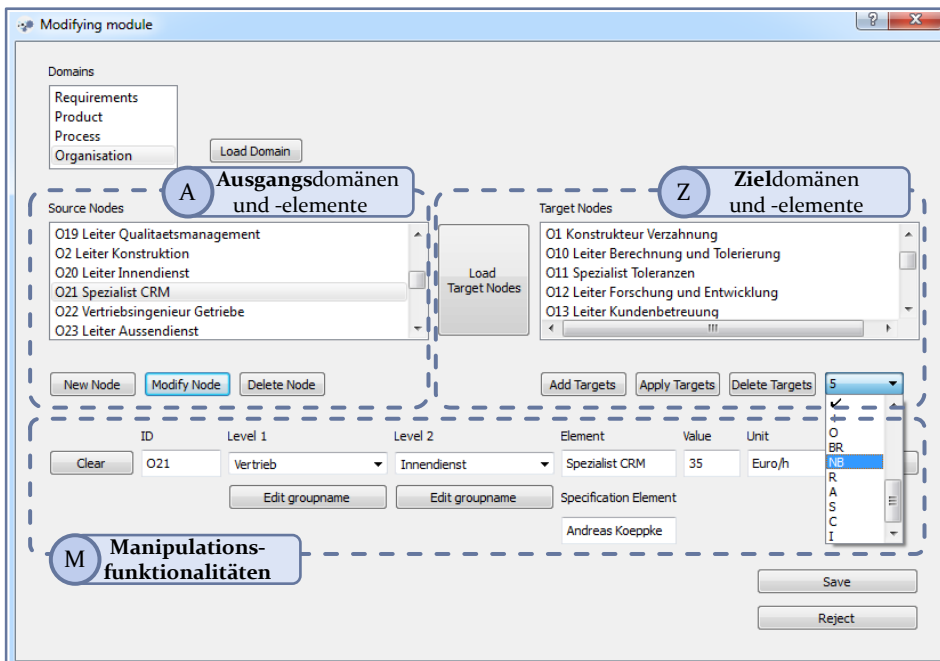


Bild 68: GUI des *Modifying Module* zur Manipulation von Elementen und Relationen

Die **Manipulation von Elementen erfolgt im Bereich M** (z. B. für das Element mit der ID *O21*). Grundsätzlich kann der Komplexitätsingenieur folgende Informationen bzw. Attribute von Elementen manipulieren (vgl. Bild 68): *ID* (Dubletten werden systemseitig vermieden), *Level 1*, *Level 2*, *Element* (i. S. v. Bezeichnung), *Value*, *Unit* und *Specification Element* (i. S. v. Spezifizierung bzw. Instanziierung). Dabei ist je nach Zugehörigkeit eines Elements zu einer Domäne unter den möglichen Eintragskategorien bzw. Einträgen etwas anderes zu verstehen (vgl. Tabelle 2 für beispielhafte, domänenspezifische Einträge). So kann der Komplexitätsingenieur einerseits bspw. für das Element mit der ID *O21* der Organisationsdomäne bei *Level 1* die Organisationseinheit *Vertrieb* manipulieren. Andererseits kann er bei *Level 2* die Abteilung oder Gruppe *Innendienst* des Elements mit der Bezeichnung (in diesem Fall i. S. v. Stelle) *Spezialist CRM*, die sich mittels des Eintrags unter *Specification Element* mit „Andreas Koeppke“ instanziiert (i. S. v. besetzt) werden kann, und dem *Value 35* und der *Unit Euro/h* manipulieren. Diese Manipulation kann entweder mit dem Button *Clear* verworfen oder mit dem Button *Accept* (hinter dem Drop-Down Menü) bestätigt werden. Bei Bedarf ist der Bereich M z. B. der Eintragskategorien und Manipulationsfunktionalitäten aufgrund des modularen Aufbaus des *Modifying Module* einfach und flexibel anpassbar.

Tabelle 2: Beispielhafte, domänenspezifische Einträge für Elemente im *Modifying Module*

Modifying Module	Produkt-domäne	Prozess-domäne	Organisations-domäne
<i>ID</i>	PID	SID	OID
<i>Level 1</i>	Modul, Baugruppe	Geschäftsprozess	Organisationseinheit
<i>Level 2</i>	Einzelteil	Phase	Abteilung, Gruppe
<i>Element</i>	Merkmal, Eigens.	Prozessschritt	Stelle
<i>Specific. Element</i>	Instanz. Merkmals	Instanz. Prozess	Personenname
<i>Value</i>	Wert (50)	Wert (1)	Wert (Stundensatz)
<i>Unit</i>	Einheit (mm)	Einheit (Stunde)	Einheit (Stundensatz)

Im Bereich Z ist das Verknüpfen von Ausgangs- mit Zielelement(en) zwar auch mit Hilfe des *Modifying Module* generell möglich, allerdings sollte die Modellierung von Relationen vorzugsweise mittels der in Kapitel 5.3 vorgestellten Methoden vorgenommen werden. Zur Modellierung von Relationen mit dem *Modifying Module* sind zuerst die Ausgangselemente

unter *Source Nodes* zu selektieren. Danach sind mittels des Buttons *Load Target Nodes* die bereits mit diesen in Relation stehenden Zielelemente aus der Wissensbasis unter *Target Nodes* zu laden (vgl. Bild 69). Dabei besteht bei Bedarf die Möglichkeit der Manipulation der Zielelemente in Bereich Z analog zu den Ausgangselementen in Bereich A. Anschließend erfolgt das Hinzufügen, Löschen oder Modifizieren von Relationen zwischen dem selektierten Ausgangselement und den auswählbaren Zielelementen. Dabei werden diejenigen Zielelemente grau hinterlegt, zu welchen das selektierte Ausgangselement bereits bestehende Relationen hat.

Ein **Beispiel zur Modellierung bzw. Manipulation von Relationen** ist in Bild 69 dargestellt. Dazu sind als Ausgangselemente Anforderungen (z. B. Element *R10 Drehzahlkennwert*) in Bereich A geladen worden. Dann sind mit dem Button *Load Target Nodes* alle Zielelemente aus den anderen, ausgewählten Domänen in den Bereich Z geladen worden. Unter der Annahme, dass das selektierte Ausgangselement *R10 Drehzahlkennwert* noch keine Relationen hatte, hat der Komplexitätsingenieur die drei Zielelemente *S9*, *S10* und *S11* der Domäne Prozess als Zielelemente ausgewählt und gemäß Bild 69 für die drei zu erstellenden Relationen je die Beziehungsart *X* gewählt. Dabei zeigt das *Modifying Module* je nach selektierten Domänen der Ausgangs- und Zielelemente kontextspezifisch nur die möglichen bzw. relevanten Arten von Relationen an. Während bspw. die RASCI-Notation für die Modellierung von Relationen zwischen Organisationselementen angezeigt wird (s. Bild 68), wird diese für die Modellierung von Relationen zwischen den Anforderungselementen und den drei Prozesselementen ausgeblendet (s. Bild 69). Mit dem Button *Apply Targets* werden die drei Relationen erstellt und abschließend mit dem Button *Save* (s. Bild 68) in den betroffenen dictionaries der Wissensbasis des Assistenzsystems konsistent gespeichert.

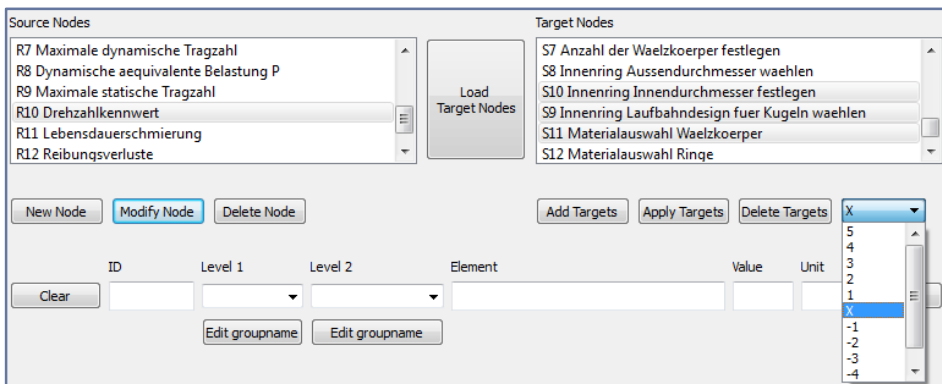


Bild 69: Manipulation von Relationen mit dem *Modifying Module*

5.6.2 Functional Analysis Module

Der **Aufruf des Moduls *Functional Analysis*** erfolgt im Menü der GUI des Assistenzsystems (s. Bild 49). Dabei wird dem Komplexitätsingenieur mittels einer Erklärung in einem Fenster empfohlen zuerst geschlossene Schleifen bzw. Kreise (*closed loops*, s. Kapitel 3.3.3) mit dem Analysemodul (s. Kapitel 5.5.1) zu identifizieren. Dies liegt daran, dass geschlossene Schleifen vor Nutzung des *Functional Analysis Module* nach Möglichkeit auflösen sind, um Endlosschleifen zu vermeiden. Danach öffnet sich die GUI des *Functional Analysis Module* mit Hilfe welcher der Komplexitätsingenieur v. a. funktionale, mathematische Relationen modellieren kann (s. Bild 70). Da i. d. R. häufig Elemente und Relationen zu manipulieren sind, besteht ein Link zum *Modifying Module*.

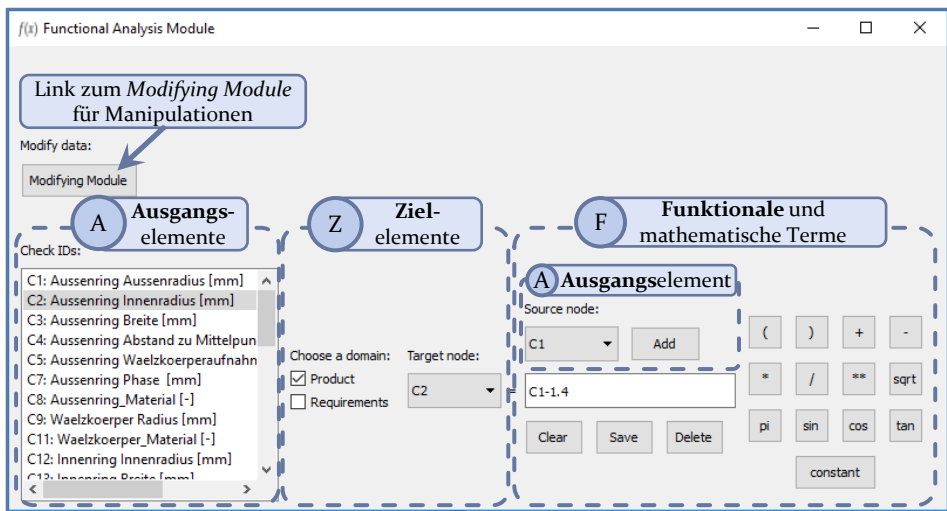


Bild 70: GUI des *Functional Analysis Module* zur Modellierung funktionaler Relationen

Dem Komplexitätsingenieur werden **im Bereich A alle möglichen Ausgangselemente** mit ihrer ID, der Bezeichnung sowie der Unit angezeigt (z. B. *C2: Aussenring Innenradius [mm]*). Da die Modellierung funktionaler und mathematischer Relationen insb. zwischen Elementen der Domänen Anforderung und Produkt Sinn ergibt, ist das *Functional Analysis Module* am besten für diese beiden Domänen geeignet.

Daher kann der Komplexitätsingenieur durch Selektion einer dieser beiden Domänen unter *Choose a domain* die Liste möglicher Zielelemente in der Drop-down-Liste unter *Target node* entsprechend einschränken (z. B. nur Produktelemente), um die **Auswahl des Zielelements im Bereich Z** zu erleichtern. So ist als zu beeinflussendes Zielelement *C2* in Bild 70 gewählt.

Das Zielelement ist auch das Element, in dessen dictionary der funktionale Term gespeichert wird und dessen Wert (*Value*) von diesem funktionalen Termin abhängig ist. **Im Bereich F wird der mathematisch-funktionale Term** (z. B. $C_1 - 1.4$) in das dafür vorgesehene Feld eingegeben. Allerdings ist dabei keine beliebige Eingabe (z. B. mathematischer Ausdrücke, Ziffern) möglich, um nicht berechenbare Terme zu vermeiden. Ausgangselemente, wie bspw. C_1 : *Aussenring Aussenradius [mm]*, werden in der Drop-down-Liste unter *Source node* selektiert und mit dem *Add*-Button dem Term hinzugefügt. Dabei werden in der Drop-down-Liste lediglich diejenigen IDs der Elemente angezeigt, die eine Relation zum Zielelement (*Target node* z. B. C_2) haben. Mit dem *Clear*-Button lassen sich bisherige Eingaben löschen. Dagegen werden mit dem Button *Delete* nicht nur die Eingaben, sondern auch der bereits gespeicherte Term des Zielelements gelöscht. Mit dem *Save*-Button wird der im Feld eingegebene Term im dictionary mit der ID des Zielelements gespeichert.

Zur **Modellierung eines mathematisch-funktionalen Terms** (z. B. $C_1 - 1.4$) für die Relation zwischen dem beeinflussten Zielelement und dem bzw. den beeinflussenden Ausgangselement(en) stehen im Bereich F mehrere mathematische Operatoren und Funktionen zur Verfügung (z. B. $**$ für die Potenz und *sqrt* für die Wurzel). Bei Bedarf können zusätzliche Funktionen und Operatoren ergänzt werden. Zudem kann der Komplexitätsingenieur mittels des *constant*-Buttons ein Dialogfenster öffnen, in dem dieser eine Dezimalzahl (z. B. 1.4) eingeben kann, wobei als Dezimaltrennzeichen ein Punkt zu verwenden ist. Unter der beispielhaften Annahme, dass von C_1 die zwei Elemente C_2 und C_{14} abhängig sind, wird bei einer Änderung des Werts von C_1 von 20.0 mm auf 18.0 mm in einem Fenster angezeigt, wie von dieser Änderung C_2 und C_{14} (d. h. deren jeweils neuer Wert) beeinflusst werden. Vor der Speicherung der Änderung ist dies vom Anwender zu bestätigen. Alle funktionalen Relationsarten werden in matrixbasierten Modellen mit dem Notationssymbol F (i. S. v. *function*) symbolisiert (z. B. im Matrix Module oder in .xlsx-Dateien).

Somit lässt sich abschließend festhalten, dass der Komplexitätsingenieur mit dem *Modifying Module* sowie dem *Functional Analysis Module* **zwei Manipulationsmodule zur Verfügung hat**, die bei der Modellierung und Manipulation von Elementen und Relationen matrixbasierter Modelle im Assistenzsystem kontextspezifisch unterstützen. Dabei lassen sich mit der mathematischen Modellierung funktionaler Terme „kaskadenartige“ bzw. verschachtelte Abhängigkeiten zwischen beliebig vielen Elementen modellieren (analog zu parametrisierten bzw. feature-basierten CAD-Modellen, die aus voneinander abhängigen Features bestehen können).

Fazit zur holistischen Manipulation und Modifikation

Die in diesem Kapitel vorgestellten Manipulationsmethoden dienen v. a. zur zielgerichteten Modifikation und Verbesserung von komplexen Systemen auf Basis der zuvor gewonnenen Ergebnisse und Erkenntnisse, können aber auch in allen Phasen des Vorgehens zum matrixbasierten Management komplexer Systeme angewendet werden. Dabei eignet sich das auf Grundlage der gewonnen wissenschaftlichen Erkenntnisse im Rahmen dieser Arbeit entwickelte Modifying Module zur generellen Manipulation von Elementen und Relationen besonders bei der Modellierung und der Verbesserung komplexer Systeme. Das implementierte Functional Analysis Module dient der mathematischen bzw. funktionalen Modellierung von Relationen zwischen Elementen, wodurch darauf aufbauende Analysemethoden verbessert und infolgedessen multikriterielle Bewertungen deutlich aussagekräftiger werden. Insgesamt bieten die neu entwickelten Manipulationsmethoden dem Komplexitätsingenieur somit eine kontextspezifische Unterstützung bei der Modellierung und Analyse.

6 Praktische Anwendungsbeispiele

„Wise executives tailor their approach to fit the complexity of the circumstances they face.”

DAVID J. SNOWDEN & MARY E. BOONE [514]

Nach SNOWDEN & BOONE passen kluge Entscheidungsträger ihren Ansatz an die Komplexität der Umstände an, mit denen sie konfrontiert sind [514]. Demgemäß wird folglich die im Kapitel 5 vorgestellte **Methodik mit dem adaptiven Assistenzsystem aus Sicht des Komplexitätsingenieurs an ausgewählte Anwendungsbeispiele angepasst** und angewendet. So ist u. a. die Anzahl und Art der Domänen mit ihren Elementen und Relationen sowie der zweckmäßige Detaillierungsgrad geeignet zu definieren. Dabei wird der Komplexitätsingenieur durch das Assistenzsystem unterstützt, das an folgenden Anwendungsbeispielen dargestellt wird.

Zudem dient die Implementierung der Teilsysteme und deren Integration (vgl. Bild 44) auch zur **Verifizierung der verschiedenen Methoden** bzw. des Gesamtkonzepts (vgl. [588]). Außerdem wird durch die nachfolgende praktische Anwendung des Assistenzsystems mit dessen Modulen in verschiedenen Praxisbeispielen die Erfüllung der Anforderungen aus Kapitel 4 sowie die Nützlichkeit demonstriert und damit die Validierung realisiert.

So wird in Kapitel 6.1 die Modellierung von DSMs am Beispiel der matrixbasierten Produktmodellierung eines Getriebes dargestellt. Dabei dient das **Getriebe auch als Anwendungsbeispiel** in den nachfolgenden Kapiteln. In Kapitel 6.2 steht die automatisierte, matrixbasierte Prozessmodellierung im Fokus. Mehrere Möglichkeiten zur computerunterstützten Modellierung von DMMs sind in Kapitel 6.3 skizziert. Anschließend werden die mit dem Assistenzsystem unterstützten Möglichkeiten zur Visualisierung in Kapitel 6.4 und zur Analyse in Kapitel 6.5 anhand Anwendungsbeispiele veranschaulicht.

Weitere Anwendungsbeispiele sind in den Publikationen des Autors der vorliegenden Arbeit zu finden (vgl. Literaturverzeichnis). So ist von LUFT & WARTZACK die matrixbasierte Produktmodellierung am Beispiel der Vorradaufhängung eines Fahrzeugs veranschaulicht [P13]. Zudem wird in LUFT ET AL. [P14] eine numerische MDM mit den Domänen Prozess, Wissen und Organisation für das industrielle Fallbeispiel einer Gasturbine modelliert und analysiert. Weitere Anwendungsbeispiele sind insb. in [P12], [P4], [P17] zu finden.

6.1 Matrixbasierte Produktmodellierung

Zur Demonstration der computerunterstützten Produktmodellierung mit Hilfe der automatisierten IT-Schnittstelle zwischen dem CAD-Modell und dem Assistenzsystem wird im Anwendungsbeispiel auf die in [S51] erstellte, **vereinfachte Variante des Demonstrator-Getriebes des Lehrstuhls für Konstruktionstechnik (KTmfk)** der FAU Erlangen-Nürnberg zurückgegriffen. Das CAD-Modell des verwendeten Getriebes umfasst die in Bild 71 dargestellten Baugruppen und Einzelteile. Dabei stellen die Antriebswelle, die Abtriebswelle und der Oberkasten eine Unterbaugruppen dar, werden aber zur besseren Übersicht in Bild 71 jeweils einzeln angeführt. Weitere Anwendungsbeispiele sind im Anhang A.22 zu finden.

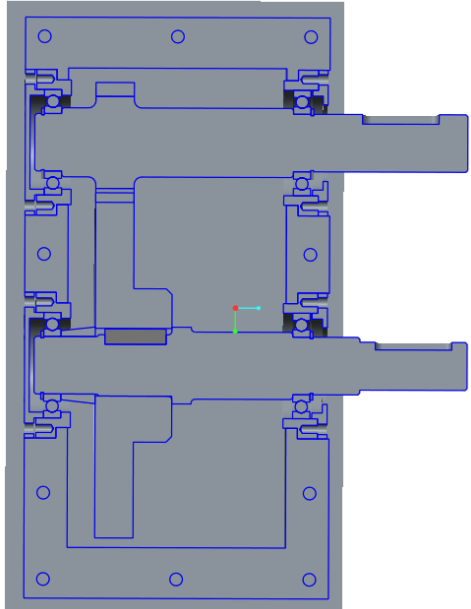
CAD-Modell des Getriebes im Schnitt	Bauelemente des Getriebes
	Bezeichnung (Anzahl)
	Unterkasten (1)
	Oberkasten (1)
	Antriebswelle (1)
	Abtriebswelle (1)
	Zahnrad groß (1)
	Zahnrad klein (1)
	
	Lagerhalbschale Loslager (4)
	Lagerhalbschale Festlager (4)
	Lagerdeckel geschlossen (2)
	Lagerdeckel offen (2)
	Rillenkugellager 16006 (4)
	Abstandshülse (1)
	Sicherungsring 30x1.5 (4)
	Passfeder 10x8x32 (1)

Bild 71: Vorstellung des verwendeten Getriebes (in Anlehnung an [S51])

Dabei sind die Parameter und Beziehungen des **Demonstrator-Getriebes im Hinblick auf die Anwendung der automatisierten IT-Schnittstelle etwas modifiziert**, um zum einen die Anzahl der Elemente und Relationen in der DSM zu reduzieren und zum anderen anschauliche Ergebnisse zu bekommen. So sind u. a. Schrauben, Scheiben, Muttern, Dichtungen und die zweite Getriebestufe entfernt und das Gehäuse folglich verkleinert. Zudem sind nur gleichartige Rillenkugellager vom Typ 16006 verbaut. Ferner ist das Gehäuse als Unter- und Oberkasten als ein Teil konstruiert.

6.1.1 Modellierung einer DSM für ein einfaches Einzelteil

Die Anwendung der IT-Schnittstelle für die automatisierte Modellierung der DSM erfolgt zuerst am **Beispiel der vereinfachten Lagerhalbschale** des Getriebes. Das CAD-Modell der verwendeten Lagerhalbschale ist in Bild 71 dargestellt und setzt sich auch aus einem zylindrischen Volumenkörper mit kreisförmig angeordneten Bohrungen zusammen.

Die automatisierte Modellierung der DSM-Lagerhalbschale erfolgt zuerst nur mit der Darstellung der direkten Relationen (d. h. ohne indirekte Relationen). Daher werden in Bild 72 nur die vom CAD-System Creo automatisch erstellten und die vom CAD-Konstrukteur modellierten direkten Beziehungen in der DSM dargestellt. Zum Vergleich erfolgt für das identische CAD-Modell die Darstellung der direkten und indirekten Relationen unmittelbar darunter in Bild 73. Darin ist zu erkennen, dass deutlich mehr Relationen bestehen und v. a. die Produkteigenschaft Masse „*P22 mass*“ aufgrund der direkten Beeinflussung des Volumens auch indirekt von vielen geometrischen Produktmerkmalen beeinflusst wird. Die DSM mit den direkten und indirekten Relationen ist im Vergleich zur DSM mit den nur direkten Relationen zwar weniger übersichtlich, aber deutlich aussagekräftiger. Für die **Repräsentation der beiden DSMs** wird die Darstellung mit Hilfe von MS Excel gewählt, weil dadurch die Lesbarkeit der DSMs in dieser Arbeit einfacher realisiert werden kann. In beiden Darstellungen werden mit einem „X“ die direkten, mit einem „Y“ die bedingten und mit Klammern „(...)“ die indirekten Relationen repräsentiert.

Das **Ergebnis der automatisierten Modellierung der DSM** zeigt, dass mit der entwickelten Methode und deren Implementierung alle Beziehungen im CAD-Modell automatisiert akquiriert und verarbeitet werden und in der DSM übersichtlich und umfassend dargestellt werden. Da nur für den Parameter „*Innendurchmesser*“, der auf KE-Ebene definiert wurde, das zugeordnete KE und damit der Pfad akquiriert werden kann, bleibt die Spalte „*Path*“ für dieses einzelne Einzelteil weitgehend leer. Allerdings sind die von Creo automatisch erzeugten Bemaßungen (z. B. von d_0 bis d_{33}) aufgrund deren nicht aussagekräftiger Bezeichnungen für darauffolgende Analyse und Visualisierungen nur bedingt geeignet. Daher kann die Interpretation der DSM v. a. bei komplexeren Produkten nur zusammen mit dem CAD-Modell erfolgen, da in diesem durch die visuelle Darstellung der einzelnen Maßangaben diese vom Komplexitätsingenieur erkannt werden. Eine Lösung dieser Problematik ist die Verwendung von aussagekräftigen Bezeichnungen von Maßangaben, was jedoch i. d. R. zu einem manuellen (Mehr-)Aufwand bei der Erstellung des CAD-Modells führt.

lagerhalbschale.prt					do	di	d2	d3	d4	d5	d6	d7	d8	d9	d12	d13	d26	d27	d28	d29	d30	p31	p32	p33	ptc_material_nam	innendurchmess	mass	volume
Read: row affects column					C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	P22	P23
Nr.	Path	me in C	Value	Unit																								
C0	d0	0	mm		X																							X
C1	d1	0	mm			X																						X
C2	d2	0	mm				X																					X
C3	d3	180	°					X																				X
C4	d4	24,5	mm						X		X																	X
C5	d5	27,5	mm							X				X														X
C6	d6	32,5	mm							X				X														X
C7	d7	38	mm								X																	X
C8	d8	13	mm									X		Y														X
C9	d9	18	mm												X													X
C10	d12	33	mm													X												X
C11	d13	0	mm														X											X
C12	d26	4,2	mm															X										X
C13	d27	118	°																X									X
C14	d28	8	mm																	X								X
C15	d29	60	mm																		X							X
C16	d30	5,79	mm																			X						X
C17	p31	4	unknown																				X					X
C18	p32	1	unknown																					X				X
C19	d33	7,5	mm																						X			X
C20	ptc_ma	STEEL	-																								X	
C21	protr	innend	27,5	-						X																		X
P22	mass	0,1239	kg																									X
P23	volume	15753	mm^3																									X

Bild 72: Automatisierte Modellierung der DSM mit nur direkten Relationen

lagerhalbschale.prt					do	di	d2	d3	d4	d5	d6	d7	d8	d9	d12	d13	d26	d27	d28	d29	d30	p31	p32	p33	ptc_material_name	innendurchmesser	mass	volume
Read: row affects column					C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	P22	P23
Nr.	Path	me in C	Value	Unit																								
C0	d0	0	mm		X																						(X)	X
C1	d1	0	mm			X																					(X)	X
C2	d2	0	mm				X																				(X)	X
C3	d3	180	°					X																			(X)	X
C4	d4	24,5	mm						X	X			(X)	X													(X)	X
C5	d5	27,5	mm							X				X													(X)	X
C6	d6	32,5	mm						X		X			(X)	X												(X)	X
C7	d7	38	mm						(X)	X				(X)													(X)	X
C8	d8	13	mm									X		Y												(X)	(Y)	X
C9	d9	18	mm												Y												(X)	X
C10	d12	33	mm													X											(X)	X
C11	d13	0	mm														X										(X)	X
C12	d26	4,2	mm															X									(X)	X
C13	d27	118	°																X								(X)	X
C14	d28	8	mm																	X							(X)	X
C15	d29	60	mm																		X						(X)	X
C16	d30	5,79	mm																			X					(X)	X
C17	p31	4	unknown																				X				(X)	X
C18	p32	1	unknown																					X			(X)	X
C19	d33	7,5	mm																						X		(X)	X
C20	ptc_material	STEEL	-																							X		
C21	protr_innend	27,5	-							X				(X)													(X)	(X)
P22	mass	0,1239	kg																									
P23	volume	15753	mm^3																									X

Bild 73: Automatisierte Modellierung der DSM mit direkten und indirekten Relationen

6.1.2 Modellierung einer DSM für eine komplexe Baugruppe

Nachdem die Modellierung einer DSM für ein exemplarisches Einzelteil in Kapitel 6.1.1 dargestellt wurde, folgt ein Beispiel für eine komplexe Baugruppe. Schließlich wird mit Hilfe der Methodik und des Assistenzsystems das Ziel verfolgt, bei der Modellierung komplexer Systeme, wie z. B. dem vorgestellten Demonstrator-Getriebe, zu unterstützen. Da das Getriebe aus rund 900 Parametern besteht (v. a. Produktmerkmale und -eigenschaften), kann dieses in dieser Arbeit nicht abgebildet werden. Daher wird in Bild 75 nur die **DSM_{Constraints} der Baugruppe Getriebe** dargestellt, wobei indirekte Relationen zur besseren Übersichtlichkeit ausgeblendet sind.

Dabei enthält die *DSM_{Constraints}* nur **Relationen zwischen allen Baugruppen und Einzelteilen des Getriebes** (s. Kapitel 5.2.1). Dementsprechend sind in Bild 75 alle in Bild 71 angeführten Bauelemente des Getriebes zu finden. Außerdem werden durch die Formatierung mittels Rahmen und der grauen Einfärbung der Baugruppenelemente die drei Baugruppen „*antriebswelle_montiert.asm*“, „*abtriebswelle_montiert.asm*“ und „*oberkasten_montiert.asm*“ ersichtlich, wobei die beiden Wellenbaugruppen jeweils die zwei weiteren Unterbaugruppen „*rillenkugellager_16006.asm*“ umfassen. Zudem sind alle Relationen der automatisiert modellierten DSM oberhalb der Diagonalen zu finden. Dies liegt daran, dass die Bauelemente entsprechend ihrer Montagereihenfolge vom Konstrukteur im CAD-Modell modelliert bzw. bzgl. dieser im CAD-Modell zusammengefügt wurden und demgemäß auch vom Assistenzsystem in die matrixbasierte Produktmodellierung des DSM Getriebe transformiert wurden. Hierfür dient die *DSM_{Constraints}* als ein beispielhafter Ausschnitt dieser rund 900 Parameter umfassenden DSM. In dieser *DSM_{Constraints}* sind die in Bild 74 erklärten Abkürzungen für Relationen und Bauelemente verwendet. Dabei sei angemerkt, dass – wie in Kapitel 5.2.1 erläutert – prinzipiell alle Beziehungen im CAD-Modell in die DSM transformiert und in dieser repräsentiert werden können.

Abk. für Relationen in der DSM			Abk. für Bauelemente der DSM		
Abk.	Englisch	Deutsch	Abk.	Englisch	Deutsch
A	align	ausgerichtet	.prt	part	Einzelteil
Ao	align_off	ausgerichteter Offset	.asm	assembly	Baugruppe
I	insert	eingefügt	.asm	subassembly	Unterbaugruppe
M	mate	zusammenfallend			
O	orient	orientiert			

Bild 74: Verwendete Relationen und Bauelemente in der *DSM_{Constraints}* für das Getriebe

lehrstuhlgetriebe.asm		Read: row affects column		Name in Cyo		Nr.:	
pr10	unterkasten.prt						
pr11	lagerhalbschale.prt						
pr12	lagerhalbschale.prt						
pr13	lagerhalbschale.loslager.prt						
pr14	lagerhalbschale.loslager.prt						
asm5	antriebswelle_montiert.asm						
pr6	antriebswelle.prt						
pr7	108832_<din6889>.prt						
pr8	zahnrad.prt						
pr9	hueise.prt						
asm10	rollenkugellager_6006.asm						
pr11	wälzkörper.prt						
pr12	innenring.prt						
pr13	ausserenring.prt						
pr14	d471_30x_5din471.prt						
asm5	rollenkugellager_6006.asm						
pr6	wälzkörper.prt						
pr7	innenring.prt						
pr8	ausserenring.prt						
pr9	d471_30x_5din471.prt						
asm20	abtriebswelle_montiert.asm						
pr21	abtriebswelle.prt						
pr22	zahnrad_klein.prt						
asm23	rollenkugellager_6006.asm						
pr24	wälzkörper.prt						
pr25	innenring.prt						
pr26	ausserenring.prt						
pr27	d471_30x_5din471.prt						
asm28	rollenkugellager_6006.asm						
pr29	wälzkörper.prt						
pr30	innenring.prt						
pr31	ausserenring.prt						
pr32	d471_30x_5din471.prt						
asm33	oberkasten_montiert.asm						
pr34	oberkasten.prt						
pr35	lagerhalbschale.prt						
pr36	lagerhalbschale.prt						
pr37	lagerhalbschale.loslager.prt						
pr38	lagerhalbschale.loslager.prt						
pr39	lagerdeckel.prt						
pr40	lagerdeckel.prt						
pr41	lagerdeckel_offen2.prt						
pr42	lagerdeckel_offen3.prt						
pr1	lagerdeckel_offen1.prt						
pr141	lagerdeckel_offen2.prt						
pr142	lagerdeckel_offen3.prt						

Bild 75: Automatisierte Modellierung der DSM für eine komplexe Baugruppe

6.2 Matrixbasierte Prozessmodellierung

Zur Veranschaulichung der automatisierten IT-Schnittstelle zwischen dem BPMN-Modell und dem Assistenzsystem für die computerunterstützte, matrixbasierte Prozessmodellierung wird als Anwendungsbeispiel zuerst in Kapitel 6.2.1 ein **vereinfachter PEP mit ausgewählten Konstruktions- und Fertigungsaktivitäten für eine Welle** verwendet. Dadurch wird die grundsätzliche Methode sowie deren Implementierung demonstriert.

Darauf aufbauend wird in Kapitel 6.2.2 ein **komplexeres BPMN-Modell des PEP für die verwendete, vereinfachte Getriebevariante** aus Kapitel 6.1 vorgestellt. Zudem wird das Ergebnis der automatisierten Transformation des BPMN-Prozessmodells in die matrixbasierte Prozessmodellierung des Assistenzsystems erläutert.

6.2.1 Modellierung einer DSM für einen einfachen Prozess

Für die bessere Veranschaulichung der automatisierten, matrixbasierten Prozessmodellierung dient der in Bild 76 dargestellte, **sehr einfache BPMN-Prozess aus ADONIS**. In diesem Prozess mit sechs Aktivitäten sind alle drei Gateways *AND*, *OR* und *XOR* sowie die entsprechenden *types* des verbindenden Objektes Sequenzfluss der BPMN in Textdarstellung zu finden. Dabei können u. a. einzelne Flussobjekte und verbindende Objekte in ADONIS als auch im Assistenzsystem z. B. durch Gestalt und Farbe bei Bedarf vom Anwender optional formatiert und somit differenziert werden.

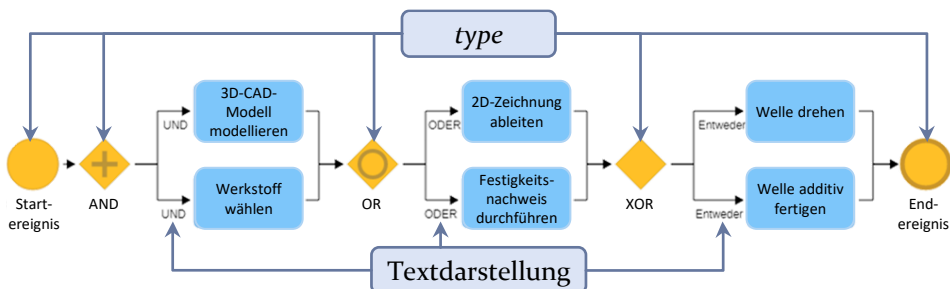


Bild 76: Einfaches BPMN-Prozessmodell des Getriebes mit drei Gateways

Dieser einfache BPMN-Prozess wird – wie in Kapitel 5.2.2 erläutert – in ein matrixbasiertes Prozessmodell durch das Assistenzsystem transformiert. Das Resultat dieser automatisierten Akquisition und Verarbeitung des BPMN-Modells kann z. B. mit Hilfe **des matrixbasierten Modellierungsmoduls des Assistenzsystems dargestellt** werden (s. Bild 77). Dabei kann der Komplexitätsingenieur mit diesem Modul die einzelnen Elemente auch beliebig anordnen und modifizieren (s. für Beispiele in Bild 77).

ID	Bezeichnung	Wert	Einheit	G0	A0	A1	G1	G2	A2	A3	A4	A5	A6	A7
G0	AND								+			+		
A0	Startereignis			x										
A1	Welle additiv f.													x
G1	OR									o	o			
G2	XOR					x							✓	
A2	3D-CAD-Modell						x							
A3	Festigkeitsn.							x						
A4	2D-Zeichnung							x						
A5	Werkstoff wäh.						x							
A6	Welle drehen													x
A7	Endereignis													

Bild 77: Automatisiert erstellte DSM eines einfachen BPMN-Prozessmodells des Getriebes

In Bild 77 sind die **drei Gateways jeweils mit Blau umrahmt**. Es ist zu erkennen, dass nach dem Auslösen des Startereignisses *A0* die Aktivitäten *A2* und *A5* aufgrund des *AND*-Gateway *G0* parallel zu bearbeitet sind und im *OR*-Gateway *G1* enden. In Folge des *OR*-Operators können die beiden Aktivitäten *A3* und/oder *A4* bearbeitet werden, welche in *G2* enden. Da im Gegensatz zur Aktivität *A1* (mit „X“ gekennzeichnet) die Aktivität *A6* (mit Häkchen „✓“ symbolisiert) die *XOR*- Bedingung erfüllt ist, wird *A6* ausgeführt (vgl. Kapitel 5.2.2). Die Aktivitäten *A1* und *A6* führen zum Endereignis *A7* (vgl. letzte Spalten in Bild 77).

Die im Assistenzsystem modifizierten Elemente und Relationen können sowohl in der Wissensbasis des Assistenzsystems als auch als *.bpmn*-Datei gespeichert werden. Diese kann anschließend bspw. in ADONIS wieder als BPMN-Prozessmodell dargestellt (s. in Bild 76) und auch in dieser Prozessmodellierungssoftware weiterverarbeitet werden. Somit ist die Forderung einer **bidirektionalen IT-Schnittstelle** zwischen domänenspezifischen Anwendungssystemen und dem Assistenzsystem erfüllt (vgl. Kapitel 4.2.2).

6.2.2 Modellierung einer DSM für einen komplexen Prozess

Nachfolgend wird die Methode zur automatisierten Modellierung der DSM am Beispiel eines komplexen BPMN-Prozessmodells des Getriebes gezeigt, wovon in Bild 78 ein Ausschnitt vereinfacht dargestellt ist. Da die generelle Funktionsweise der implementierten Methode bereits in Kapitel 6.2.1 an einem Beispiel veranschaulicht wird, soll in Bild 78 nur die **Struktur eines komplexen BPMN-Prozessmodells** abgebildet werden, so dass dessen Lesbarkeit sekundär ist. Die folgende Abbildung stellt einen Ausschnitt aus einem umfangreichen PEP des in Kapitel 6.1 vorgestellten Getriebes dar.

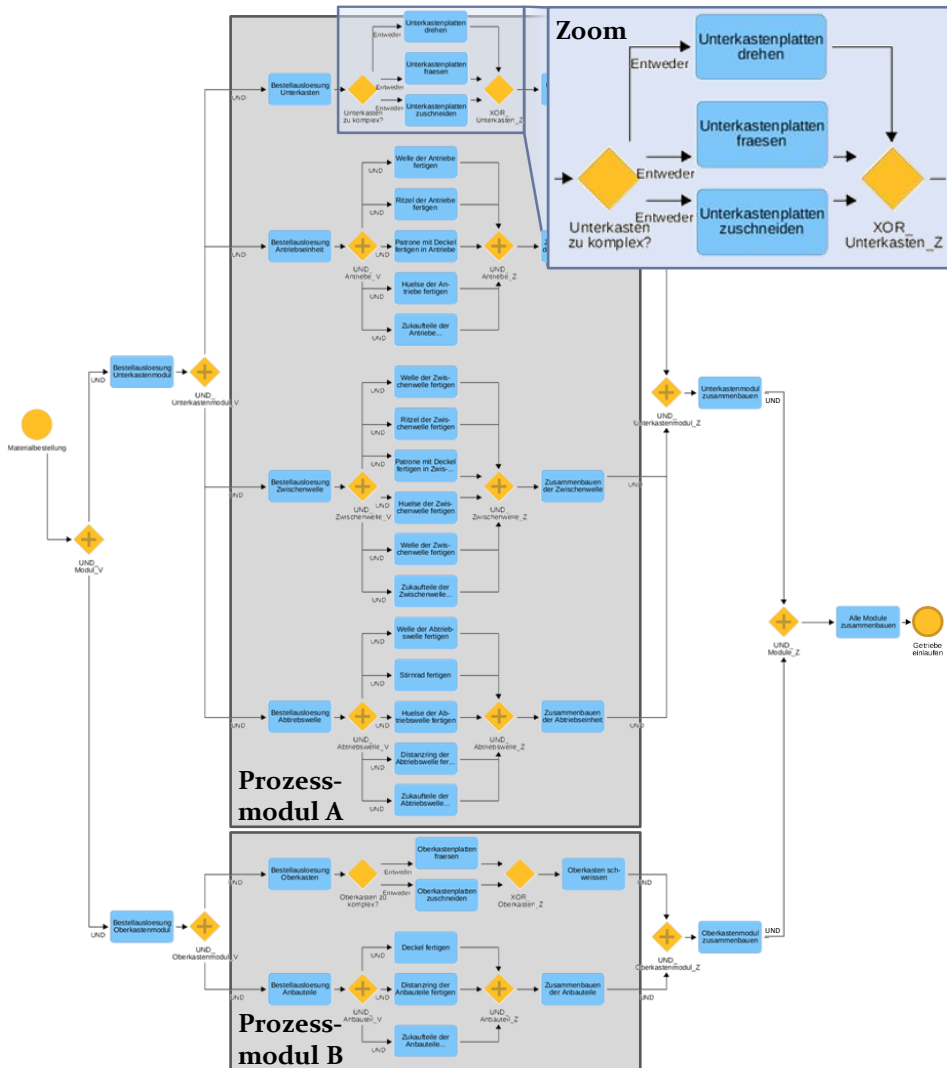


Bild 78: Komplexeres BPMN-Prozessmodell des Getriebes (in Anlehnung an [S61])

Das in ADONIS erstellte BPMN-Prozessmodell in Bild 78 wird als *.bpmn*-Datei exportiert und in die Wissensbasis des Assistenzsystems (s. DSM-Proc. in Bild 48) mit Hilfe der implementierten Methode in Kapitel 5.2.2 importiert (s. in Bild 49 Menü „Convert Tools“). Dadurch kann die **automatisiert erstellte DSM-Prozess des Getriebes** in einem der Module des Assistenzsystems visualisiert, analysiert, manipuliert und auch mit anderen DSMs verknüpft werden (s. Bild 48). So wird z. B. die DSM-Prozess im matrixbasierten Modellierungsmoduls des Assistenzsystems in Bild 79 dargestellt (mögliche Option im Menü unter „Manipulation Tools“ s. in Bild 49).

6.3 Modellierung von DMMs

Nach der Modellierung der DSM folgt nun deren Verknüpfung mit DMMs. Zur Illustration der in Kapitel 5.3 vorgestellten **Methoden zur computerstützten Modellierung von DMMs** dienen folgende Kapitel. In Kapitel 6.3.1 werden die grafische Modellierung und in Kapitel 6.3.2 Möglichkeiten des Moduls Domain Mergers zur Modellierung der DMM vorgestellt.

6.3.1 Grafische Modellierung einer DMM

Die automatische und grafische Modellierung von DMMs wird am Beispiel des in Kapitel 6.1 vorgestellten Getriebes gezeigt. Dazu wird in der ersten Phase des computerunterstützten Vorgehens zur Modellierung von DMMs (s. Bild 58) die $DSM_{Produkt}$ (s. Kapitel 6.1) und die DSM_{Wissen} des Getriebes gewählt und die leere $DMM_{Produkt-Wissen}$ vom Assistenzsystem erzeugt. In Phase zwei werden **mittels des Algorithmus automatisch potenzielle Relationen zwischen Elementen der DMM** erkannt und gesetzt. Danach wird die DMM in yEd importiert. Die $DMM_{Produkt-Wissen}$ wird in der vierten Phase in yEd je nach dem gewählten Layout als Graph visualisiert. Das Resultat der automatisch modellierten Relationen ist in Bild 80 mittels des Radialen Baum-Layouts gezeigt, wobei zur besseren Lesbarkeit die Produktelemente in Grün und die Wissens Elemente in Blau eingefärbt sind.

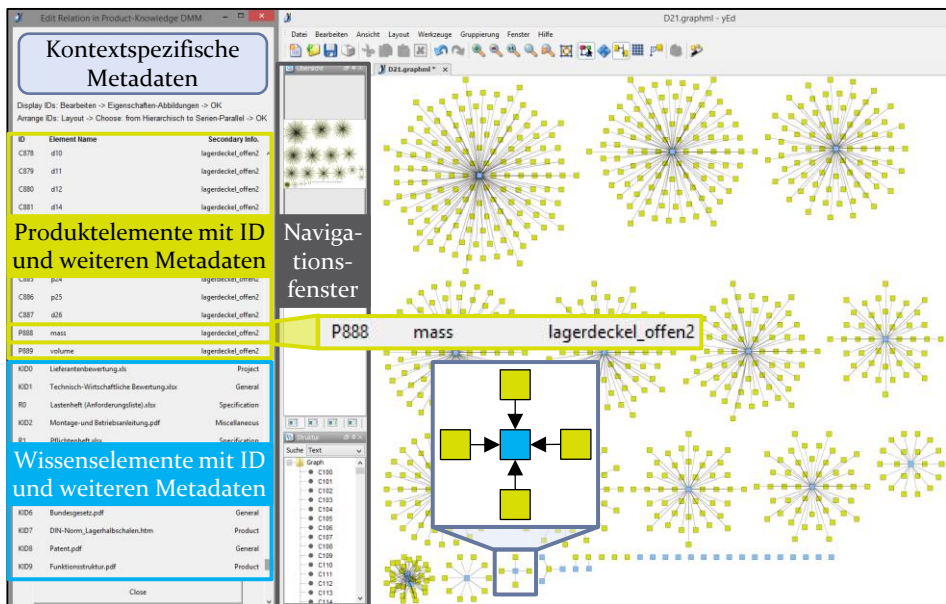


Bild 80: Darstellung der automatisch erstellten Relationen der $DMM_{Produkt-Wissen}$ in yEd (in Anlehnung an [S13])

Die $DMM_{\text{Produkt-Wissen}}$ des Anwendungsbeispiels umfasst 930 Elemente und somit rechnerisch maximal 863.970 mögliche Relationen, wovon in der Praxis i. d. R. nur ein kleiner Bruchteil existiert. Der in Kapitel 5.3.1 vorgestellte Suchalgorithmus hat in diesem Anwendungsbeispiel genau 1.422 Relationen gesetzt. In Bild 80 ist aufgrund der Vielzahl an Elementen nur ein Ausschnitt der $DMM_{\text{Produkt-Wissen}}$ im Radialen Baum-Layout (s. Kapitel 3.3.2) dargestellt. Dadurch kann der Komplexitätsingenieur einfach und schnell erkennen, **welche einzelnen Produktelemente ein bestimmtes Wissensselement beeinflussen**. So ist z. B. das Wissensselement „*CAD-Datei des oberkasten.prt*“ (nicht zu verwechseln mit dem Produktelement „*oberkasten.prt*“ i. S. v. geometrisches Einzelteil) von den vier Produktelementen „*Länge*“, „*Breite*“, „*Höhe*“ und „*Material*“ abhängig (s. vergrößertes, kreisförmiges Cluster in Bild 80). Je größer daher der Clusterkreis ist, desto mehr Produktelemente beeinflussen das im Kreismittelpunkt angeordnete Wissensselement. Dagegen hat der Algorithmus für 16 Wissensselemente (s. in Bild 80 unten) keine Relationen zu Produktelementen identifiziert. Bei diesen Wissensselementen handelt es sich bspw. um eine .xlsx-Datei (z. B. *Technisch-Wirtschaftliche Bewertung.xlsx*) und eine .pdf-Datei (z. B. *Patent.pdf*). Dies liegt daran, dass zwischen den Metadaten der Produktelemente und den Metadaten dieser 16 Wissensselemente keine hinreichende Übereinstimmung vom Algorithmus erkannt wurde.

Nach dem automatischen Setzen von Relationen durch den Algorithmus ist es die Aufgabe des Komplexitätsingenieurs diese bspw. mit Hilfe der Funktionalitäten von yEd zu überprüfen und bei Bedarf weitere Relationen manuell zu modellieren. Für die **Modellierung und Manipulation von Relationen in yEd** stehen verschiedene Layouts zur kontextspezifischen Unterstützung zur Verfügung. Für die Navigation durch komplexe Systeme stehen zudem einige, übersichtliche Optionen im Navigationsfenster bereit (s. Bild 80). Dabei werden in Abhängigkeit der Navigation automatisch und kontextspezifisch die relevanten Metadaten zu den gerade jeweils im Fokus stehenden Elementen angezeigt (s. Bild 80 linkes Fenster). Dies ist möglich, da aus dem Assistenzsystem zu allen Elementen und Relationen auch die jeweils relevanten Metadaten in yEd übergeben werden.

Dadurch wird der **Komplexitätsingenieur bei der Modellierung von DMMs signifikant unterstützt**. So führt bspw. das automatische Setzen der 1.422 Relationen im angeführten Anwendungsbeispiel zu einer Zeiteinsparung von rund acht Stunden unter der Annahme, dass das Suchen des Ausgangs- und Zielelements jeweils 10 Sekunden dauert. Zudem wird der Komplexitätsingenieur durch die grafische Modellierung der Relationen und die verschiedenen Layouts bestmöglich unterstützt.

6.3.2 Modellierung mit dem Domain Merger

Anhand des in Kapitel 6.1 vorgestellten Getriebes werden zwei weitere Anwendungsbeispiele zur Modellierung mit dem Domain Merger dargestellt. So ist in Bild 81 bspw. die Modellierung von zwei Relationen der Art „X“ auf einmal zwischen dem Ausgangselement „R5: R_Innenring Breite“ auf der linken Seite und jeweils den beiden Produktmerkmalen „C1: Aussenring Aussenradius“ und „C2: Innenring Aussenradius“ auf der rechten Seite gezeigt. Stattdessen oder zusätzlich kann z. B. auch mittels einer „X“-Relation modelliert werden, dass die Anforderung „R5: R_Innenring Breite“ den Prozessschritt „S9: Innenring Aussendurchmesser definieren“ beeinflusst (s. Bild 82). Der Komplexitätsingenieur kann bei Bedarf mittels des Buttons „clear“ die ausgewählten Elemente jederzeit entfernen.

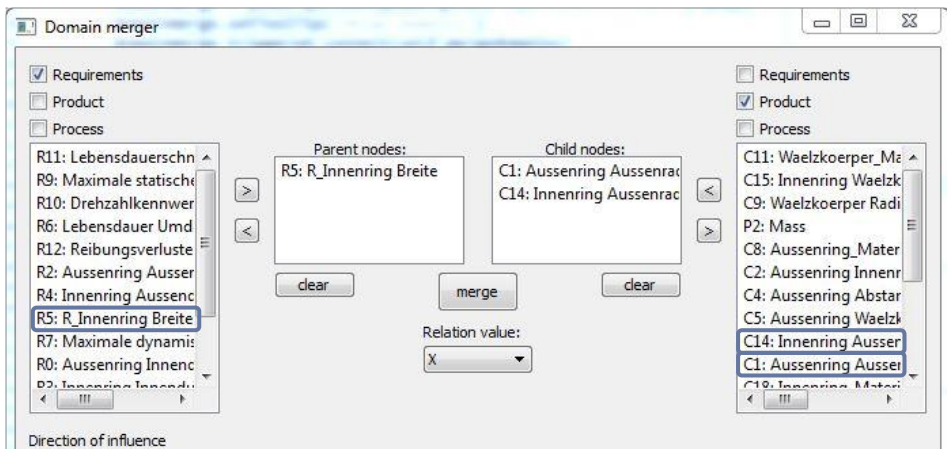


Bild 81: GUI des Moduls Domain Merger zur Modellierung der DMM_{Requirements-Product}

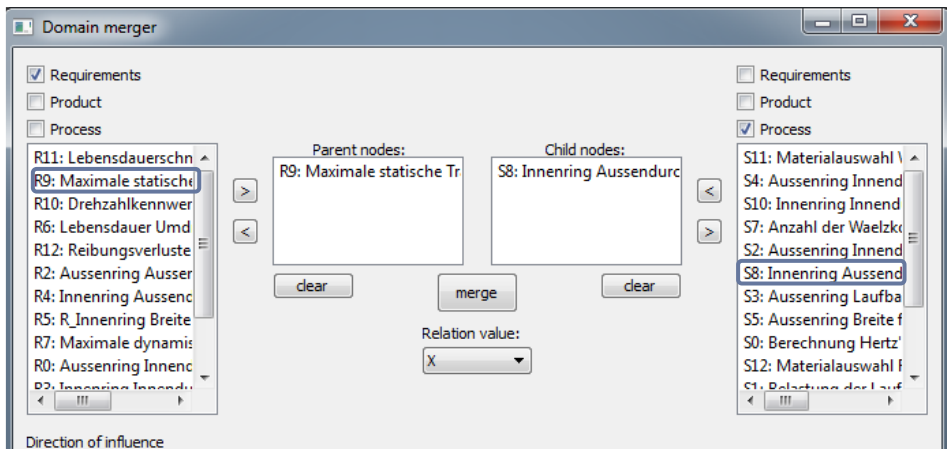


Bild 82: GUI des Moduls Domain Merger zur Modellierung der DMM_{Requirements-Process}

6.4 Visualisierung von matrixbasierten Modellen

In diesem Kapitel erfolgt die **praktische Anwendung der konzeptuell erarbeiteten graphbasierten Visualisierungsmethoden**. Zuerst wird in Kapitel 6.4.1 das Assistenzsystem interne, graphbasierte Visualisierungsmodul und danach in Kapitel 6.4.2 das externe Visualisierungsprogramm Tulip, welches in das Assistenzsystem integriert ist, angewendet.

6.4.1 Graphbasiertes Visualisierungsmodul

Da Screenshots von größeren Graphen aufgrund deren eingeschränkter Lesbarkeit nicht zweckdienlich sind, werden stark **vereinfachte Graphen** des Demonstrator-Getriebes genutzt. So sind drei Visualisierungsstile des graphbasierten Visualisierungsmoduls zusammen mit deren jeweiligen Farblegenden in Bild 83 und etwas vergrößert in Anhang A.24 dargestellt.

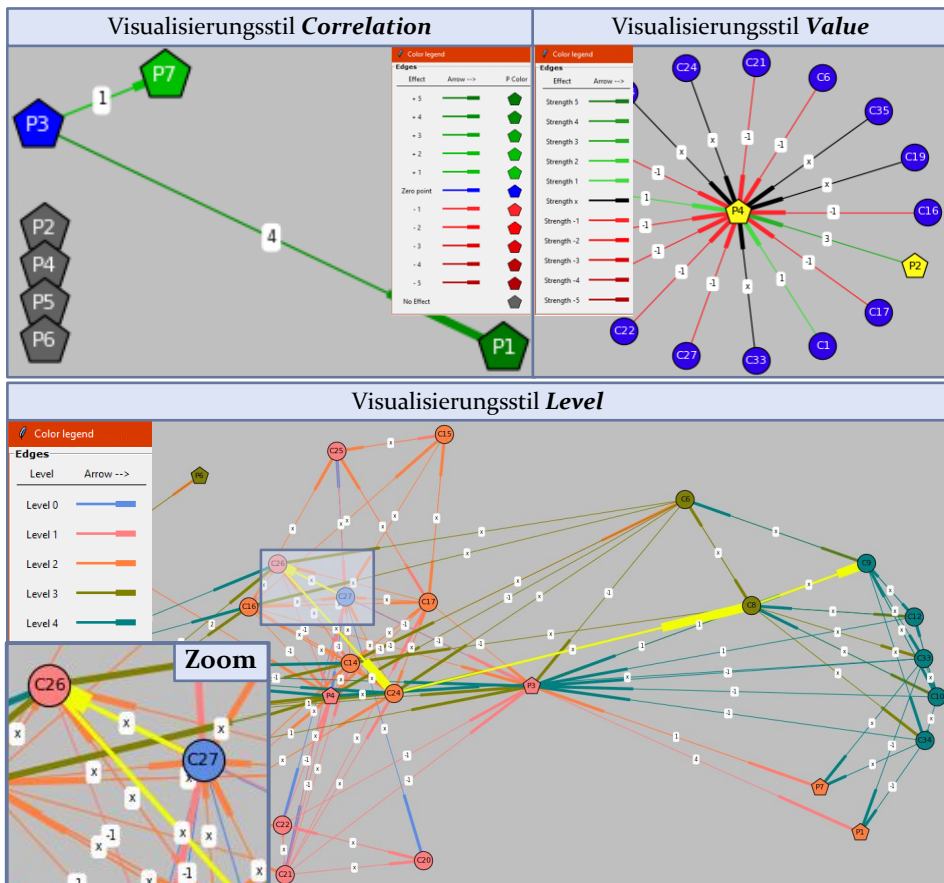


Bild 83: Drei verschiedene Visualisierungsstile des graphbasierten Visualisierungsmoduls

Der **Visualisierungsstil Correlation** (s. Kapitel 5.4.1) dient ausgehend von einem selektierten Ausgangselement (z. B. Eigenschaft P_3 in blau) dazu alle direkt abhängigen Elemente anzuzeigen. In Bild 83 werden aufgrund der gewählten Einstellungen komplementäre Elemente (z.B. P_1 und P_7) und deren Relationen in grün dargestellt. Dabei wird sowohl durch die Länge der Relation (d. h. Abstand der Elemente zueinander) als auch durch die Ziffer (z. B. 4 zwischen P_3 und P_1) der Relation die Stärke bzw. Wertigkeit der komplementären Beziehung beschrieben. Da die Relationen-Richtung durch eine „Verdickung“ der Linie am jeweiligen Ende symbolisiert wird, handelt es sich bei beiden Relationen um einseitig gerichtete Relationen (z. B. P_3 beeinflusst P_1). Aufgrund der vorgenommenen Einstellungen des Komplexitätsingenieurs werden erstens nur Eigenschaften, zweitens keine konfliktären Eigenschaften und drittens die Eigenschaften mit indifferenten Beziehungen (s. Bild 83 links unten in grau) nur aufgelistet dargestellt. Dadurch können insb. komplementäre, konfliktäre und indifferente Auswirkungen in Folge der Modifikation eines Elements erkannt werden.

Mit dem **Visualisierungsstil Value** (s. Kapitel 5.4.1) sind in Bild 83 im Graphlayout *Circle* alle Produktmerkmale (in Dunkelblau) und Produkteigenschaften (in Gelb) dargestellt, die eine bestimmte Produkteigenschaft (z. B. P_4 im Kreismittelpunkt) direkt beeinflussen (d. h. *Reverse Direction*). So wird P_4 bspw. von P_2 positiv mit der Stärke +3 und von $C16$ negativ mit der Stärke -1 sowie von $C19$ mit einer indifferenten Beziehung x (i. S. v. nicht mit einem bestimmten Wert beschreibbaren Beziehung) beeinflusst. Dadurch lassen sich u. a. folgende Fragen beantworten: Welches gewählte Element wird von welchen Elementen wie, in welcher Stärke und in welchem Grad (z. B. direkt, indirekt) beeinflusst? Welches gewählte Element beeinflusst welche Elemente wie, in welcher Stärke und in welchem Grad?

Mit dem **Visualisierungsstil Level** (s. Kapitel 5.4.1) werden in Bild 83 die Elemente und deren Relationen je nach Zugehörigkeit zu einer bestimmten Ebene eingefärbt. Ausgehend von einem gewählten Ausgangselement $C27$ (s. Zoom) auf *Level 0* werden alle Abhängigkeiten 1. Grades (d. h. direkte Relationen) auf *Level 1* (z. B. $C26$), 2. Grades auf *Level 2* bis zu in diesem Beispiel 4. Grades auf *Level 4* in jeweils unterschiedlichen Farben (s. *Color legend*) dargestellt. Dabei können z. B. die Wertigkeiten der Relationen als auch bidirektionale (d. h. wechselseitige) Relationen visualisiert werden. Dadurch werden Änderungsfortpflanzungen und -ketten über mehrere Ebenen hinweg transparent dargestellt und nachvollziehbar gemacht. Hierfür eignen sich v. a. auch Pfaddarstellungen, wie der in Bild 83 in Gelb markierte Pfad zwischen dem Ausgangselement $C27$ und dem Zielelement $C9$ in *Level 4*.

6.4.2 Externes Visualisierungsprogramm Tulip

Nach der Auswahl der mit Hilfe von Tulip zu visualisierenden Matrix bzw. Matrizen über die GUI des Assistenzsystems (s. Bild 49) werden diese von der Wissensbasis in Tulip transformiert (s. Kapitel 5.4.2). Anschließend können in Tulip mittels eines Konfigurationsfensters einige Einstellungen vorgenommen werden (z. B. Layout- und Farbeinstellungen). Auf Basis von Einstellungen und bspw. von bereits im Assistenzsystem gruppierten (i. S. v. geclusterten oder Ebenen oder Domänen zugeordneten) Elementen wird dem Anwender in Tulip eine Auswahl an möglichen Ansichten mit jeweils möglichen Subgraphen vorgeschlagen.

Im Folgenden dient **ein vereinfachtes Wälzlager mit ausgewählten Anforderungen und Prozessschritten als Anwendungsbeispiel**, welches mit Tulip in der Ansicht *Overview* mit dem Layout *Bubble Tree* in Bild 84 dargestellt wird. Darin sind die Anforderungselemente grün, die Produktelemente blau und die Prozesselemente rot eingefärbt.

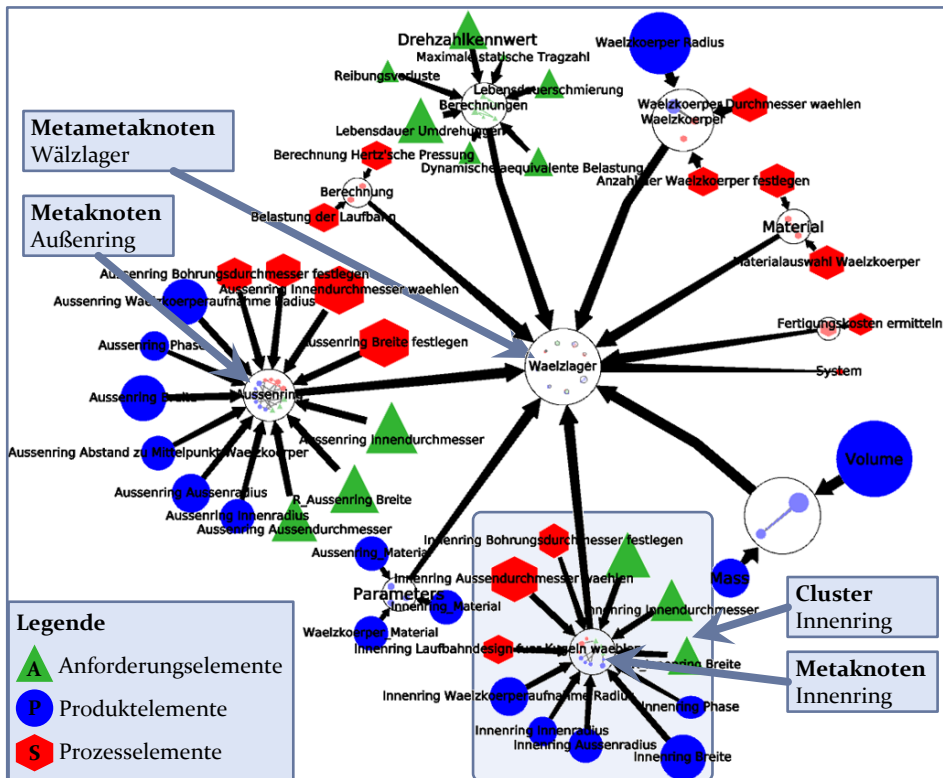


Bild 84: Darstellung eines vereinfachten Wälzlagers mit Tulip in der Ansicht *Overview* mit dem Layout *Bubble Tree* (in Anlehnung an [S24])

Aus der **Visualisierung des matrixbasierten Modells des Systems Wälzlager** mit den drei Domänen Anforderung, Produkt und Prozess kann der Komplexitätsingenieur viele Erkenntnisse gewinnen. Aufgrund der unterschiedlichen Farben und Notationssymbole lassen sich die einzelnen Elemente den diversen Domänen gut zuordnen. Die Größe der Elemente in dieser Darstellung steht in Abhängigkeit zur Anzahl der Relationen des jeweiligen Elements. So hat bspw. das Produktelement *Volume* relativ zu den anderen Elementen am meisten Relationen zu anderen Elementen und ist dementsprechend am größten (s. Bild 84, vgl. auch Anhang A.24). Über die Tulip-Funktion *Display node or edge properties* lassen sich für alle Elemente und Relationen Informationen, die aus dem Assistenzsystem übernommen oder in Tulip ergänzt wurden, anzeigen und ggf. manipulieren.

Für eine **bessere Übersichtlichkeit** besteht die Möglichkeit des Zooms, womit auch ausgeblendete Bezeichnungen angezeigt werden, und des Einklappens von sog. Metaknoten (z. B. Metaknoten Innenring), womit die Elemente des Cluster Innenring ausgeblendet werden. Zudem lassen sich die Elemente vom Anwender auch beliebig und interaktiv verschieben. Eine von zahlreichen weiteren Darstellungsmöglichkeiten ist die Ansicht *Highlight node neighborhood*, in der direkt in Abhängigkeit stehende Elemente angezeigt und andere Elemente transparent dargestellt werden. Diese Ansicht ist am Beispiel des Metaknoten *Aussenring* in Bild 85 gezeigt.

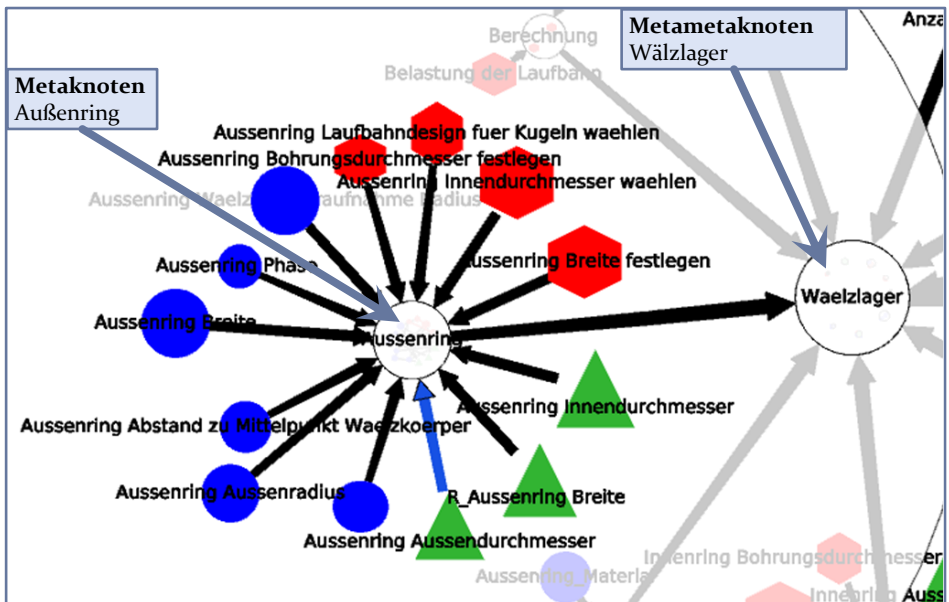


Bild 85: Darstellung eines vereinfachten Wälzlagers mit Tulip in der Ansicht *Highlight node neighborhood* für den Metaknoten *Aussenring* (in Anlehnung an [S24])

6.5 Analyse von matrixbasierten Modellen

Die Darstellung der **Anwendung der Analysemethoden** ist Gegenstand dieses Kapitels. Dazu wird in Kapitel 6.5.1 das mathematische Analysemodul beispielhaft angewendet. Danach erfolgt die exemplarische Anwendung des Moduls zur Anforderungsanalyse in Kapitel 6.5.2 und des Moduls zur Kostenanalyse in Kapitel 6.5.3. Dabei sei darauf hingewiesen, dass zwar alle Analysemethoden am zuvor vorgestellten, vereinfachten Lehrstuhlgetriebe angewendet wurden, aber aufgrund der Darstellungsmöglichkeiten in dieser Arbeit nur anhand ausgewählter Beispiele gezeigt werden können.

6.5.1 Mathematisches Analysemodul

Nach Auswahl des vereinfachten Wälzlager-Produktmodells sowie des **Clustering mit dem ZAKARIAN-Algorithmus** (s. Kapitel 5.5.1) wird dem Komplexitätsingenieur das Resultat in Bild 84 angezeigt. Darin ist gezeigt, dass das gewählte Clusterverfahren für die 17 Elemente des Wälzlagers vier Cluster gebildet hat. So umfasst bspw. das Cluster 1 alle mit dem *Wälzkörper Radius* in Relation stehenden Elemente. Somit repräsentieren Cluster diejenigen Elemente, die sich v. a. gegenseitig stark beeinflussen, womit Cluster eine gute Ausgangsbasis zur Modularisierung sind. Dabei bleibt aber die Interpretation durch den Komplexitätsingenieur wichtig, da das *Volumen* z. B. anstatt Cluster 3 dem Cluster 4 zugeordnet werden könnte.



Bild 86: Resultat des Clustering mit dem ZAKARIAN-Algorithmus für vereinfachtes Wälzlager

Neben der CPI-Methode wird als eine weitere, exemplarische Methode des Analysemoduls die **Vorwärtsgerichtete Analyse** (s. Kapitel 5.5.1) mit Hilfe des Anwendungsbeispiels dargestellt. Nach Auswahl der zu analysierenden Matrix oder Matrizen (z. B. DSM-Produkt mit den sieben Elementen E und DSM-Prozess mit den zwölf Elementen P) und der Vorwärtsgerichteten Analyse ist durch den Anwender in einem weiteren Fenster das Ausgangselement für die Vorwärtsgerichtete Analyse zu selektieren. Dazu werden in einem Auswahlfenster alle Elemente der ausgewählten Matrix bzw. Matrizen aufgelistet. Nach Auswahl des Ausgangselements (z. B. Element E_5 in Bild 87) werden im Fenster *Result of Feed-Forward-Analysis* die von diesem Element auf der 1. und 2. Ebene beeinflussten Elemente aufgezählt. So ist zu erkennen, dass Element E_5 die vier Elemente E_6 , P_2 , P_{10} und P_{11} direkt beeinflusst. Zudem werden die Elemente E_4 , E_5 , P_{11} , P_1 , P_{12} und E_6 indirekt (d. h. Relation 2. Grades) von E_5 beeinflusst, wie bspw. E_4 durch E_6 und E_5 durch E_6 , P_2 , P_{10} und P_{11} (d. h. durch alle Elemente, die auch direkt mit E_5 in Relation stehen).

Somit wird der Komplexitätsingenieur mit Hilfe der Vorwärtsgerichteten Analyse bei der **Analyse der Auswirkungen von geplanten Modifikationen von bestimmten Systemelementen** unterstützt. In dem in Bild 87 gezeigten Anwendungsbeispiel beeinflussen gleich vier Elemente wiederum das Ausgangselement, welches geplant ist zu modifizieren. Aufgrund dadurch steigender Komplexität sind derartige Änderungsketten bzw. -kreise nach Möglichkeit ebenso zu vermeiden wie Elemente, die über verschiedene Ebenen mehrfach beeinflusst werden (z. B. Element P_{11}).

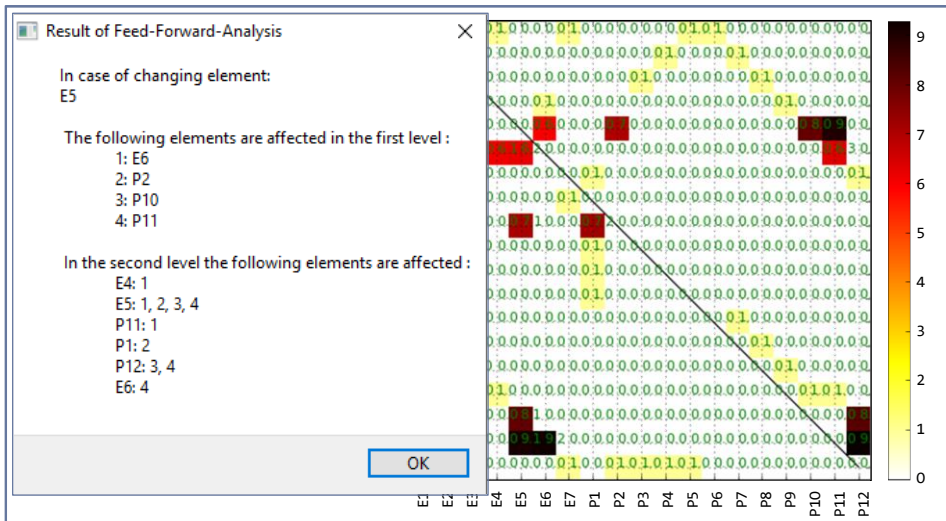


Bild 87: Resultat der Vorwärtsgerichteten Analyse für eine vereinfachte MDM

6.5.2 Requirements Analysis Module

Nachfolgend wird das in Kapitel 5.5.2 vorgestellte **Modul Requirements Analysis anhand des bereits bekannten Wälzlagers** mit ausgewählten Anforderungen angewendet. Dabei liegt der Fokus auf der multikriteriellen Bewertungen der Erfüllung von (quantifizierten) Anforderungen und somit von entsprechenden Zielwerten durch Produkteigenschaften und den mit diesen in Beziehung stehenden Merkmalen. Für diese Aufgabenstellung werden dem Komplexitätsingenieur durch das *Requirements Analysis Module* folgende Teilmodule zur Verfügung gestellt:

- *Requirements Browser* dient zur Eingabe und Manipulation von Anforderungen sowie von wichtigen Informationen zu Anforderungen
- *Search Requirements* dient zur Suche und Filterung von Anforderungen (s. Bild 65)
- *Filter Requirements* dient zur kontextspezifischen Filterung von Anforderungen und deren Visualisierung als Matrix oder Graph
- *Setpoint Comparison* dient für (graphische) Soll-Ist-Vergleiche von Zielwerten der Anforderungen mit Produkteigenschaftswerten
- *Structural Graphs* dient zur Visualisierung z. B. in Form von Pfaden von Anforderungs- und Produktelementen (s. Bild 64).

Die **Basis für die Analyse von Anforderungs- und Produktelementen** stellen die erforderlichen $DSM_{Specification}$, $DSM_{Product}$ und $DMM_{Specification-Product}$ sowie $DMM_{Product-Specification}$ (s. Bild 48) dar. Die Modellierung dieser matrix-basierten Modelle ist in Kapiteln 5.2 und 5.3 erläutert. Zur Manipulation und Ergänzung von Elementen und deren Relationen (inkl. hilfreicher Informationen und Attribute) kann das *Modifying Module* (s. Kapitel 5.6.1) und das *Functional Analysis Module* (s. Kapitel 5.6.2) verwendet werden.

Der Komplexitätsingenieur kann mit Hilfe des **Requirements Browser** Anforderungen mit ihren jeweils zugehörigen Informationen (vgl. Bild 88) erzeugen, ergänzen und manipulieren. So kann der Anwender im Bereich *General Information* Anforderungen erzeugen bzw. manipulieren, im Bereich *Semantic Information* Anforderungen im Feld *text* konkretisieren bzw. erläutern sowie bzgl. *type*, *group* und *class* kategorisieren als auch im Bereich *Quantification* Anforderungen bezüglich Zielwerte(s) und Einheit quantifizieren. So wird am Wälzlagerbeispiel der Aussendurchmesser D mit der ID R_1 um semantische Informationen im Bereich *Semantic Information* angereichert (s. Bild 88). Da in diesem Beispiel der Aussendurchmesser D exakt 26 mm betragen soll, ist in Bild 88 der Operator „==“ selektiert (vgl. auch Tabelle 1 in Kapitel 5.5.2).

The screenshot shows the 'Requirements Browser' window with three main sections: General Information, Semantic Information, and Quantification.

- General Information:**
 - Name: Aussendurchmesser D
 - ID: R1
 - Level 1: (empty)
 - Level 2: (empty)
- Semantic Information:**
 - Text: Der Aussendurchmesser des Waelzlagers muss 26 mm betragen.
 - Categories:
 - Type: Fixed Requirement
 - Group: Technology
 - Class: Functional Requirement
- Quantification:**
 - Operator: ==
 - Lower Setpoint: (empty)
 - Upper Setpoint: (empty)
 - Exact Setpoint: 26
 - Unit: mm

At the bottom, there are three buttons: New, Delete, and Save.

Bild 88: Manipulation von Metainformationen der Anforderung im *Requirements Browser*

Für die **Analyse, Visualisierung und Bewertung von Änderungen und deren Auswirkungen** auf bestimmte Systemelemente, wie bspw. Anforderungen in Folge der Modifikation eines bestimmten Systemelements, wie bspw. Produktmerkmale, dient das Teilmodul *Setpoint Comparison*. Damit lassen sich die mit einer quantifizierten Anforderung einhergehenden Sollwerte bzw. Sollwertintervalle mit den verknüpften Produkteigenschaften und -merkmalen und deren Werten vergleichen. Dabei kann der Soll-Ist-Vergleich zum einen mit dem *Current Value* des Produktelements und zum anderen auch mit einem *New Value* (i. S. v. alternativen Wert) für ein Produktelement vorgenommen werden. Dabei wird der *Current Value* für das ausgewählte Element aus der Wissensbasis des Assistenzsystems geladen. Dagegen dient der gesetzte Wert im Feld *New Value* nur zur Darstellung von Änderungen und deren Auswirkungen und wird nicht gespeichert.

Zur **Veranschaulichung des Teilmoduls *Setpoint Comparison*** dienen die beispielhaft in Bild 89 dargestellten, ausgewählten Anforderungen. So wird durch den graphischen Soll-Ist-Vergleich der Anforderung „Lebensdauer modifiziert“ mit der Produkteigenschaft „modifizierte Lebensdauer“ des Wälzlagers im Diagramm dargestellt (s. Bild 89 links unten), dass die Produkteigenschaft (s. grün gefärbte Raute) die Anforderung (gestrichelte schwarze Linie, d. h. Sollwert > 50.000 h) erfüllt. Durch die Eingabe eines alternativen Werts (*New Value*) und der Aktualisierung des Diagramms (*Redraw Graph*) kann der Komplexitätsingenieur den Wert des Produktelements modifizieren und dadurch herausfinden, ob alternative Werte den Sollwert (s. Bild 89 links unten) bzw. das Sollwertintervall (s. Bild 89 rechts unten) der Anforderung erfüllen. Sofern die Speicherung modifizierter Werte von Systemelemente in die Wissensbasis des Assistenzsystems vorgenommen werden soll, ist dazu das *Modifying Modul* (vgl. Kapitel 5.6.1) oder das *Functional Analysis Module* (vgl. Kapitel 5.6.2) zu verwenden.

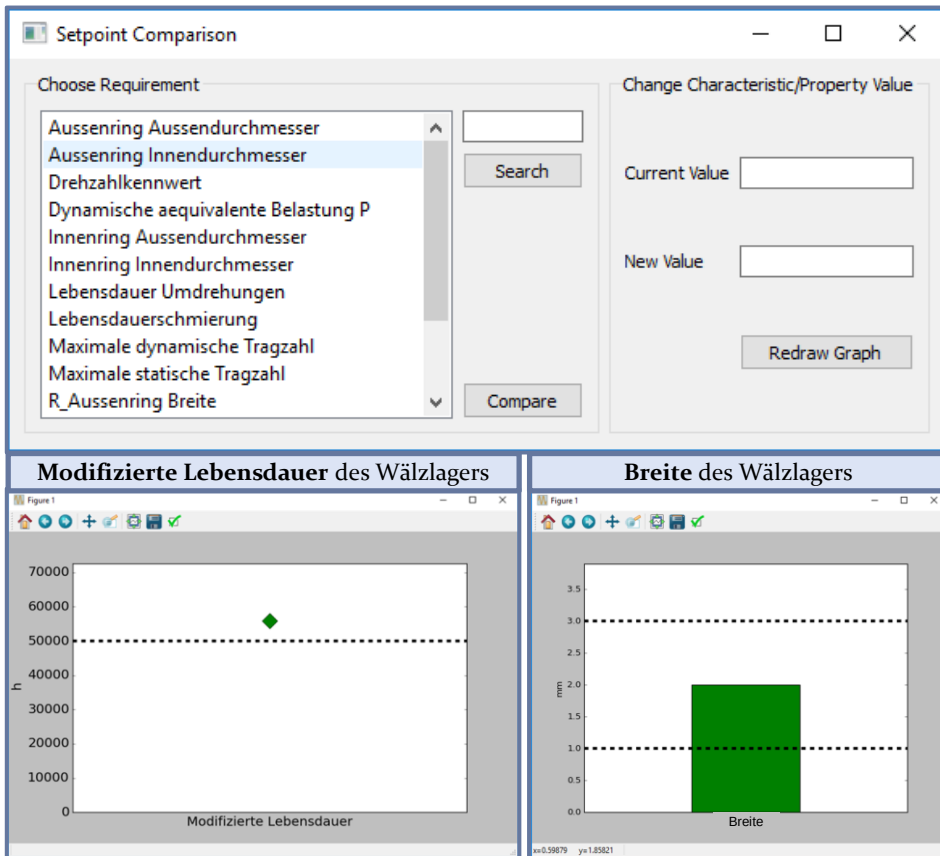


Bild 89: Soll-Ist-Vergleich von Anforderungs- und Produktelementwerten

6.5.3 Cost Analysis Module

Die **Anwendung des Moduls Cost Analysis** (s. Kapitel 5.5.3) am Beispiel der Wissensdomäne steht im Mittelpunkt dieses Kapitels. Der Fokus liegt auf der Darstellung verschiedener Nutzenpotentiale des Moduls für den Komplexitätsingenieur zum kostenorientierten Komplexitätsmanagement. Für diese Aufgabenstellung stellt das Assistenzsystem dem Anwender viele Funktionalitäten zur Verfügung, wovon nachfolgend einige exemplarisch vorgestellt werden.

Mit dem €-Button unterhalb der *DSM-Knowledge* (vgl. Bild 49) wird das Kostenmodell für die Wissensdomäne erstellt bzw. das bereits vorhandene geladen. Dazu sind zuerst einige Optionen vom Anwender zu wählen, wie bspw. ob alle Dokumente eines gewissen Typs (z. B. CAD-Dateien) zu einer Kostenposition kombiniert werden sollen (z. B. *KID11 CAD combined* in

Bild 90). Dadurch hat der Anwender die Auswahl zwischen der Zuweisung der jeweiligen Kosten zu den einzelnen CAD-Dateien und der Zuweisung der aggregierten Kosten für alle WI-Elemente des Typs CAD. Somit kann der Komplexitätsingenieur je nach Zielsetzung einen geeigneten Detaillierungsgrad der Zuweisung von Kosten zu WI-Elementen wählen. Dabei geht ein hoher (niedriger) Detaillierungsgrad der Kostenzuweisung mit einem hohen (niedrigen) Aufwand und einem hohen (niedrigen) Nutzen einher. Anschließend öffnet sich das **Fenster Cost Request for the Knowledge Domain**, in welchen bestimmte Werte durch Eingaben oder mittels eines Drop-Down-Menüs ergänzt und manipuliert werden können (s. Bild 90). Dabei werden in diesem Beispiel die zeitabhängigen Kosten durch die Angabe der *Lead Time* sowie der *Wage Rate* und die zeitunabhängigen Kosten durch die *Acquisition Costs* bestimmt (vgl. Kapitel 5.5.3). So beträgt bspw. die Modellierungsdauer 15 Minuten für das CAD-Modell der Abtriebswelle (s. *KID11 abtriebswelle.prt.12*) durch einen Konstrukteur mit dem fiktiven Stundensatz von 28€/h, wobei keine Akquisitionskosten anfallen.

Cost Request for the Knowledge Domain

Fill in only integers (whole numbers)!

Please fill in the information.

Target Costs: The amount of costs this domain may cause.
 Lead Time: The amount of time required to create the Knowledge Element.
 Wage Rate: The rate of pay of the responsible employee.
 Acquisition Costs: Costs caused due to external acquisition of a Knowledge Element.

Target Costs of this Domain in € 1200

ID	Element Name	Lead Time in min	Wage Rate in €	Acquisition Costs in €
KID4	Markt-Konkurrenz-Analyse.pptx	190	low: 12 €/h	0
KID5	Trendanalyse.pptx	220	high: 28 €/h middle: 20 €/h low: 12 €/h	0
KID6	Bundesgesetz.pdf	26	low: 12 €/h	0
KID7	DIN-Norm_Jagerhalbschalen.pdf	15	low: 12 €/h	300
KID8	Patent.pdf	20	low: 12 €/h	400
KID9	Funktionsstruktur.pdf	40	high: 28 €/h	0
KID10	Morphologischer Kasten.xlsx	20	middle: 20 €/h	0
KID11	abtriebswelle.prt.12	15	high: 28 €/h	0
KID12	sicherungsring.prt.1	4	high: 28 €/h	15
...				
	Done			

Aggregierte Kosten für alle KID-Elements des Typs CAD

KID11	CAD combined	284	high: 28 €/h	15

Eingaben des Datentype integer

Auswahl mit Drop-Down-Menü

Bild 90: Vereinfachtes Kostenmodell der Wissensdomäne (in Anlehnung an [S13])

Nach der Modellierung und Manipulation von Kostenmodellen können diese mit dem Assistenzsystem visualisiert werden (s. *Cost Visualization* in der GUI in Bild 49). Dafür kann der Komplexitätsingenieur im **Fenster *Cost Visualization*** sowohl eines der verfügbaren Kostenmodelle wählen und andererseits die anzuzeigenden Metadaten selektieren (s. Bild 91).

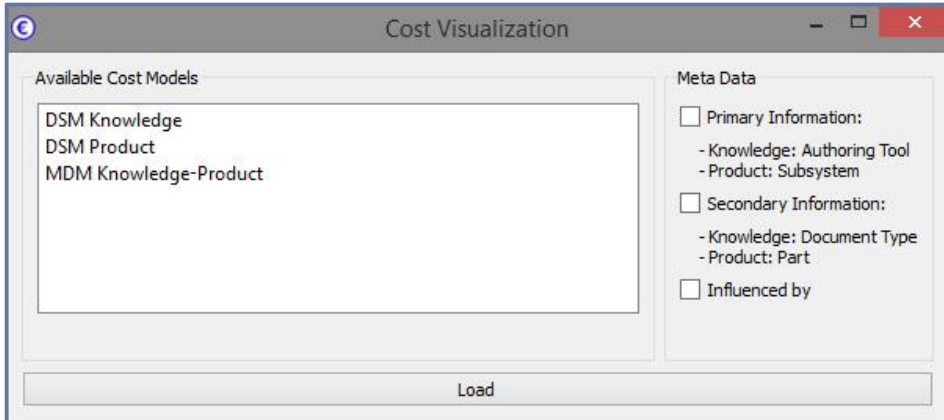


Bild 91: Auswahlmöglichkeiten bzgl. des zu visualisierenden Kostenmodells

Anschließend erfolgt die **Darstellung des Kostenmodells**, wie bspw. für die ausgewählte Wissensdomäne mit den selektierten *Metadaten Primary Information* und *Influenced by* (s. Bild 92). Im vereinfachten Anwendungsbeispiel in Bild 92 sind neben ausgewählten Wissenselementen auch zwei Spezifikationselemente in das Kostenmodell integriert als auch alle CAD-Dateien zum Wissenselement *CAD combined* zusammengefasst. Zudem werden die *Total Costs* als Summe aller *Single Costs* (EK) in Grün angezeigt, da diese unterhalb der *Target Costs* sind. Mittels Mouseovers werden dem Komplexitätsingenieur bei Bedarf detailliertere Informationen zu Spaltenbezeichnungen als auch zu Einträgen angezeigt, wie z. B. die Berechnung der *Single Costs* für das *KID7* in Bild 92, was sowohl der Übersichtlichkeit dient als auch einen großen Nutzen zur Analyse und Bewertung darstellt. Besonders hilfreich für das Komplexitätsmanagement ist die Auflistung der IDs der Ausgangselemente, welche das jeweilige Zielelement direkt und indirekt beeinflussen. Die *Accumulated Costs* für ein Element setzen sich aus dessen eigenen *Single Costs* und der *Single Costs* der beeinflussenden Ausgangselemente zusammen. So ergeben sich bspw. die *Accumulated Costs* (140,67 €) für das Wissenselement *KID4* aus der Addition der eigenen *Single Costs* (38,0 €) mit den *Single Costs* von *KID5* (102,67 €). Demzufolge wird dem Anwender nachvollziehbar und übersichtlich aufgezeigt, welche Kosten die Generierung eines bestimmten Wissenselement unter Berücksichtigung aller der zuvor erforderlichen Wissenselemente verursacht.

Der **Prozentsatz in der Spalte *Ratio*** ist der Quotient aus *Accumulated Costs* und *Total Costs*, d. h. Kostenanteil eines bestimmten WI-Elements über den gesamten, bisherigen PEP an den Total Costs. Damit dient dieser Prozentsatz dem Komplexitätsingenieur dazu die entstehenden Änderungskosten bei einer Veränderung eines bestimmten Wissenslements besser abzuschätzen. Dabei kann generell bzgl. eines bestimmten Elements davon ausgegangen werden, dass je größer der Prozentsatz des Elements ist und/oder je später dieses Element im PEP generiert wurde, desto größer die Auswirkung(en) auf die Kosten bei dessen Änderung sind.

Weitere und **detailliertere Darstellungen von Kostenmodellen** der Wissensdomäne und der Produktdomäne sind im Anhang A.25 zu finden. So werden im Anhang A.25 z. B. in Bild 160 die CAD-Dateien – im Gegensatz zum Bild 92 – nicht zusammengefasst, sondern jeweils einzeln dargestellt. Dadurch wird zwar einerseits aufgrund des höheren Detaillierungsgrads die Analyse der Relationen auch zwischen den einzelnen CAD-Dateien ermöglicht jedoch andererseits die Übersichtlichkeit der Visualisierung (negativ) beeinflusst. Somit muss der Komplexitätsingenieur nach subjektiver Präferenz und verfolgter Zielsetzung zwischen den verschiedenen, möglichen Kostenmodellen fallweise entscheiden.

Zusammenfassend lässt sich schlussfolgern, dass **Kostenmodelle für die Analyse, Visualisierung sowie Optimierung** der Kosten der Elemente in den einzelnen Domänen dienen. So kann der Komplexitätsingenieur durch die gezielte Modifikation vieler der zuvor beschriebenen Faktoren die Auswirkungen auf die einzelnen Elementkosten (*EK*), akkumulierten Kosten (*AK*) und *Totals Costs* mittels der durch das Assistenzsystem bereit gestellten Module i. S. einer Sensitivitätsanalyse simulieren. Auf Grundlage der dadurch gewonnen Ergebnisse und Erkenntnisse kann der Anwender sowohl die einzelnen Kosten als auch die Gesamtkosten des zu entwickelnden Produkts als auch der gesamten PE optimieren. Dadurch unterstützt das Assistenzsystem den Komplexitätsingenieur auch beim holistischen, kostenorientierten Komplexitätsmanagement.

7 Kritische Diskussion und Reflexion

„Es gibt nichts Praktischeres als eine gute Theorie“

KURT T. LEWIN (1890 - 1947)

Nach dem bedeutenden, praktischen Psychologen LEWIN, der auch als ein Begründer der modernen experimentellen Sozialpsychologie gilt, gibt es nichts Praktischeres als eine gute Theorie. Demgemäß ist auch eine gute **Theorie für das holistische Komplexitätsmanagement** in der PE das Ziel dieser Arbeit (s. Kapitel 1.3). Dafür wurden in Kapitel 2 zunächst die theoretischen Grundlagen zur Modellierung komplexer Systeme erläutert. Anschließend wurden auf Grundlage des Stands der Forschung und Technik (s. Kapitel 3) der Handlungsbedarf und die Anforderungen aus Sicht der Wissenschaft und der Wirtschaft in Kapitel 4 abgeleitet.

Die aufgrund dieser Anforderungen vom Autor entwickelte **Methodik zur Unterstützung des Komplexitätsingenieures beim holistischen Komplexitätsmanagement** in der integrierten PE wird zusammen mit dem implementierten Assistenzsystem (vgl. Kapitel 5) in diesem Kapitel kritisch diskutiert. Dazu wird das in dieser Arbeit entwickelte Fachkonzept und dessen prototypische Implementierung in ein Assistenzsystem bzgl. der Zielsetzung und der Anforderungen sowohl aus Sicht der Wissenschaft in Kapitel 7.1 als auch aus Sicht der Wirtschaft in Kapitel 7.2 kurz reflektiert. Dabei erfolgt die Evaluation der Methodik auf Grundlage der Ergebnisse und Erkenntnisse aus den Anwendungsbeispielen in Kapitel 6.

7.1 Diskussion aus Sicht der Wissenschaft

Nachfolgend werden die in Kapitel 4.1.1 erkannten **vier Handlungsfelder aus Sicht der Wissenschaft** und die daraus in Kapitel 4.2.2 abgeleiteten Anforderungen der Methodik und dem Assistenzsystem gegenübergestellt. Dazu wird das Kapitel anhand dieser Handlungsfelder gegliedert, welche z. T. auch für die industrielle Praxis Relevanz haben.

Handlungsfeld Modellierung von DSMs

Aus Sicht der Wissenschaft besteht Handlungsbedarf bzgl. der möglichst automatisierten Modellierung von DSMs auf Basis der bereits bestehenden Modelle aus den diversen Domänen (s. Kapitel 3.1). Daher ist in Kapitel 5.1 eine neue Methodik vorgestellt worden, die auf Grundlage einer definierten Aufgaben- oder Problemstellung die Modellierung von DSMs umfassend

unterstützt. Hierfür wurden geeignete **IT-Schnittstellen zwischen den domänenspezifischen IT-Systemen und dem neuen Assistenzsystem** entwickelt, um die bis dato manuelle und damit sehr zeitaufwändige und fehleranfällige Akquisition der benötigten Daten und Informationen für die matrixbasierte Modellierung der in der PE relevanten Domänen zu automatisieren und somit signifikant zu verbessern. Dies beschreiben u. a. BIEDERMANN & LINDEMANN und BROWNING als eines der wichtigsten Forschungsfelder [57], [88]. Dieses Handlungsfeld mit den dazugehörigen Anforderungen wird v. a. durch das Teilkonzept zur Modellierung von DSMs und dessen Implementierung in ein Assistenzsystem erfüllt (s. Kapitel 5.2).

Im Gegensatz bspw. zum Ansatz zur Modulpartition für 3D-CAD-Baugruppenmodelle nach LI & XIE [349] können mit der in Kapitel 5.2.1 dargestellten Methode nicht nur die Relationen zwischen Bauelementen, sondern auch die verschiedenen Arten von Relationen zwischen diesen dargestellt werden. Dieser **Fortschritt zum Stand der Forschung und Technik** führt dazu, dass Modelle komplexer Systeme sowohl mehr nützliche Informationen enthalten als auch weniger aufwändig zu modellieren sind.

Handlungsfeld Modellierung von DMMs

Für die Modellierung von DMMs sind jeweils **alle miteinander in Relation stehenden Elemente aus zwei verschiedenen DSMs zu verbinden**. So sind bspw. die Anforderungen mit den Produkteigenschaften und -merkmalen mittels DMMs zu verknüpfen. Zur Modellierung von DMMs sind unterstützende Methoden in Kapitel 5.3 vorgestellt. Mit Hilfe der ersten Methode wird auf Basis von zwei DSMs automatisiert eine zunächst leere DMM zu erzeugt, die in MS Excel oder direkt im Assistenzsystem ausgefüllt werden kann. Diese Methode stellt nur eine geringe Verbesserung zum Stand der Technik dar. Die zweite, neue Methode des smarten Mappings und Checks von Worten bzw. Bezeichnungen für die Identifikation und Modellierung potenzieller Relationen erleichtert die DMM-Erstellung dagegen deutlich. Da diese Methode jedoch u. a. die Verwendung von zuvor definierten eindeutigen Elementbezeichnungen aus verschiedenen Domänen erfordert, ist deren Einsatz in der Praxis nur dann hilfreich, wenn auch eine detaillierte Konstruktionsrichtlinie strikt befolgt wird, was aufgrund der Erfahrungen des Autors in der Praxis häufig nicht der Fall ist.

Neben diesen beiden Methoden werden auf Basis einer bidirektionalen IT-Schnitte zwischen dem Assistenzsystem und dem frei verfügbaren yEd-Grafikeditor auch **grafische Funktionen zur Modellierung von DMMs** zur Verfügung gestellt (s. Kapitel 5.3.1). Durch diese graphbasierte Computerunterstützung wird die Modellierung von DMMs deutlich erleichtert.

Die in Kapitel 5.3.2 dargestellte vierte Methode mit dem darauf aufbauenden Modul des Assistenzsystems **Domain Mergers dient zur effizienten Modellierung und Manipulation von Relationen**, die domänenspezifisch als auch domänenübergreifend sein können. Mittels der vielfältigen Optionen für die intuitive Modellierung und Manipulation von Relationen, die mit einer benutzerfreundlichen und adaptiven GUI bereitgestellt werden, sinken der Aufwand und die Fehleranfälligkeit signifikant.

Handlungsfeld Visualisierung und Kommunikation

Im Gegensatz zu matrixbasierten Darstellungen von Modellen komplexer Systeme dient die graphbasierte Visualisierung von DSMs, DMMs sowie MDMs zur besseren Verständlichkeit und Nachvollziehbarkeit von und zielgerichteteren Kommunikation zwischen Personen. Dafür stellt das graphbasierte Visualisierungsmodul des Assistenzsystems (s. Kapitel 5.4.1) **verschiedene Visualisierungsstile und Graphlayouts für die ziel- und bedarfsgerechte Visualisierung** bereit. So hilft das computerunterstützte Vorgehen zur graphbasierten Visualisierung bspw. bei der Wahl des Ausgangselements, der Beeinflussungsrichtung, Anzahl der Levels sowie Darstellung von verschiedenen Pfaden zwischen bestimmten Elementen.

Zudem ist das Visualisierungsmodul nicht nur durch weitere Python-Bibliotheken bei Bedarf anpassbar, sondern **auch mit externen Visualisierungsprogrammen flexibel erweiterbar**, wie z. B. Tulip (s. Kapitel 5.4.2). Somit werden die Anforderungen in Kapitel 4.1.1 erfüllt als auch ein Fortschritt zum Stand der Forschung und Technik realisiert. So enthält das Assistenzsystem im Vergleich zu den in Kapitel 3.3.4 vorgestellten Softwaresystemen mehr und bessere Methoden und hilfreiche Funktionalitäten zur Visualisierung von Modellen komplexer Systeme.

Handlungsfeld Analyse und Bewertung

Da die Softwarelösungen zum Komplexitätsmanagement (s. Kapitel 3.3.4) zur matrixbasierten Analyse und Bewertung nicht oder nur eingeschränkt geeignet sind, sind vom Autor – wie in Kapitel 5.5 beschrieben – nicht nur bestehende, sondern auch neue **Methoden zur Analyse und Bewertung** in das Assistenzsystem implementiert worden. Dafür sind in das Analysemodul sowohl individuelle Analysen für bestimmte Domänen als auch Verfahren zur Durchführung holistischer, domänenübergreifender Analysen integriert (s. Kapitel 5.5.1). Zudem sind für die Analyse und multikriteriellen Bewertung von komplexen Systemen zur Erfüllung von Spezifikationen das Modul *Requirements Analysis* sowie für die Analyse und Erreichung von Zielkosten das Modul *Cost Analysis* entwickelt und implementiert.

Diese entwickelten, computerunterstützten Methoden zur **holistischen Manipulation und Modifikation** von Elementen und Relationen (vgl. Kapitel 5.6) können grundsätzlich in allen fünf Phasen des Vorgehens zum matrixbasierten Management komplexer Systeme (vgl. Bild 45) eingesetzt werden. Dabei hat sich gezeigt, dass das *Modifying Module* sowohl bei der Modellierung als auch bei der Verbesserung komplexer Systeme besonders hilfreich für den Komplexitätsingenieur ist. Dagegen dient das Modul *Functional Analysis* v. a. zur mathematischen Modellierung von Relationen, wodurch die darauf aufbauenden Analysen zur multikriteriellen Bewertung deutlich aussagekräftiger und somit verbessert werden.

7.2 Diskussion aus Sicht der Wirtschaft

Zudem werden die **drei Handlungsfelder aus Sicht der industriellen Praxis** (s. Kapitel 4.1.2) beschrieben und die dazugehörigen Anforderungen der Methodik und dem Assistenzsystem (s. Kapitel 4.2.2) gegenübergestellt. Dazu ist dieser Abschnitt nach diesen Handlungsfeldern gegliedert.

Handlungsfeld Zeit- und Kostenreduzierung

Da z. B. Produktmodelle in der Praxis deutlich mehr Elemente als die Beispiele zur Veranschaulichung in dieser Arbeit enthalten, ist die manuelle Modellierung eines matrixbasierten Produktmodells mit allen Elementen und Relationen relativ fehleranfällig und zeitintensiv. Die (weitgehend) **automatisierte Modellierung der DSMs** auf Grundlage der z. B. aus CAD-Modellen oder Prozessmodellen in das Assistenzsystem importierten Daten mit den dargestellten Methoden und IT-Schnittstellen ist sehr einfach umzusetzen und zudem auch sehr nützlich, weil dadurch sowohl der manuelle Zeitbedarf als auch die Fehleranfälligkeit auf ein Minimum sinken.

Außerdem sind die vielen **automatisierten, bidirektionalen IT-Schnittstellen** zwischen dem Assistenzsystem und den spezifischen IT-Systemen bzw. Modellen der diversen Domänen v. a. aus den folgenden beiden Gründen zur Zeit- und Kostenreduzierung vorteilhaft. Erstens können dadurch die matrixbasierten Modelle im Assistenzsystem schnell aktualisiert werden, was aufgrund hoher Veränderlichkeit (Dynamik) im Zeitablauf von äußerst komplexen Systemen (s. Kapitel 2.2.2) oft erforderlich ist. Zweitens können im Assistenzsystem vorgenommene Modifikationen und Manipulationen der Werte von Elementen und Relationen sowie teilweise der Elemente und Relationen selbst in deren originären, domänenspezifischen Modellen bzw. IT-Systemen überführt werden. Jedoch besteht in Bezug zur bidirektionalen IT-Schnittstelle noch Verbesserungspotential.

Handlungsfeld Adaptier- und Erweiterbarkeit

Im Rahmen des Systementwurfs (s. Bild 44) wurde das **Fachkonzept des Assistenzsystems adaptiv und modular gestaltet**, sodass zum einen auch weitere Anwendungsprogramme, wie z. B. NX, für die automatisierte Modellierung mit dem Assistenzsystem verknüpft werden können und zum anderen neben den direkten auch indirekte IT-Schnittstellen zwischen Anwendungsprogrammen und dem Assistenzsystem möglich sind.

Aus industrieller Sicht ist es von Vorteil, dass Daten aus dem CAD-System direkt und auch indirekt über ein Excel-Format in das Assistenzsystem eingelesen werden können. Dies liegt zum einen daran, dass die Möglichkeit einer Bearbeitung der Daten in MS Excel für jedermann durchführbar ist und zum anderen die Daten aus MS Excel auch in eine Vielzahl weiterer IT-Werkzeuge eingelesen werden können. Für die indirekten IT-Schnittstellen wurden die **Exportmöglichkeiten der Anwendungsprogramme** analysiert. So bietet z. B. Enterprise Architect eine Modellierung in der Notationsform SysML und einen Export im *xmi*-Format, während es z. B. für CAD-Programme sinnvoll ist, den Export in das *xlsx*-Format zu realisieren, da in vielen IT-Systemen eine Importfunktion aus MS Excel verfügbar ist. Daher wurden zum direkten als auch indirekten Datenimport aus diversen IT-Werkzeugen in das Assistenzsystem spezifische Methoden entwickelt.

Handlungsfeld Nützlichkeit und Einfachheit

Neben den Elementen der verschiedenen Domänen können z. T. auch in den entsprechenden Modellen **vorhandene domänenspezifische Relationen im Rahmen der Modellierung von DSMs** berücksichtigt werden. So können z. B. aus parametrisierten CAD-Modellen in diesen vorhandene mathematisch-funktionale Relationen und Bedingungstypen (z. B. align, insert) zur automatisierten DSM-Modellierung verwendet oder im Assistenzsystem durch das Modul *Functional Analysis* ergänzt werden. Aufgrund der dadurch deutlich höheren Aussagekraft der Relationen kann der Komplexitätsingenieur effektivere und effizientere Bewertungen vornehmen und somit zielführendere Entscheidungen in der IPE treffen.

Auf Basis der Ergebnisse und Erfahrungen aus bisherigen Anwendungen der entwickelten Analyse- und Visualisierungsmethoden hat sich gezeigt, dass insb. die **Analyse und Visualisierung des Top-Down-Pfads und Bottom-Up-Pfads** für den Anwender besonders nützlich sind. Außerdem sind v. a. auch die holistische Analyse und Visualisierung von Elementen unterschiedlicher Domänen für ein kontextsensitives und zielgerichtetes Iterationsmanagement hilfreich.

8 Zusammenfassung und Ausblick

„So eine Arbeit wird eigentlich nie fertig, man muss sie für fertig erklären, wenn man nach Zeit und Umständen das Möglichste getan hat.“

JOHANN W. VON GOETHE (1749 – 1832)

GOETHE, einer der bekanntesten deutschen Dichter, hat dieses Zitat auf die Ausarbeitung seines Schauspiels „Iphigenie auf Tauris“ bezogen. Analog ist auch der Autor dieser Arbeit der Auffassung, dass das von ihm bearbeitete **Forschungsthema nie vollkommen abgeschlossen sein kann**. Daher wird anschließend an die zusammenfassenden Schlussfolgerungen in Kapitel 8.1 ein Ausblick auf weiteren Forschungsbedarf in Kapitel 8.2 gegeben.

8.1 Zusammenfassung und Schlussfolgerungen

Je höher die Komplexität und die oft damit einhergehende Volatilität, Unsicherheit und Mehrdeutigkeit (vgl. VUCA) werden, desto wichtiger ist das Komplexitätsmanagement in der PE. Daher ist das Ziel dieser Arbeit eine Methodik zu entwickeln, mit der **Entwickler beim holistischen Komplexitätsmanagement in der PE unterstützt** werden, indem geeignete Methoden für die Modellierung, Analyse, Visualisierung und Bewertung von komplexen Systemen zur Entscheidungsfindung bereitgestellt werden.

Im Rahmen dieser Arbeit wurde erkannt, dass zur Realisierung der Zielsetzung ein **grundlegendes Verständnis der Modellierung komplexer Systeme** erforderlich ist. Daher sind Grundlagen zu technischen Systemen und Komplexität sowie darauf aufbauend zur Erstellung von Modellen für komplexe System aus Sicht der PE in Kapitel 2 erläutert. Dadurch kann die aus der Vielzahl, Vielfalt, Veränderlichkeit und Vieldeutigkeit von Elementen und Relationen komplexer Systeme resultierende Herausforderung des Komplexitätsmanagements in der PE besser nachvollzogen werden.

Die **Diskussion des Stands der Forschung und Technik** hat gezeigt, dass domänenspezifische Modelle für ein ganzheitliches Komplexitätsmanagement in der PE unzureichend sind. Dagegen sind das Modellbasierte Systems Engineering, die Modellierung komplexer Systeme mittels Ontologien und die matrixbasierte Modellierung in Form von DSMs, DMMs und MDMs deutlich erfolversprechender. Aufgrund der universalen Eignung zur formalen und somit durch Rechner verarbeitbaren Modellierung aller Elementen und Relationen komplexer Systeme bilden die Ansätze für das matrixbasierte Komplexitätsmanagement den Fokus dieser Arbeit.

Die **Evaluation bestehender Methoden und IT-Werkzeuge für das matrixbasierte Komplexitätsmanagement** hat ergeben, dass großer Handlungsbedarf aus wissenschaftlicher und industrieller Sicht besteht. So ist die Transformation von Modellen aus den diversen Domänen in DSMs möglichst zu automatisieren, um Aufwand und Fehleranfälligkeit deutlich zu reduzieren. Zudem gibt es zur Modellierung von domänenübergreifenden Relationen in DMMs kaum Methoden und es existiert bislang keine Rechnerunterstützung. Weiterer Handlungsbedarf resultiert aus der Notwendigkeit geeigneter Visualisierungsmöglichkeiten verschiedener Matrizen, um z. B. die Änderungen und deren Auswirkungen auf bestimmte Systemelemente in Folge der Modifikation eines bestimmten Systemelements darzustellen. Für die Analyse von Matrizen sind zwar zahlreiche Methoden vorhanden, diese sind aber auf die unterschiedlichen Spezifika der Systemmodelle der einzelnen Domänen in der PE geeignet anzupassen.

Die Zielsetzung der Arbeit ist daher ein **Konzept zu entwickeln mit dem Entwickler beim holistischen Komplexitätsmanagement in der PE** unterstützt werden, indem kontextbezogen jeweils geeignete Methoden für die Modellierung, Analyse, Visualisierung und Bewertung komplexer Systeme zur Verfügung gestellt werden. Dazu ist die IT-Architektur des prototypisch implementierten Assistenzsystems aus Modulen aufgebaut, so dass dieses modular um weitere IT-Werkzeuge erweitert sowie flexibel um zusätzliche Domänen ergänzt werden kann. So setzt sich das Assistenzsystem aus einer benutzerfreundlichen GUI in Menü- und Kachelform und einer dynamischen Wissensbasis zum Management aller relevanten Daten und Metadaten zusammen. Außerdem besteht das Assistenzsystem neben einigen Konvertierungs-, Integrations-, Manipulationsmethoden v. a. aus den drei nachfolgenden wesentlichen Modulen: Erstens dem *matrixbasierten Modellierungsmodul* für die (teil-)automatisierte Modellierung von DSMs und für die computerunterstützte, grafische Modellierung von DMMs; zweitens dem *graphbasierten Visualisierungsmodul* zur Darstellung netzwerkartiger Relationen mit vielfältigen Graphlayouts; drittens dem *mathematischen Analysemodul* für die Anwendung mehrerer mathematischer Algorithmen sowie den beiden Modulen *Requirements Analysis* und *Cost Analysis*. Zudem unterstützt das entwickelte Konzept den Komplexitätsingenieur mit dem Assistenzsystem bei der Bewertung und der Verbesserung der Modelle komplexer Systeme durch Funktionalitäten zur Manipulation und Modifikation von Elementen und Relationen. Hierfür stehen dem Anwender das *Modifying Module* und das *Functional Analysis Module* zur Verfügung. Dadurch wird das holistische Komplexitätsmanagement in der PE ziel- und bedarfsgerecht unterstützt.

Durch die praktische **Anwendung des prototypisch implementierten Assistenzsystems** wird die Methodik zur Unterstützung des holistischen, matrixbasierten Komplexitätsmanagement in der IPE validiert. Als praktisches Beispiel für die Anwendung der drei wesentlichen Module des Assistenzsystems dient ein vereinfachtes Getriebe. Dabei werden die Nutzenpotentiale der beispielhaft eingesetzten Methoden präsentiert, die Anwendbarkeit der verschiedenen Funktionalitäten des Assistenzsystems gezeigt und die Zweckmäßigkeit des Konzepts nachvollziehbar dargestellt.

Die **kritische Diskussion des entwickelten Konzepts** und prototypisch implementierten Assistenzsystems zur Unterstützung des Anwenders beim holistischen Komplexitätsmanagement in der IPE hat dargelegt, dass dieses sowohl bei der Erstellung matrixbasierter Modelle komplexer Systeme als auch bei deren Analyse, Visualisierung und Bewertung sehr nützlich ist. So wird der Komplexitätsingenieur durch die weitgehend automatisierte Akquisition von Elementen und deren Relationen (z. B. aus CAD-Modellen) von der aufwendigen und fehleranfälligen manuellen Modellierung befreit. Zudem wird der Anwender durch computerunterstützte, grafische Methoden auch bei der Modellierung von DMMs deutlich entlastet. Darüber hinaus ermöglicht das Assistenzsystem je nach Aufgabe den Einsatz jeweils geeigneter Methoden zur mathematischen Analyse, graphbasierten Visualisierung und multikriteriellen Bewertung. Infolgedessen wird der Komplexitätsingenieur im Vergleich zu den bereits bestehenden Methoden und IT-Werkzeugen mit dem in dieser Arbeit entwickelten Konzept und implementierten Assistenzsystem beim holistischen Komplexitätsmanagement in der IPE besser unterstützt.

8.2 Ausblick und weiterführende Forschungsfragen

Dennoch besteht für Wissenschaftler und Anwender auch in Zukunft noch Forschungs- und Entwicklungsbedarf zur Verbesserung des Komplexitätsmanagements in der Entwicklung. Das in der vorliegenden Arbeit dargelegte Konzept des Assistenzsystems ist zwar in sich geschlossen, kann aber aufgrund des modularen Aufbaus **um weitere Module flexibel erweitert werden**. Daher sind im Folgenden ausgewählte Ansätze zur Weiterentwicklung sowie weiterführende Forschungsfragen kurz skizziert.

Aus den Ergebnissen der Forschungsarbeiten des Autors lässt sich die Erkenntnis ableiten, dass die Bewertung der Aufwand-/Nutzenverhältnisse für eine **computerunterstützte Akquisition** unterschiedlicher Produkt-, Prozess-, Wissens- und Organisationselemente und deren Relationen vom spezifischen Anwendungsfall und der jeweiligen Zielsetzung abhängig ist.

Daher ist das generische, computerunterstützte Akquisitionskonzept bisher lediglich für die in Bild 43 angeführten domänenspezifischen IT-Systeme und deren Modelle im Assistenzsystem implementiert (vgl. Bild 48). Dieses allgemeingültige Konzept kann in zukünftigen Forschungsarbeiten für die **Modellierung sowohl aller weiterer Produktelemente** (z. B. Funktionen, Wirkprinzipien) **als auch zusätzlicher Domänen** in der multidisziplinären Entwicklung angepasst und verbessert werden.

Neben der weitgehend automatisierten Akquisition von Elementen und Relationen zur Modellierung von DSMs besteht noch Forschungsbedarf zur Weiterentwicklung der automatischen Erkennung und **Modellierung von Relationen zwischen Elementen unterschiedlicher Domänen**. Zwar sind in dieser Arbeit vier nützliche Methoden für das automatische Erkennen von Relationen zur Modellierung von DMMs dargelegt (z. B. mit Hilfe von Metadaten), jedoch lassen sich durch diese Methoden (noch) nicht alle Relationen zwischen Domänen zuverlässig erkennen. Daher bedarf es weiterer Forschungsarbeiten zur Entwicklung von Methoden, die das automatische Identifizieren von Relationen zwischen verschiedenen Domänen geeignet bzw. noch besser unterstützen.

Die Methoden sowie implementierten Module des Assistenzsystems sind zwar so entwickelt, dass zwischen den Anwendungssystemen und dem Assistenzsystem grundsätzlich bidirektionale Schnittstellen bestehen, aber die Verwendung verschiedener IT-Werkzeuge in der PE führt i. d. R. zu Herausforderungen. So haben bspw. Modifikationen an einem CAD-Modell Änderungen auch in der Prozess- und Organisationsdomäne zur Folge, welche (erst) im Assistenzsystem identifiziert, analysiert und transparent dargestellt werden können. Daraus resultiert die Forschungsfrage, wie **das Assistenzsystem in die Anwendungssysteme integriert** werden kann, so dass dieses in den jeweiligen Anwendungssystemen (z. B. im CAD-System) den jeweiligen Domänenexperten (z. B. den Konstrukteur) in geeigneter Art und Weise unterstützt. Dadurch kann das Assistenzsystem im Sinne des Predictive Engineering nach WARTZACK [588] dem jeweiligen Experten z. B. direkt im Anwendungssystem darstellen, welche Änderungen und Auswirkungen die Modifikation eines bestimmten Elements (aufgrund welcher Relationen) zur Folge hat. Dabei werden die Auswirkungen auf Elemente nicht nur innerhalb einer Domäne, sondern auf alle Domänen angezeigt. Dafür sind die wesentlichen Konzepte und Funktionalitäten des Assistenzsystem in die einzelnen Anwendungssysteme zu integrieren, wobei das Assistenzsystem für das Management der Relationen zwischen den Elementen der Domänen der verschiedenen Anwendungssysteme dient.

Darauf aufbauende und somit weiterführende Forschungsbedarfe im Sinne des Predictive Engineering sind Ansätze für **Konzepte für ein proaktives Assistenzsystem**. Dies bedeutet, dass eine Weiterentwicklung des hier vorgestellten Assistenzsystems Komplexitätsingenieure und Anwender dabei unterstützen soll, proaktiv holistische Analysen vorzunehmen, um daraus Verbesserungsmaßnahmen ableiten zu können (d. h. z. B. Aufzeigen von Modularisierungs-, Konfigurations- und Wertanalysepotentialen). Zudem stellt sich daran anknüpfend auch die Frage, wie das Assistenzsystem weiterentwickelt werden kann, um zukünftig proaktiv den Anwendern vorzuschlagen, welche Elemente wie zu modifizieren sind, um bestimmte Anforderungen (besser) zu realisieren.

Zusammenfassend betrachtet ist in der vorliegenden Arbeit ein hilfreiches **Konzept zur Unterstützung des Komplexitätsingenieurs beim holistischen Komplexitätsmanagement in der PE** entwickelt und vorgestellt worden. Durch dessen Implementierung in ein Assistenzsystem und die beispielhafte Anwendung wurden sowohl die zuverlässige praktische Anwendbarkeit als auch der große Nutzen der Methodik dargestellt. Zudem bietet das Konzept einige Ansatzpunkte für weitere Forschungsarbeiten. Damit wird im Sinne des Eingangszitats aus Goethes Werk also auch dieses Thema nie in sich abgeschlossen sein, sondern sicherlich spannende und weitere nützliche Erkenntnisse bereithalten.

9 Summary and outlook

„Such a work is actually never finished, you have to declare it as finished, when you have done the most possible in terms of time and circumstances.”

[Citation translated from German by the author of this thesis]

JOHANN W. VON GOETHE (1749 – 1832)

GOETHE, one of the most famous German poets, related this quote to the elaboration of his play „Iphigenie auf Tauris”. Analogously, the author of this thesis also believes that the **research topic he is working on can never be completely finished**. Therefore, following the summarized conclusions in chapter 9.1, an outlook on further research needs is given in chapter 9.2.

9.1 Summary and conclusions

The higher complexity and the accompanying volatility, uncertainty, and ambiguity (cf. VUCA), the more important complexity management in product development will become. Therefore, the goal of this thesis is to develop a methodology to support **developers in holistic complexity management in product development** by providing appropriate methods for modelling, analysing, visualizing, and evaluating complex systems for decision making.

To realize this objective, a **fundamental understanding of the modelling of complex systems** is required. Thus, the basics of technical systems and complexity as well as the creation of models for complex systems from the perspective of product development are explained in chapter 2. This enables a better understanding of the challenges of complexity management in product development resulting from the multitude, variety, variability, and ambiguity of elements and relations of complex systems.

The **discussion of the state of the art in research and technology** has shown that domain-specific models are insufficient for holistic complexity management in product development. In contrast, model-based systems engineering, the modeling of complex systems by means of ontologies, and matrix-based modeling in the form of DSMs, DMMs, and MDMs are significantly more promising. Due to their universal suitability for the formal modeling of all elements and relations of complex systems, which can thus be processed by computers, the approaches for matrix-based complexity management are the focus of this thesis.

The **evaluation of existing methods and IT tools for matrix-based complexity management** has shown that there is a great need for action from a scientific and industrial point of view. For example, the transformation of models from the various domains into DSMs should be automated as far as possible to significantly reduce effort and susceptibility to errors. Moreover, there are hardly any methods for modeling cross-domain relations in DMMs and no computer support exists so far. Further need for action results from the necessity of suitable visualization options for different matrices, e.g., to show the changes of and their effects on certain system elements as a consequence of the modification of a certain system element. Numerous methods are available for the analysis of matrices, but these need to be adapted to the different specifics of the system models of the individual domains in product development.

The target of this thesis is therefore to **develop a concept that supports developers with holistic complexity management in product development** by providing suitable methods for modeling, analyzing, visualizing, and evaluating complex systems in each context. For this purpose, the IT architecture of the prototypically implemented assistance system is composed of modules, so that it can be modularly extended by further IT tools and flexibly supplemented by additional domains. Thus, the assistance system consists of a user-friendly GUI in the form of menus and tiles and of a dynamic knowledge base for managing all relevant data and metadata. In addition, the assistance system comprises several conversion, integration, and manipulation methods as well as the following three essential modules: First, the *matrix-based modeling module* for the (partially) automated modeling of DSMs and for the computer-aided, graphical modeling of DMMs; second, the *graph-based visualization module* for the representation of network-like relations with various graph layouts; third, the *mathematical analysis module* for the application of several mathematical algorithms as well as the two modules *Requirements Analysis* and *Cost Analysis*. Furthermore, the developed concept supports the complexity engineer by using the assistance system in the evaluation and the improvement of the models of complex systems by functionalities for the manipulation and modification of elements and relations. For this purpose, the *Modifying Module* and the *Functional Analysis Module* are available for the user. Thus, the holistic complexity management in product development is supported in a target- and demand-oriented manner.

The practical **application of the prototypically implemented assistance system** validates the methodology for supporting holistic, matrix-based complexity management in integrated product development. A

simplified gearbox serves as a practical example for the application of the three essential modules of the assistance system. The potential benefits of the methods used as examples are presented, the applicability of the different functionalities of the assistance system is shown and the usefulness of the concept is demonstrated in a comprehensible way.

The **critical discussion of the developed concept** and prototypically implemented assistance system for supporting the user in holistic complexity management in integrated product development has demonstrated that it is particularly useful both for the creation of matrix-based models of complex systems and for their analysis, visualization, and evaluation. For example, the largely automated acquisition of elements and their relations (e.g., from CAD models) renders time-consuming and error-prone manual modeling obsolete for the complexity engineer. In addition, computer-aided, graphical methods also significantly support the user when modeling DMMs. Moreover, depending on the task, the assistance system enables the use of methods that are respectively suitable for mathematical analysis, graph-based visualization, and multi-criteria evaluation. Hence, compared to the already existing methods and IT tools, the concept and implemented assistance system developed in this thesis supports the complexity engineer in holistic complexity management in integrated product development.

9.2 Outlook and further research questions

Nevertheless, there is still a need for future research and development for scientists and users to improve complexity management in development. Even though the concept of the assistance system presented in this paper is self-contained, it can be **flexibly extended by further modules** due to its modular structure. Therefore, selected approaches to further develop the assistance system presented here as well as further research questions will briefly be outlined in the following.

From the results of the author's research, it can be concluded that the evaluation of the effort/benefit ratios for **computer-aided acquisition** of different product, process, knowledge, and organizational elements and their relations depends on the specific use case and the respective objective. For this reason, the generic, computer-aided acquisition concept has so far only been implemented in the assistance system for the domain-specific IT systems and their models listed in figure 43 (cf. figure 48). This generally valid concept can be adapted and improved in future research for the modeling of all other product elements (e.g., functions, operating principles) and of further domains in multidisciplinary development.

In addition to the largely automated acquisition of elements and relations for modeling DSMs, research is needed to further develop the automatic detection and **modeling of relations between elements of different domains**. While four useful methods for automatically detecting relations for modeling DMMs are outlined in this paper (e.g., using metadata), not all relations between domains can (yet) be reliably detected by these methods. Therefore, further research is needed to develop methods that appropriately support the automatic identification of relations between different domains.

Although the methods and implemented modules of the assistance system have been developed in such a way that bidirectional interfaces exist between the application systems and the assistance system, the use of different IT tools in product development usually leads to challenges. For example, modifications to a CAD model also result in changes in the process and organizational domain, which can (only) be identified, analyzed, and transparently displayed in the assistance system. This leads to the research question of **how the assistance system can be integrated into application systems** so that it supports the respective domain expert (e.g., the designer) in the respective application systems (e.g., in the CAD system) in a suitable manner. In this way, the assistance system can, in the sense of Predictive Engineering according to WARTZACK [588], show the respective expert, e.g., directly in the application system, which changes, and effects (based on which relations) will result from the modification of a certain element. In this case, the effects on elements are displayed not only within one domain, but on all domains. For this purpose, the essential concepts and functionalities of the assistance system have to be integrated into the individual application system(s), whereby the assistance system serves to manage the relations between the elements of the domains of the various application systems.

Based on this, further research needs in the sense of Predictive Engineering are approaches for **concepts for a proactive assistance system**. This means that future assistance systems should enable complexity engineers and users to proactively perform holistic analyses in order to be able to derive improvement measures from them (e.g., identifying modularization, configuration, and value analysis potentials). In addition, this also raises the question of how the assistance system can be further developed to proactively suggest users in the future which elements should be modified in which way in order to (better) realize certain requirements.

In summary, a helpful **concept for supporting the complexity engineer in holistic complexity management in product development** has been developed and presented in the present thesis. Through its implementation in an assistance system and its exemplary application, both the sound practical applicability and the significant benefit of the methodology have been demonstrated. In addition, the concept offers numerous starting points for further research work. Thus, in the sense of the opening quotation from Goethe's work, this topic will never be self-contained, but will always hold exciting and further useful findings.

Anhang

A.1 Definition und Bestimmung von Begriffen

In diesem Anhang sind die für die vorliegende Arbeit grundlegenden und **relevanten Begriffsbestimmungen** in alphabetischer Abfolge zu finden. Dabei sind die folgenden Erläuterungen und Präzisierungen zu Begriffen aus den jeweils entsprechenden Kapiteln als Ergänzung und Vertiefung für ein besseres Verständnis zu verstehen.

Bauelement: Unter dem Überbegriff Bauelement werden in dieser Arbeit *Baugruppen* und *Einzelteile* subsummiert (s. [127], vgl. auch [227]).

Baugruppe: Eine **Baugruppe** ist ein Bauelement, das eine abgegrenzte bzw. geschlossene Gruppe (i. S. v. Modul) von mind. zwei Einzelteilen und/oder (Unter-) Baugruppen darstellt, die miteinander verbunden sind (s. [127], vgl. auch [227]).

Einzelteil: Ein Einzelteil ist ein Bauelement, das aus einem Werkstoff nach einer technischen Spezifikation gefertigt wird bzw. ist und nicht zerstörungsfrei zerlegt werden kann (s. [127], vgl. auch [227]).

Element (i. S. v. Systemelement): Der Elementbegriff im Rahmen der Systemtheorie sehr relativ zu betrachten. Je nach Zweck und Zielsetzung sind Elemente der relevanten Domänen, wie Produkt- und Prozessdomäne, entsprechend fein granular zu zerlegen. Für Kernphysiker und bspw. auch Werkstoffingenieure kann die Zerlegung von Produktelementen bis auf atomare Ebene (und z. T. noch granularer) zweckmäßig sein (z. B. für die Entwicklung von lithographischen Systemen). Dagegen erscheint in der PE im Maschinen- und Anlagenbau die Zerlegung der Systemelemente bis auf Ebene von *Einzelteilen* oder *Baugruppen* als sinnvoll.

Relation (ähnlich zu Beziehung, Interdependenz, Wechselwirkung, Zusammenhang): Mit Relationen insb. die Beziehungen zwischen Elementen eines Systems bezeichnet. Relationen sind neben den Elementen für die spezifischen Eigenschaften eines Systems verantwortlich.

A.2 Grundlagen und Erläuterungen zu VUCA

Einführung zu VUCA

Da VUCA auch immer mehr die PE prägt, werden im folgenden Abschnitt **wesentlichen Grundlagen zu VUCA für ein besseres Verständnis** erläutert. VUCA ist ein Akronym für die englischen Begriffe volatility ‚Volatilität‘ (Unbeständigkeit), uncertainty ‚Unsicherheit‘, complexity ‚Komplexität‘ und ambiguity ‚Mehrdeutigkeit‘.

Dabei wird zwar angeführt, dass es bereits früher VUCA-Weltperioden gab, aber das Ausmaß, die Intensität und das Tempo von VUCA in den letzten Jahren deutlich zugenommen haben [281]. Spätestens seit der Finanzkrise Ende der 2000er Jahre hat das Bewusstsein für die global vernetzte Geschäftswelt zugenommen [362]. Mit der rasanten Ausbreitung der neuartigen Atemwegserkrankung Covid-19 und der damit einhergehenden Pandemie wurde uns im Jahr 2020 erneut bewusst, wie vernetzt unsere Welt inzwischen ist – und wie komplex die Zusammenhänge darin.

In der heutigen Zeit sind Unternehmen immer mehr mit Rahmenbedingungen konfrontiert, die sich mit dem Akronym VUCA beschreiben lassen [122]. Dabei ist Digitalisierung einer der wesentlichen Treiber dieser VUCA-Welt, wobei mit VUCA in der Regel die breiten gesellschaftlichen Veränderungen beschrieben werden [346]. Neben diesem VUCA-Begriff und Digitalisierung beschreiben auch Schlagworte wie z. B. Agilität, Disruption und Nachhaltigkeit die gegenwärtige Situation in Wirtschaft und Wissenschaft [253], [24]. Als Konsequenz müssen auf der einen Seite Entscheider (z. B. CEOs) auch VUCA in ihren Entscheidungsprozessen mit einbeziehen [207], [217] und auf der anderen Seite ist es auch die **Aufgabe von Forschern dafür erforderliche Methoden, Modelle und IT-Werkzeuge zur Unterstützung dieser Entscheider** zu entwickeln. Daher beschäftigen sich sowohl Unternehmen als auch Forschungsorganisationen sehr intensiv mit diesen Themen.

Die **Wurzeln von VUCA** lassen sich im militärischen Bereich verorten und werden u. a. in einigen Publikationen ausführlich beschrieben (z. B. in [281], [346]). Der VUCA-Begriff wurde an einer Graduiertenschule für US-Elitegeneräle geprägt [281]. Da die US-Armee durch den Kampf gegen Terrororganisationen mit neuen Herausforderungen konfrontiert gewesen ist (z. B. keine Kommandostrukturen bei Terrororganisationen erkennbar, da gegnerisches Netzwerk komplexe Kommunikationsbeziehungen aufweist), wurde die Eliteausbildung hinsichtlich der neuen VUCA-Anforderungen gänzlich umgestellt [346].

Wie die inzwischen große Zahl an Publikationen zu VUCA verdeutlichen, wurden und werden wesentliche Erkenntnisse auf andere Bereiche in Wissenschaft und Wirtschaft übertragen [346], [217]. So organisieren sich auch Unternehmen immer mehr in Netzwerkstrukturen, um sich im Hinblick auf die sich verändernden Rahmenbedingungen schnell, flexibel und adaptiv anpassen zu können [346]. Um beispielsweise in der **VUCA-Dimension Komplexität** auf die Kontextsensibilität rasch und bedarfsgerecht reagieren oder sogar agieren zu können, vernetzten sich Unternehmen immer mehr über ihre eigenen Grenzen hinweg [346].

Erläuterung der VUCA-Dimensionen

Für ein besseres **Verständnis der vier VUCA-Dimensionen** werden diese im Kontext von Wirtschaft und Gesellschaft erläutert. Die folgenden Begriffserläuterungen basieren insb. auf [109], [362], [399] (s. für Details [45], [281], [356]):

- **Volatility** (dt. Volatilität, Unbeständigkeit): Volatilität beschreibt Änderungen mit hoher Geschwindigkeit, d. h. z. B. verschiedene unerwartete, außergewöhnliche und vielschichtige Schwankungen in Wirtschaft und Gesellschaft. Volatilität zeigt, wie schwierig es ist, diese Veränderungen mit Hilfe von Mustern zu prognostizieren bzw. zu identifizieren und zu beschreiben (im Gegensatz zu einem vorhersehbaren Verlauf der Wirtschaft in einer früheren, stabileren Welt). Erfahrungen und Best Practices aus der Vergangenheit sind daher nur noch bedingt geeignet, um Lösungen für die Gegenwart oder Zukunft zu finden.
- **Uncertainty** (dt. Unsicherheit): Unsicherheit beschreibt die Schwierigkeit der Vorhersehbarkeit der Zukunft. Unsicherheit zeigt also, wie schwierig es ist, Prognosen über die Zukunft zu machen. Während es in einer früheren, stabileren Welt durch statistische Regressionsmodelle möglich war, die Zukunft vorherzusagen, wird es heute immer schwieriger, zukünftige Entwicklungen zu extrapolieren und mit einer Wahrscheinlichkeitsverteilung zu assoziieren. Unsicherheit kann auch als mangelnde Klarheit angesehen werden, um einen (zukünftigen) Zustand angemessen zu bewerten.
- **Complexity** (dt. Komplexität): Komplexität beschreibt die Schwierigkeit in einer miteinander verbundenen bzw. vernetzten Welt Ursache und Wirkung miteinander zu verbinden, d. h. z. B., dass die Komplexität von Wirtschafts- und/oder Gesellschaftssystemen zu unscharfen Darstellungen der Ursache-Wirkungs-Beziehungen führt (d. h. lineare Kausalität stößt an ihre Grenzen), da diese

schwer festzustellen sind. Komplexität kann auch als ein Zustand angesehen werden, in der die Vernetzung von Elementen so hoch ist, dass dieselben Eingaben (zu unterschiedlichen Zeitpunkten) bei denselben externen Bedingungen zu unterschiedlichen Ausgaben bzw. Reaktionen des Systems führen können.

- **Ambiguity** (dt. Mehrdeutigkeit): Mehrdeutigkeit beschreibt, dass mehrere spezifische und unterschiedliche Interpretationen beispielsweise eines bestimmten Zustands zulässig sind (im Gegensatz zu Unbestimmtheit, die einen Zustand durch mangelnde Klarheit charakterisiert). Im Gegensatz in einer früheren, stabileren Welt gibt es heute kaum nur einen einzigen Weg für die Erklärung eines bestimmten Ereignisses. Stattdessen gibt es häufig mehrere gültige Alternativen, die sich in Zukunft als wahr bzw. richtig erweisen könnten. Die Antwort „es kommt darauf an“ löst klare „Ja/Nein-Antworten bzw. Lösungen“ ab. Mehrdeutigkeit kann auch als Resultat der drei zuvor beschriebenen Dimensionen angesehen werden. Mehrdeutigkeit ist auch dadurch charakterisiert, dass kausale Zusammenhänge oft unklar sind und daher die Bedeutung oder Interpretation einer Entscheidungs- oder Problemsituation nicht nach einer bestimmten Regel oder einem definierten Analyseprozess gelöst werden kann, wodurch häufig mehr als eine mögliche Entscheidung oder Lösung für eine Situation gleichermaßen gültig ist.

Da die vier Dimensionen jeweils Unterschiedliches ausdrücken und daher weder identisch noch untereinander austauschbar sind, lassen sich auch **verschiedene generische Handlungsstrategien** formulieren [346]. So strukturieren BENNETT & LEMOINE die vier VUCA-Dimensionen in einem Portfolio mit den zwei Achsen „Wissen über die Situation“ und „Vorhersehbarkeit der Ergebnisse möglicher Aktionen“ [45]. Je nach Situation werden Volatilität, Unsicherheit, Komplexität und Mehrdeutigkeit als unterschiedliche Kontexttypen betrachtet, in denen verschiedene handlungsorientierte Ansätze erfolgreich sind [362], [217].

Die folgende Darstellung in Bild 93 gibt einen **Überblick der verschiedenen Charakteristika, (Gestaltungs-)Ansätze und Beispiele für die vier VUCA-Dimensionen** in Anlehnung an [45], [362] und [346]. Dabei wird auf der einen Achse unterschieden, wie gut Ergebnisse von (Inter-)Aktionen (z. B. Entscheidungen) vorhergesagt werden können, und auf der anderen Achse differenziert, wie gut ist der Informationsstand über eine Situation (z. B. Entscheidungssituation) eingeschätzt werden kann [45], [346].

Wie gut können die Ergebnisse von (Inter-)Aktionen vorhergesagt werden?	gut	Komplexität Charakteristik: Situation mit vielen vernetzten Elementen; einige Informationen stehen zur Verfügung oder können vorhergesagt werden, aber der Umfang oder die Art der zu verarbeitenden Informationen kann sehr umfangreich sein. Ansatz: Restrukturieren; Spezialisten hinzuziehen oder entwickeln und Ressourcen aufbauen, die der Komplexität angemessen sind. Beispiel: Geschäftstätigkeit in vielen Ländern mit jeweils einzigartigen regulatorischen Rahmenbedingungen und kulturellen Werten und/oder Verbreitung mehrerer Produkte für mehrere Kundensegmente.	Volatilität Charakteristik: Herausforderung ist unerwartet oder instabil und kann von unbekannter Dauer sein, aber sie ist nicht unbedingt schwer zu verstehen; das Wissen darüber ist oft vorhanden. Ansatz: Puffer vorsehen und Ressourcen reservieren (z. B. Lagerbestände oder Talente vorhalten) und dabei auf adäquates Gleichgewicht zwischen Kosten und Risiken achten. Beispiel: Preisschwankungen, nachdem eine Naturkatastrophe Lieferanten vom Markt (vorübergehend) gedrängt hat.
		Mehrdeutigkeit Charakteristik: Situation, in der kausale Zusammenhänge völlig unklar sind; es gibt keine Präzedenzfälle; unbekannte und/oder konträre Informationen, die dennoch jeweils richtig oder auch falsch sein können. Ansatz: Experimentiere iterativ mit intensiven Feedbackschleifen, um herauszufinden, was in einem bestimmten Kontext funktioniert; um Ursache und Wirkung zu verstehen, Hypothesen aufstellen und testen; Experimente so gestalten, dass die gelernten Lektionen breit angewendet werden können. Beispiel: Markteintritt in unreife oder aufstrebende Märkte oder Einführung von Produkte außerhalb der eigenen Kernkompetenzen; starten einer Unternehmensgründung.	Unsicherheit Charakteristik: Situation mit grundsätzlich allen Informationen, die grundlegenden Ursachen und (Aus-) Wirkungen des Ereignisses sind im Wesentlichen bekannt, trotzdem hohe Ungewissheit bzw. Schwankung der Eintrittswahrscheinlichkeit der bzw. möglichen Veränderung. Ansatz: Investiere in Informationen und sammle, interpretiere und teile diese zur besseren Prognose; verwende Informationsanalysen, KI-Methoden und -Tools. Beispiel: Bevorstehende Produkteinführung eines Konkurrenten verunsichert die Zukunft des Unternehmens und des Marktes.
	schlecht	schlecht	
		Wie gut ist der Informationsstand über Situationen?	
			gut

Bild 93: Erläuterung der VUCA-Dimensionen (in Anlehnung an [45], [362] und [346])

Komplexität als Schlüsselkonzept von VUCA

Auch wenn die soeben angeführte Erläuterung der VUCA-Dimensionen zusammen mit der Darstellung in Bild 93 eine gute Herangehensweise ist, um beispielsweise Führungskräften pragmatische Erläuterungen und handlungsorientierte Ansätze für die vier VUCA-Dimensionen zu geben, liegt dieser Arbeit die in [362] beschriebene Art der Betrachtung des VUCA-Phänomens zugrunde. In dieser Publikation werden die vier Dimensionen nicht als unterschiedliche oder getrennte Kontexttypen angesehen, sondern **Komplexität zum Schlüsselkonzept bzw. Ausgangspunkt** gemacht und die anderen drei Dimensionen als Konsequenzen der Komplexität betrachtet [362]. Demnach wird das VUCA-Phänomen als eine Folge der natürlichen Welt angesehen, die als ein System mit hoher Komplexität konzipiert werden kann (vgl. Bild 94). Volatilität und Unsicherheit sind zwei beobachtbare Auswirkungen bzw. Folgen der Komplexität. In Entscheidungsfindungs- und Wahrnehmungsprozessen kann dies zu Mehrdeutigkeiten führen (vgl. Bild 94). Diese Erläuterungen basieren auch auf der Systemtheorie und der Komplexitätstheorie (vgl. Kapitel 2).

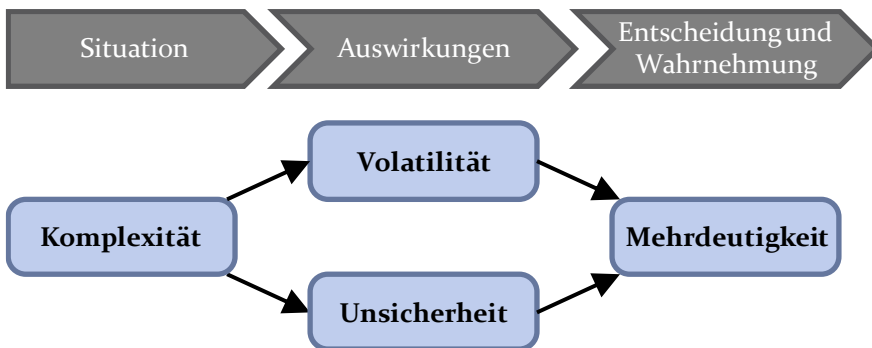


Bild 94: Interdependenzen zwischen den VUCA-Dimensionen (in Anlehnung an [362])

Nach MACK & KHARE können die Dimensionen des VUCA-Phänomens nicht nur als objektive Eigenschaften sondern auch als subjektive Attribute verstanden werden, die diesen von einzelnen Individuen zugeschrieben werden [362]. Die beiden Autoren vermuten, dass in Zeiten hoher Vernetzung und Dynamik und damit Komplexität traditionelle, einfache Denkmodelle, Entscheidungsregeln und Heuristiken, die in einer früheren, stabilen und traditionellen sowie mechanistischen Weltanschauung funktioniert haben, nicht mehr geeignet sind [362]. Hinter dieser früheren Weltanschauung steht das menschliche Bedürfnis nach einfachen, rationalen Ursache-Wirkungs-Denkmodellen, die Menschen dabei unterstützen, Entscheidungen zu treffen und mit der Umwelt umzugehen. Die heutige

Welt ist hingegen komplexer (und/oder zumindest das Verständnis der Komplexität der Welt wird immer komplexer, d. h. besseres Verständnis wie Dinge miteinander verbunden sind) und daher besteht der **Bedarf nach neuartigen, differenzierten Modellen und Ansätzen in Bezug auf Kognition, Bewertung und Handeln für das Bewältigen dieser komplexen Systeme** und Situationen für Manager [362], [217].

Erläuterungen zu VUCA Prime

Die zuvor beschriebenen Herausforderungen einer zunehmend volatilen, unsicheren, komplexen und mehrdeutigen Welt können nach JOHANSEN mit Hilfe einer **positiven Lesart des Akronyms VUCA** gelöst werden: Vision (dt. Vision), Understanding (dt. Verstehen bzw. Verständnis), Clarity (dt. Klarheit) and Agility (dt. Agilität bzw. Beweglichkeit) [281]. Nach seiner Auffassung lassen sich die besten Vorgesetzten in der VUCA-Welt durch diejenigen Führungsfähigkeiten charakterisieren, die er als VUCA Prime bezeichnet [356]. Daher werden diese vier VUCA-Prime Fähigkeiten und Fertigkeiten im Rahmen von Führungskräfteentwicklungsschulungen vermittelt [399]. Im Folgenden werden diese vier VUCA-Prime Fähigkeiten v. a. nach [109] skizziert:

- **Vision**, um **Volatilität** zu begegnen: Im Hinblick auf die Vision geht es darum, ein klares Zukunftsbild für eine Organisation aufzuzeigen, um die wichtigsten Prioritäten sowie die langfristige Richtung vorzugeben. Dabei ist das Stiften von Sinn durch die Vision wichtiger als die konkreten Pläne für deren Erreichung. Die Unternehmensvision ist in eine Mission sowie in Strategien und Ziele zu übersetzen bzw. zu kaskadieren. Dabei ist die Vision allen Stakeholdern der Organisation unter Berücksichtigung deren verschiedenen Präferenzen anschaulich zu vermitteln (z. B. durch Vorleben durch die Führungskräfte), um auch ein Gefühl von Vertrauen, Identifikation und Gemeinschaft zu erreichen (z. B. durch Orientierung an gemeinsamen Werten). Zur Vermeidung von Unklarheit und Verwirrung ist auf die Änderung der Vision zu verzichten.
- **Verstehen**, um **Unsicherheit** zu begegnen: In Bezug auf Verstehen geht es um das Verstehen der Vision, der individuellen Rolle in der Organisation sowie den eigenen Beitrag zum Erfolg der Vision. Dabei ist die Förderung von regelmäßiger, offener Kommunikation, aktivem Zuhören, empathischer Beziehungsfähigkeiten besonders wichtig, um Verständnis (z. B. für den Kontext) zu bekommen und zu geben (vgl. [421]). Durch klare Verantwortlichkeiten, verständliche Erwartungen sowie gesetzte Ziele und Grenzen, werden

wesentliche Grundlagen gelegt, um auch bei unsicheren Bedingungen richtig reagieren zu können. Für ein besseres Verständnis der VUCA-Welt hilft sowohl die Nutzung von diversen Informationsquellen (z. B. von Kunden, Lieferanten, Wettbewerbern) als auch die Entwicklung von Kooperationen innerhalb (z. B. abteilungsübergreifende Teams) und außerhalb (z. B. mit Kooperationspartnern) von Organisationen bei.

- **Klarheit**, um **Komplexität** zu begegnen: Mit Klarheit ist weder Einfachheit noch Gewissheit gemeint. Mit Klarheit ist mehr die klare Richtung und weniger die detaillierte Beschreibung des Endzustands gemeint. Klarheit wird nicht durch vereinfachen der Themen, sondern vielmehr durch die Themen einfach machen erreicht. Dazu sind alle relevanten Aspekte transparent zu machen, verständlich darzustellen und unnötige Details zu entfernen bzw. zu priorisieren (z. B. durch Systemdenken). Klarheit meint auch die Fähigkeit mangelndes Wissen auf einem Gebiet einzugestehen und die Bereitschaft sich neues Wissen anzueignen (z. B. durch Kommunikation). Durch Schaffung von Klarheit ist es Vorgesetzten möglich, auf die wichtigen Themen fokussiert zu reagieren und mit den verfügbaren und relevanten Informationen klare Entscheidungen zu treffen und strukturierte Handlungen abzuleiten. Dies bedeutet zum Beispiel das Identifizieren aller in, einer komplexen Situation beteiligten, Personen, ihrer Herausforderungen und Absichten sowie kausalen Beziehungen.
- **Agilität**, um **Mehrdeutigkeit** zu begegnen: Bei Agilität geht es darum, Herausforderung zu meistern, indem Personen oder Organisationen vernetzt, flexibel und beweglich sind und sich flexibel und rasch ändern können, um sich verändernden Rahmenbedingungen schnell anpassen können. Hierfür sollten Strategien Elemente zur Förderung einer Unternehmenskultur enthalten, die positiv und offen für wiederkehrenden Wandel und technologische Innovationen (z. B. aus dem Digitalisierungsbereich) ist. Im Gegensatz zu traditionellen Organisationsformen, die in der langfristigen, linearen Planung verankert sind, können daher agile Organisationen mit dem Tempo des Wandels der Welt Schritt halten und sich verändern (vgl. [24]). Zudem sind Prozesse mit kurzen Feedbackzyklen erforderlich, die es ermöglichen, zu experimentieren, zu scheitern, schnell zu lernen und zu verbessern (s. auch [430], [474]).

Zusammenfassend wird der Zusammenhang bzw. Übergang zwischen VUCA auf der einen und VUCA Prime auf der anderen Seite wird durch die folgende Darstellung anschaulich skizziert (vgl. Bild 95). So symbolisiert das Ufer unten die Erfahrungen aus der Vergangenheit und gegenwärtige (Rahmen-)Bedingungen sowie der Horizont oben den zukünftigen Sollzustand [356]. Der Fluss dazwischen versinnbildlicht den Übergang zur VUCA Prime Welt, wobei es keinen eindeutigen **Übergang von VUCA zu VUCA Prime** gibt [356].

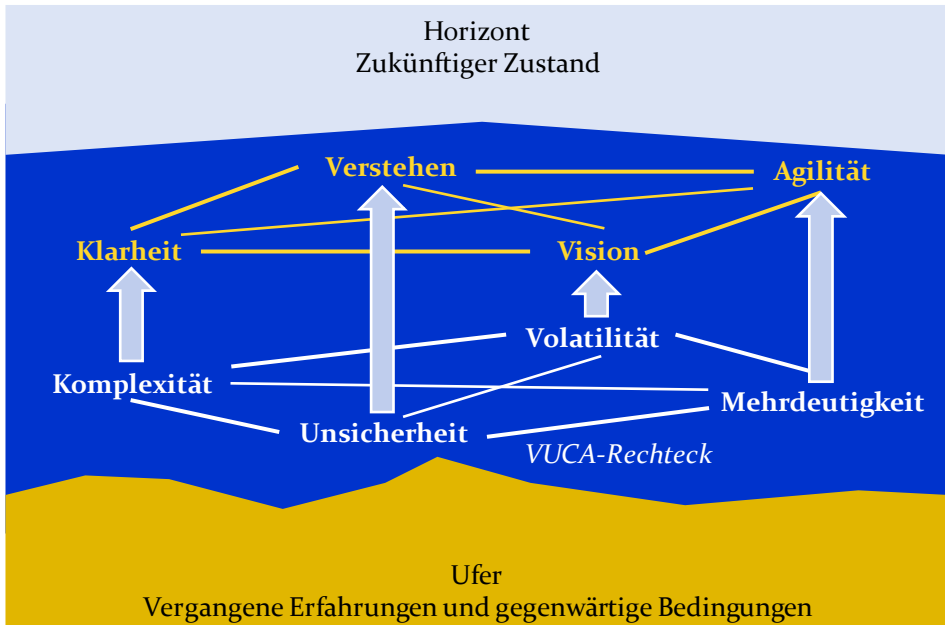


Bild 95: Übergang von VUCA zu VUCA Prime (in Anlehnung an [356])

Kompetenzen des Komplexitätsingenieurs in einer VUCA-Welt

In Kapitel 1.4 sind wichtige **Kompetenzen des Komplexitätsingenieurs** zusammenfassend erläutert wurden (vgl. Bild 96). Diese werden in diesem Abschnitt aus Perspektive der VUCA-Welt ergänzt bzw. aus Grundlage der Besonderheiten und Spezifika der VUCA-Welt abgeleitet.

Die einleitend beschriebene VUCA-Welt führt u. a. nach [282] zu grundlegenden Veränderungen der (Zusammen-)Arbeit von Führungskräften und Mitarbeiter in Organisationen. Deswegen sind das Wissen und die Kompetenzen, die früher in Organisationen benötigt wurden, um weitere erforderliche Fähigkeiten und Fertigkeiten ergänzt werden [282].

Wie zuvor im Abschnitt „Erläuterungen zu VUCA Prime“ kur dargelegt, sind die VUCA-Prime Fähigkeiten, die auch mit dem Akronym VUCA (Prime) bezeichnet werden (also Vision, Verstehen, Klarheit und Agilität) für den Übergang von VUCA zu VUCA Prime von Bedeutung. Für die Rolle des Komplexitätsingenieurs lassen sich aus Perspektive der **VUCA-Welt** folgende, ergänzende Kompetenzen ableiten:

Vision: Für den Umgang mit volatilen und sprunghaften Rahmenbedingungen sind Problemlösungskompetenzen gefragt, um die volatilen Entwicklungen antizipieren und/oder darauf bei Bedarf reagieren zu können. Hierfür benötigt der Komplexitätsingenieur gute Kenntnisse zentraler Problemlösungsmethoden, um diese kontextbezogen anzuwenden. Hierbei sollte das konstruktionsmethodisch geläufige „generierende Vorgehen“, das v. a. für prinzipiell neue Lösungen geeignet ist, durch das oft eingesetzte „korrigierende Vorgehen“, das kaum Zeitaufwand erfordert, ergänzt werden [151] (i. S. v. adaptive Führungskompetenz). Darüber hinaus ist es zielführend, wenn der Komplexitätsingenieur mittels seines guten Problemverständnisses zusammen mit den Akuteren der relevanten Fachbereiche (vgl. Kapitel 1.4) gemeinsame, langfristige Probleme definiert und sich auf diese weitgehend unabhängig von volatilen Veränderungen fokussiert.

- **Unsicherheit:** Ein wichtiger Grund für die flexible Anwendung von Methoden nach [151] ist die Ergebnisunsicherheit des Entwicklungsprozesses. Kritisches Denkvermögen ist beim Umgang mit Unsicherheit eine hilfreiche Kompetenz des Komplexitätsingenieurs [282]. Daher sind von diesem je nach Entwicklungsfortschritt und Komplexität der Aufgabenstellung geeignete Methoden und IT-Werkzeuge einzusetzen (z. B. aus dem Risikomanagement die FMEA-Methode und aus dem Komplexitätsmanagement die DSM-Methode), mit welchen Erkenntnissen gewonnen werden, um ein besseres Verständnis bzw. Verstehen des komplexen Systems zu erreichen. Zudem müssen die Komplexitätsingenieure Modellierungs- und Analyseunsicherheiten richtig beurteilen können – ähnlich wie auch andere Rollen der PE wie z. B. Daten-, Messverfahrens-, Informationsunsicherheiten [151]. Da jede Bewertung und infolgedessen auch jede Entscheidungsfindung unsicher ist [P25] (s. auch [5], [589]), bedarf es auch guter Bewertungs- und Entscheidungskompetenzen, um auch ohne sichere Bewertungs- bzw. Entscheidungsgrundlage entschieden agieren zu können.

- **Klarheit:** Für das Managen von Komplexität ist diese – sofern möglich und soweit möglich – zu vermeiden, zu reduzieren oder/und zu beherrschen. Hierfür sind wesentliche Methoden und IT-Werkzeuge des Komplexitätsmanagements zu beherrschen (s. Kapitel 2 und Kapitel 5), die vom Komplexitätsingenieur auch beim „Denken in System“ unterstützen. Dabei brauchen Komplexitätsingenieure auf der einen Seite Gewissenhaftigkeit und Detailverliebtheit, um alle relevanten Elemente und Relationen eines Systems zu modellieren, und auf der anderen Seite Abstraktionsvermögen und Fokussierung, um nicht für die Zielsetzung relevante Systemelemente und -relationen zu vernachlässigen. Zur Kompetenz des Systemdenkens gehört auch das Differenzieren in Systemgrenzen im Hinblick auf diversen Domänen (horizontale Perspektive) und in Bezug auf unterschiedliche hierarchische Ebenen, wie z. B. Produktarchitektur- und Prozesshierarchieebenen (vertikale Perspektive). Dabei kann kritisches Hinterfragen von Hierarchien und Systemgrenzen zu mehr Klarheit und einer besseren „Navigation“ durch komplexe Systeme führen. Zur übersichtlichen Veranschaulichung und zielgerichteten Interpretation der vielen Interdependenzen sind vertiefende Kenntnisse der Graphen- und Netzwerktheorie und darauf aufbauender Methoden und IT-Werkzeuge erforderlich.
- **Agilität:** Da sich komplexe Systeme u. a. durch den Entwicklungsfortschritt dynamisch verändern, muss der Komplexitätsingenieur agil sein. Dabei helfen agile Vorgehensweisen, Methoden und Werkzeuge, wie diese z. B. aus der Scrum-Philosophie bekannt sind (vgl. [379], [24]). Durch positive und proaktive Veränderungskompetenz und Bereitschaft zum häufigen Wandel gelingt es dem Komplexitätsingenieur die (bereits erstellten) Modelle auf die sich dynamisch verändernden komplexen Systeme zweckmäßig und adaptiv anzupassen. Aufgrund dieser ausgeprägten Agilität und Anpassungsfähigkeit richtet sich der Komplexitätsingenieur flexibel an den Anforderungen aus und kommt rasch zu verbesserten Bewertungsergebnissen.

Außerdem soll der Komplexitätsingenieur v. a. auch **Resilienz** besitzen, um widerstandsfähig für die Herausforderungen in einer VUCA-Welt zu sein. So sind der flexible Umgang in unsicheren Situationen in einem volatilen Umfeld, das Treffen von Entscheidungen unter ambivalenten Rahmenbedingungen sowie das Akzeptieren von Fehlschlägen als eine mögliche Option Verhaltensaspekte, um sich resilient in einer durch VUCA geprägten Welt zu bestehen (vgl. auch [253], [606]).

Kompetenz des systemischen Denkens und Handelns

Nach der Ableitung von Kompetenzen für den Komplexitätsingenieur aus VUCA-Welt-Perspektive wird in diesem Abschnitt auf die wohl wichtigste Kompetenz eingegangen. Diese ist nach der Auffassung des Autors dieser Arbeit Kompetenz für **systemisches Denken und Handeln**.

Systemisches Denken ist nach SENGE eine von fünf Disziplinen, die für lernende Teams und Organisationen erforderlich ist und die Basis für alle anderen Disziplinen bildet [504]. In [466] wird diese fünfte Disziplin des systemischen Denkens bzw. **Systemdenkens** als Kompetenz zum Erkennen ganzheitlicher Strukturen und wechselseitiger Abhängigkeiten von Systemen verstanden. Dazu gehörten auch das Wissen und Bewusstsein, dass die eigenen, gegenwärtigen Aktionen (z. B. Modifikationen von Elementen, Entscheidungen) spätere – sowohl positive als auch negative – potenzielle Auswirkungen auf Systemelemente zur Folge haben können. Außerdem umfasst dies auch das Verständnis bzw. Verstehen sowie das (gezielte) Nutzen der (System-)Struktur [466]. Das Denken in Systemen und das Bewusstsein für Systeme hat zur Folge, dass zum einen nichts für sich allein bzw. isoliert betrachtet werden kann und zum anderen zwischen verschiedenen Elementen eines Systems Relationen bestehen können, die nicht auf dem ersten Blick offenkundig sind [466]. Im Gegensatz zu „linearem“ bzw. geradlinigen Denken, wie z. B., dass ein Effekt von A auf B das Ergebnis C zur Folge hat, kann beim systemischen Denken der Effekt von A auf B auch D und E beeinflussen und das Ergebnis F zu Folge haben (und dabei kann F eventuell auch erst mit einem gewissen zeitlichen Verzug eintreten) [466]. Vor diesem Hintergrund ist das systemische Denken (bzw. das Systemdenken) die bedeutendste Kompetenz des Komplexitätsingenieurs.

Neben der Kompetenz des systemischen Denkens und Handels, also ein bestimmtes System als Ganzes betreffend zu betrachten (in dieser Arbeit auch als ganzheitlich oder holistisch Systemdenken bezeichnet), muss ein guter Komplexitätsingenieur auch ausreichende Kenntnisse der und in den einzelnen Fachdisziplinen – i. S. v. **grobem Spezialistenwissen** – haben (vgl. Anhang A.3). Dies liegt u. a. daran, dass ohne „Überblickswissen“ in den relevanten, einzelnen Systemdomänen ein Systemdenken über bzw. zwischen diesen Domänen vorhandenen Interdependenzen kaum vorstellbar ist. Zudem müssen sich nach BÖCHER, der tiefgreifend wissenschaftlich mit der Natur von Systemen und Systemdenken auseinandergesetzt hat, Systemdenken und Detailwissen bzw. Spezialistenwissen wechselseitig befruchten [67].

In ähnlicher Art und Weise diskutieren auch EHRENSPIEL & MEERKAMM das **Zusammenwirken von Spezialisten und Integratoren** in [151] und insb.

in [149] und [150]. So erläutern die beiden Autoren ausführlich auf der einen Seite die Notwendigkeit der steigenden Spezialisierung in Wissenschaft und Wirtschaft in Folge der (exponentiell) zunehmenden Informations- und Wissensflut und auf der anderen Seite die Erfordernis und auch die Schwierigkeit die verschiedenen Fachbereiche in ein holistisches System zu integrieren [151] (vgl. auch Dualismus i. S. v. zweier Grundprinzipien, die sich ergänzen).

Spezialisten sind in ihren jeweiligen Domänen Experten, die (fast) alles wissen und auch können (sollten). Dies können in der Produktdomäne bspw. Simulations- und Versuchsexperten sein. Da selbst in diesen beiden Spezialgebieten der Simulation und des Versuches wiederum eine zunehmende Spezialisierung erforderlich ist, gibt es auch immer mehr spezialisierte Experten in den zwei Fachgebieten, wie z. B. Simulationsexperten für hydrodynamische Simulationen (evtl. auch noch für das Teilgebiet Wirbel und/oder ggf. auch noch für bestimmte Bedingungen) in spezialisierten Teams. Da ein Entwickler nicht alles wissen und v. a. können kann, gibt es Spezialisten für alle relevanten Fachgebiete in einem Unternehmen. Aufgrund der Entwicklung von Fachsprachen für die jeweiligen Fachgebieten können sich Spezialisten untereinander oft nur sehr schwer verstehen (vgl. auch [151]). Darüber hinaus haben Spezialisten i. d. R. keinen Blick und oft auch kein Interesse für das Gesamtsystem [150] (z. B. dem gesamten Produkt oder sogar der holistischen Entwicklung mit all seinen Teildomänen).

Demgegenüber steht die von EHRENSPIEL & MEERKAMM als sog. **Integrator** bezeichnete Rolle, die i. d. R. von integrativen Persönlichkeiten ausgeübt wird [149], [150]. Diese Integratoren sind aufgrund ihrer fachlichen, methodischen, sozialen und personalen Kompetenzen in der Lage, mit den Spezialisten erfolgreich zusammen zu arbeiten und ggf. Kompromisse bei gegensätzlichen Perspektiven und Schnittstellenthemen zu finden, um so zum Erfolg des Unternehmens signifikant beizutragen [149], [150]. Dabei haben sich sowohl in Forschungs- als auch Praxisarbeiten gezeigt, dass der Erkenntnisgewinn und der Ursprung von Ideen für spätere Innovationen immer weniger in Arbeiten eines Spezialgebiets sondern vielmehr in interdisziplinären Teamarbeiten entsteht (vgl. auch [151]). Daher fordern EHRENSPIEL & MEERKAMM in [151] und v. a. in [149] und [150] integrierende Methoden, Techniken und Formen der Zusammenarbeit zum „Brücken bauen zwischen den Spezialgebieten – ganz im Sinne eines systemischen Denkens und Handelns. Daher haben die erforderlichen Kompetenzen des Integrators eine äußerst große Schnittmenge mit denen eines Komplexitätsingenieurs.

A.3 Kompetenzen des Komplexitätsingenieurs

Die in Kapitel 1.4 vorgestellte Rolle des Komplexitätsingenieurs wird in diesem Anhang konkretisiert. Dazu werden wesentliche **Kompetenzen des Komplexitätsingenieurs** skizziert.

Gemäß den bereits erwähnten Studien (vgl. [347], [348]) des World Economic Forums ist komplexes Problemlösen bzw. das **Lösen komplexer Probleme** eine seit Jahren der wichtigsten und daher gefragtesten Fähigkeiten von Menschen in Wissenschaft und Wirtschaft. Darüber hinaus wird in dieser Studie u. a. auch auf die hohe Relevanz der folgenden Fähigkeiten aufmerksam gemacht [348]:

- Fähigkeiten zur geeigneten Anwendung und Manipulation von Informationen bei der Problemlösung;
- Fähigkeiten zur zielgerichteten Lösung von Problemen mit mathematischen Beziehungen;
- Fähigkeiten zur Erarbeitung eines Verständnisses, wie ein System funktionieren soll und wie sich Änderungen der Randbedingungen, der Betriebsbedingungen und der Umweltbedingungen auf die Outputs bzw. Ergebnisse auswirken;
- Fähigkeiten zur Berücksichtigung der relativen Kosten und des Nutzens potenzieller Maßnahmen, um die am besten geeignete auszuwählen;
- Fähigkeiten zur Ermittlung von Maßnahmen bzw. Indikatoren der Systemleistung und der erforderlichen Maßnahmen zur Korrektur oder Verbesserung dieser Leistung unter (verhältnismäßiger) Rücksichtnahme auf die (weiteren) Systemzielen.

Daher benötigt der Komplexitätsingenieur für seine Aufgaben neben den auch für Entwickler grundsätzlich wichtigen **Kompetenzen** (vgl. die Arbeiten des Autors [P15], [P36] sowie die umfangreichen Forschungsarbeiten in [335] und [336]) einige ergänzende Fähigkeiten und Fertigkeiten. Die wesentlichen Kompetenzen des Komplexitätsingenieurs sind in Bild 96 überblicksartig in den vier Kategorien Fach-, Methoden-, Sozialkompetenzen und personale Kompetenzen dargestellt.

In größeren Unternehmen ist es sinnvoll, auch zwischen den Rollen Komplexitätsingenieur und Komplexitätsmanager zu differenzieren. Dabei ist die Rolle des **Komplexitätsmanagers** der des Komplexitätsingenieurs grundsätzlich sehr ähnlich, jedoch strategischer und übergeordneter (vgl. analog zu Wissensmanager und -ingenieur in [P7], [P20]).

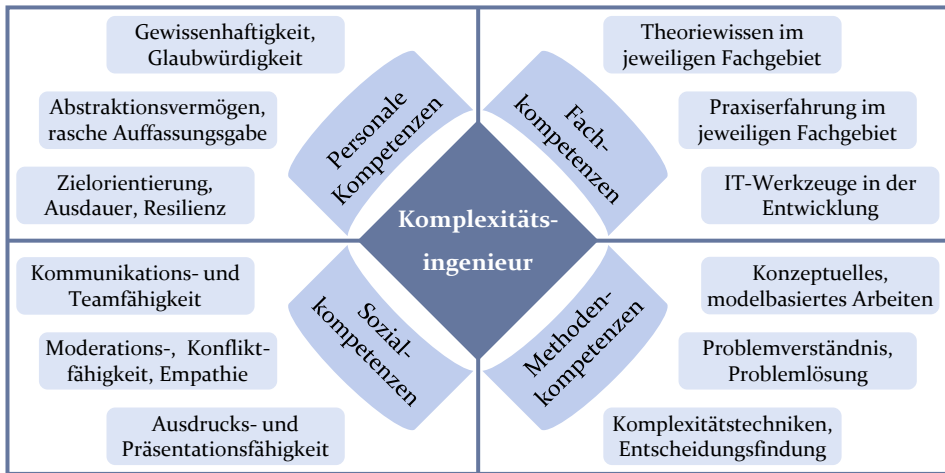


Bild 96: Wesentliche Kompetenzen des Komplexitätsingenieurs

Die in Bild 96 für die vier Kompetenzkategorien jeweils angeführten drei Kompetenzen sind als Beispiele für ein **Kompetenzprofil** eines Komplexitätsingenieurs anzusehen. Im Hinblick auf die interdisziplinären Aufgaben der Rolle ist sowohl Theoriewissen als auch Praxiserfahrung in den diversen Fachgebieten erforderlich (s. Kapitel 1.4). Zudem muss der Komplexitätsingenieur mit den wesentlichen IT-Systemen der relevanten Disziplinen vertraut sein. Für das Lösen von schwierigen Problemen sowie dem Managen komplexer Systeme ist ein ausgeprägtes Verständnis des Problems erforderlich (vgl. Zitat „Ein richtig erkanntes Problem ist schon die halbe Lösung“ von CHARLES F. KETTERING (1876-1958)). Dabei helfen Komplexitätsingenieuren Methodenkompetenzen zum konzeptuellen, modelbasierten und systematischen Arbeiten sowie zum Modellieren, Analysieren, Visualisieren und Bewerten von komplexen Systemen, um bestmögliche Entscheidungen für die jeweilige Situation zu finden (z. B. mit Design Thinking). Mit Blick auf die Sozialkompetenzen zeichnet sich ein guter Komplexitätsingenieur aufgrund der durch Interdisziplinarität geprägten Zusammenarbeit mit mehreren Fachbereichen durch Team-, Kommunikations-, Moderations-, Konfliktfähigkeit und Empathie aus. Da Komplexitätsingenieure häufig komplexe Probleme und nicht immer einfache Lösungen vorstellen, sind hervorragende Ausdrucks- und Präsentationsfähigkeiten essenziell. Zu den wichtigen personalen Kompetenzen eines Komplexitätsingenieurs gehören Glaubwürdigkeit, Gewissenhaftigkeit und Vertrauen, da die oft umfangreichen Modelle der komplexen Systeme nicht von allen anderen Teammitgliedern bis ins letzte Detail überprüft werden können. Aufgrund der komplexen Aufgabenstellungen ist auch ein gutes Abstraktionsvermögen sowie eine rasche Auffassungsgabe hilfreich, um

sich auf die für die jeweilige, kontextspezifische Entwicklungssituation relevanten Systemelemente und -relationen zu fokussieren (vgl. auch [421]). Da Komplexitätsingenieure i. d. R. mit sehr komplexen Problemstellungen konfrontiert werden, benötigen diese auch Zielorientierung, Beharrlichkeit und Ausdauer. Besonders auch in Folge der VUCA-Rahmenbedingungen (vgl. Kapitel 1 und Anhang A.2) wird v. a. Resilienz – oft als Zukunftskompetenz deklariert [253] – immer wichtiger (s. Anhang A.2 für vertiefende Erläuterungen zu diesen und zusätzlichen Kompetenzen v. a. aus Perspektive der VUCA-Welt und aus den Ausführungen von EHRENSPIEL & MEERKAMM sowie [606]).

Dabei ist das Zitat von DAVID J. SNOWDEN & MARY E. BOONE (s. Kapitel 6) zu berücksichtigen, das deutlich macht, dass kluge Führungskräfte – wie v. a. auch Komplexitätsingenieure – ihren Ansatz an die Komplexität der Umstände anpassen, mit denen sie konfrontiert sind [514].

A.4 Rechenbeispiel zum Anwendungsszenario Rohr

Nachfolgend wird das in Kapitel 1.5 beschriebene, beispielhafte Anwendungsszenario des einseitig, fest eingespannten Rohrs (vgl. Bild 6) mittels eines Rechenbeispiels mit beispielhaften Werten sowie unter Verwendung mathematischer Gleichung konkretisiert. Dadurch werden Relationen zwischen den Produktelementen selbst sowie auch zwischen ausgewählten Produkt- und Spezifikationselementen zur besseren **Nachvollziehbarkeit** mathematisch beschrieben und somit veranschaulicht.

Spezifikations- bzw. Anforderungsdomäne

Einige, beispielhafte **Anforderungen** für das Anwendungsszenario sind in Tabelle 3 zu finden. Darin wird die Spezifikationsdomäne mit SZ sowie die dazugehörigen Anforderungen mit ANF, Nutzungsbedingungen mit NUT und Umgebungsbedingungen mit UMG abgekürzt. Zu den wesentlichen Elementen der Spezifikationsdomäne gehören u. a. fertigungstechnische Anforderungen (z. B. Schweißbarkeit, Gießbarkeit, Kalt- und Warmumformbarkeit), ökologische Anforderungen (z. B. Umweltbelastung bei der Produktherstellung und Produktnutzung, Recyclingfähigkeit) und ökonomische Anforderungen (z. B. Verfügbarkeit und Kosten von Materialien, Fertigungsmaschinen, Infrastruktur, Arbeitsplätze).

Tabelle 3: Ausgewählte Anforderungen der Spezifikationsdomäne für das Rohr

Dom.	Typ	Bezeichnung	Symbol	Einheit
SZ	ANF	Maximaler Preis	€	EUR
SZ	ANF	Maximales Gewicht	F_G	N
SZ	ANF	Erforderliche Wandstärke	$s_{min-max}$	mm
SZ	ANF	Mindestquerschnittsfläche	A_{min}	mm ²
SZ	NUT	Maximal wirkende Kraft	F_N	N
SZ	UMG	Temperatur	ϑ	°C

Produktdomäne

Wie in Bild 6 skizziert, ist das Rohr einseitig links fest eingespannt und hat als **Produktmerkmale** (MER) den Innendurchmesser d_i und den Außendurchmesser d_a sowie die Länge l . Zudem hat das Rohr einen Werkstoff W , der sich grundsätzlich durch seine bestimmte stoffliche bzw. chemische Zusammensetzung und seinen kristallinen Aufbau charakterisieren lässt.

Dadurch ergeben sich die Materialkennwerte, wie bspw. die Dichte ρ und das Elastizitätsmodul E . Diese beeinflussen auch die wesentliche Werkstoffeigenschaften, wie bspw. mechanische, elektrische und magnetische Eigenschaften des Rohrs (für Details s. z. B. [270]).

Die geometrischen **Produktmerkmale** werden wie in Tabelle 4 festgelegt. So beträgt die Länge $l=1.000,0\text{ mm}$, der Außendurchmesser $d_a=50,0\text{ mm}$ und der Innendurchmesser $d_i=46,0\text{ mm}$. Durch die Festlegung des Werkstoffs W , wie z. B. die Wahl des Edelstahl X5CrNi8-10, wird die chemische Zusammensetzung in Massen-% nach der DIN EN 10088-3:2014-12 [128] des Werkstoffs festgelegt. Durch diese definierte, stoffliche Zusammensetzung und den kristallinen Aufbau ergeben sich die Materialkennwerte, wie bspw. die Dichte $\rho=0,008\text{ g/mm}^3$ und das von der Temperatur abhängige Elastizitätsmodul $E=200.000,0\text{ N/mm}^2$ (bei einer Temperatur $\vartheta = 20^\circ\text{C}$) des gewählten austenitischen Chrom-Nickel-Stahls (für Details s. z. B. [270]). Das Werkstoffnummern-System gemäß der Norm DIN EN 10027 Teil 2 [129] ordnet dem Werkstoff X5CrNi8-10 die Werkstoffnummer 1.4301 zu. Dieser Austenit-Stahl ist u. a. aufgrund seiner hohen Korrosionsbeständigkeit und sehr guten Weiterverarbeitung in den unterschiedlichsten Anwendungsbereichen (z. B. Maschinen- und Anlagenbau, Fahrzeugbau, Medizintechnik) stark verbreitet und gilt daher als Standard Chrom-Nickel-Stahl.

Tabelle 4: Produktmerkmale mit beispielhaften Ausprägungen einiger Elemente

Do.	Typ	Bezeichnung	Sym- bol	Ausprägung	Einheit
PD	MER	Innendurchmesser	d_i	46,0	mm
PD	MER	Außendurchmesser	d_a	50,0	mm
PD	MER	Länge	l	1.000,0	mm
PD	MER	Dichte des Werkstoffs	ρ	0,008	g/mm^3
PD	MER	E-Modul des Werkstoffs	E	200.000,0	N/mm^2

Aus der Festlegung dieser Produktmerkmale ergeben sich mit Hilfe von mathematischen Zusammenhängen u. a. die nachfolgenden erläuterten **Produkteigenschaften** (EIG). Sofern der Innen- und Außendurchmesser des Rohrs als Produktmerkmale festgelegt werden, ergibt sich die Wanddicke bzw. -stärke s des Rohrs (als Produkteigenschaft). Allein durch die Auslegung der beiden Durchmesser lassen sich auch die Querschnittsfläche A , die zwei Flächenträgheitsmomente I_y und I_z , das Widerstandsmoment W_y

sowie das Volumen V des Rohrs berechnen. Aus dem Volumen und der Dichte des Werkstoffs resultiert die Masse m . Die Dehnsteifigkeit DS (bzw. EA) ist das Ergebnis der Multiplikation des E-Moduls mit der Querschnittsfläche und die Biegesteifigkeit BS (bzw. EI) das Resultat aus dem E-Modul mit dem Flächenträgheitsmoment. Diese werden in Tabelle 5 überblicksartig dargestellt und nachfolgend kurz mathematisch hergeleitet.

Tabelle 5: Produkteigenschaften mit beispielhaften Ausprägungen einiger Elemente

Do.	Typ	Bezeichnung	Sym-bol	Ausprägung	Einheit
PD	EIG	Biegesteifigkeit	BS	$17.401,9 \cdot 10^6$	Nmm ²
PD	EIG	Dehnsteifigkeit	DS	$60.318,6 \cdot 10^3$	N
PD	EIG	Masse	m	2.382,9	g
PD	EIG	Widerstandsmoment	W_y	3.480,4	mm ³
PD	EIG	Flächenträgheitsmoment	I_y / I_z	87.009,6	mm ⁴
PD	EIG	Volumen	V	301.592,9	mm ³
PD	EIG	Querschnittsfläche	A	301,6	mm ²
PD	EIG	Wandstärke	s	2,0	mm

In diesem Beispiel wird der Außen- und Innendurchmesser d_a bzw. d_i als Produktmerkmal definiert, so dass mit der Gleichung (8) die Wandstärke s des Rohrs berechnet werden kann. Alternativ hätten als Produktmerkmale bspw. auch der Außendurchmesser d_a und die Wandstärke s definiert und somit der Innendurchmesser d_i als Produkteigenschaft berechnet werden können.

$$s = \frac{d_a - d_i}{2} = \frac{50,0 \text{ mm} - 46,0 \text{ mm}}{2} = 2,0 \text{ mm} \quad (8)$$

Die Querschnittsfläche A des Rohrs resultiert aus der Auslegung der beiden Durchmesser d_a und d_i des Rohrs (s. Gleichung (9)). Zur Berechnung des Volumens des Rohrs werden neben den beiden Durchmessern auch noch der Wert für die Rohrlänge l erforderlich (s. Gleichung (10)).

$$A = \pi * \left[\left(\frac{d_a}{2} \right)^2 - \left(\frac{d_i}{2} \right)^2 \right] = \pi * \left[\left(\frac{50,0 \text{ mm}}{2} \right)^2 - \left(\frac{46,0 \text{ mm}}{2} \right)^2 \right] \quad (9)$$

$$= 301,6 \text{ mm}^2$$

$$V = \pi * l * \left[\left(\frac{d_a}{2} \right)^2 - \left(\frac{d_i}{2} \right)^2 \right] = l * A \quad (10)$$

$$= 1.000,0 \text{ mm} * 301,6 \text{ mm}^2 = 301.592,9 \text{ mm}^3$$

Die beiden Flächenträgheitsmomente I_y und I_z des Rohrs (vgl. Bild 6 für das Koordinatensystem) sind das Ergebnis aus der Definition der Werte für die Durchmesser d_a und d_i (s. Gleichung (11) und [225]).

$$I_y = I_z = \frac{\pi}{4} * \left[\left(\frac{d_a}{2} \right)^4 - \left(\frac{d_i}{2} \right)^4 \right] = \frac{\pi}{4} * \left[\left(\frac{50 \text{ mm}}{2} \right)^4 - \left(\frac{46 \text{ mm}}{2} \right)^4 \right] \quad (11)$$

$$= 87.009,6 \text{ mm}^4$$

Das Widerstandsmoment W wird grundsätzlich aus dem Quotienten von Flächenträgheitsmoment I und dem maximalen Abstand einer Randfaser $z_{\max}$ berechnet [225]. Daher wird für die Berechnung des Widerstandsmoments W_y (bzw. W_z) für den Querschnitt des Rohrs das Trägheitsmoment I_y (bzw. I_z) der maximale Abstand $z_{\max}$ gleich $d_a/2$ (d. h. gleich dem Außenradius des Rohrs) gesetzt (s. Gleichung (12) und vgl. [225]).

$$W_y = \frac{I_y}{\left(\frac{d_a}{2} \right)} = \frac{\pi[(d_a/2)^4 - (d_i/2)^4]}{4 * (d_a/2)} = \frac{87.009,6 \text{ mm}^4}{50,0 \text{ mm}/2} \quad (12)$$

$$= 3.480,4 \text{ mm}^3$$

Anschließend wird mit dem berechneten Volumen V und der durch die Auswahl des Werkstoffs Edelstahl X5CrNi8-10 festgelegten Dichte ρ die Masse m (in der Einheit Kilogramm) des Rohrs nach Gleichung (13) berechnet. Dabei ist die Masse eines Körpers (z. B. des Rohrs) unabhängig vom Ort überall gleich.

$$m = V * \rho = 301.592,9 \text{ mm}^3 * 0,008 \frac{\text{g}}{\text{mm}^3} = 2.382,9 \text{ g} \quad (13)$$

Im Gegensatz dazu beschreibt das Gewicht bzw. die Gewichtskraft F_G (in der Einheit Newton) wie stark ein Körper an einem bestimmten Ort durch die ortsabhängige Gravitation (z. B. die Erdanziehung) angezogen wird (bildlich gesprochen nach „unten“ gezogen wird). Die Gewichtskraft F_G ist

das Resultat aus der Multiplikation der Masse m mit der Schwerebeschleunigung bzw. Erdbeschleunigung g (s. Gleichung (14)), welche ortsabhängig ist und daher je nach Ort auf der Erde einen etwas anderen Wert hat. Daher ist die Schwerebeschleunigung prinzipiell ein Element des Typs (UMG). Auf der Erdoberfläche wird für die Schwerebeschleunigung i. d. R. der Näherungswert $g = 9,81 \text{ m/s}^2$ angenommen. Deshalb ist im Anwendungsbeispiel das maximale Gewicht des Rohrs bzw. die maximale Gewichtskraft F_G , die auf das Rohr wirkt, als eine Anforderung angeführt (vgl. Tabelle 3).

$$F_G = m * g = 2.382,9 \text{ g} * 9,81 \frac{\text{m}}{\text{s}^2} = 23,4 \text{ N} \quad (14)$$

Die Steifigkeit eines Körpers (z. B. Rohr) beschreibt den Widerstand gegen elastische Verformung durch eine Kraft (z. B. der Gewichtskraft F_G) oder ein Moment. So lassen sich je nach Beanspruchungsart unterschiedliche Arten von Steifigkeiten, wie Dehn-, Biege-, Schub- und Torsionssteifigkeit, voneinander unterscheiden. Dabei ist die Dehnsteifigkeit DS das Ergebnis des Produkts aus dem Elastizitätsmodul E in Richtung der Belastung und der Querschnittsfläche A des Rohrs senkrecht zur Richtung der Belastung (s. Gleichung (15)). Die Biegesteifigkeit BS berechnet sich aus der Multiplikation des Elastizitätsmoduls E mit dem Flächenträgheitsmoment I der Querschnittsfläche an der Angriffsstelle des (Biege-)Moments (s. Gleichung (16)). Tiefergehende Erläuterungen zur Schub- und Torsionssteifigkeit sind [225] zu entnehmen.

$$\begin{aligned} DS &= E * A = 200.000,0 \frac{\text{N}}{\text{mm}^2} * 301,6 \text{ mm}^2 = 60.318.578,9 \text{ N} \\ &= 60.318,6 * 10^3 \text{ N} \end{aligned} \quad (15)$$

$$\begin{aligned} BS &= E * I = 200.000,0 \frac{\text{N}}{\text{mm}^2} * 87.009,6 \text{ mm}^4 \\ &= 17.401.910.026,80 \text{ Nmm}^2 = 17.401,9 * 10^6 \text{ Nmm}^2 \end{aligned} \quad (16)$$

Das **Produktverhalten** (VER) bzw. i. d. R. die Verhaltensweisen des Produkts sind die Folge der Interdependenzen zwischen bestimmten Produktmerkmalen und/oder Produkteigenschaften auf der einen und den Umgebungs- und Nutzungsbedingungen auf der anderen Seite. So resultiert z. B. eine Dehnung (i. S. v. Längenänderung) des Rohrs an der Stelle $x = l$ aus der Einwirkung einer äußeren Kraft bzw. Last F_N als eine Nutzungsbedingung bei einer bestimmten Dehnsteifigkeit des Rohrs. Diese Verformung ist auch vom Eigengewicht sowie von der Temperatur ϑ als eine Umgebungsbedingung abhängig. Die Durchbiegung $\omega_{\max}$ und Neigung $\omega'_{\max}$ des Rohrs an der

Stelle $x = l$ hängen neben der Temperatur ϑ und der Kraft bzw. Last F_N auch von der Biegesteifigkeit BS (bzw. EI) des Rohrs ab. Drei ausgewählte Produktverhaltensweisen des Rohrs sind in Tabelle 6 beispielhaft dargestellt und werden nachfolgend kurz mathematisch hergeleitet.

Tabelle 6: Produktverhalten mit beispielhaften Ausprägungen einiger Elemente

Do.	Typ	Bezeichnung	Sym- bol	Ausprägung	Einheit
PD	VER	Neigung	ω'_{max}	0,029	-
PD	VER	Durchbiegung	ω_{max}	19,2	mm
PD	VER	Längsdehnung	ε	0,000017	-

Das elastische Rohr mit konstanter Querschnittsfläche A ist, wie in Bild 6 skizziert, einseitig fest eingespannt und hat im unbelasteten Zustand die Länge l . In der Kontinuumsmechanik wird unter der Verformung (synonym für Deformation bzw. Verzerrung) eines Körpers die Änderung seiner Form aufgrund der Einwirkung einer (äußeren) Einzelkraft verstanden. Dabei kann generell zwischen der Änderung der Länge (d. h. Längsdehnung) und der Änderung des Winkels (d. h. Scherung) unterschieden werden. So lässt sich die Längsdehnung ε des Rohrs an der Stelle $x = l$ aus der Einwirkung einer äußeren Kraft F_N an derselben Stelle ermitteln (s. Gleichung (17)). Die Dehnung ist prinzipiell auch von der Temperatur ϑ bzw. T (d. h. einer Umgebungsbedingung) abhängig. Für die Berechnung der Dehnung ε wird jedoch vereinfachend von einer gleichförmigen Temperaturänderung (d. h. $\Delta T = \text{konstant}$) und von einer konstanten Dehnsteifigkeit DS ausgegangen.

$$\varepsilon = \frac{\Delta l}{l} = \frac{F_N}{E * A} = \frac{1.000,0 \text{ N}}{200.000,0 \frac{\text{N}}{\text{mm}^2} * 301,6 \text{ mm}^2} = 0,000017 \quad (17)$$

Die Durchbiegung ω_{max} (auch mit f symbolisiert) und Neigung ω'_{max} des Rohrs in z -Richtung an der Stelle $x = l$ sind sowohl von der Biegesteifigkeit BS des Rohrs als auch von der Nutzungsbedingung F_N (d. h. der an der Stelle $x = l$ wirkenden Kraft) und der Umgebungsbedingung Temperatur ϑ abhängig (vgl. Bild 6). Dabei ist die sog. Biegelinie der Verlauf $\omega(x)$ der Durchbiegung des Rohrs [225]. Für die vereinfachte Berechnung der Durchbiegung (s. Gleichung (18)) und Neigung (s. Gleichung (19)) wird von einer gleichförmigen Temperaturänderung (d. h. $\Delta T = \text{konstant}$) und von einer konstanten Dehnsteifigkeit DS ausgegangen.

$$\omega_{max} = \frac{F_N * l^3}{3 * E * I} = \frac{1.000,0 \text{ N} * (1.000,0 \text{ mm})^3}{3 * 200.000,0 \frac{\text{N}}{\text{mm}^2} * 87.009,6 \text{ mm}^4} \quad (18)$$

$$= 19,2 \text{ mm}$$

$$\omega'_{max} = \frac{F_N * l^2}{2 * E * I} = \frac{1.000,0 \text{ N} * (1.000,0 \text{ mm})^2}{2 * 200.000,0 \frac{\text{N}}{\text{mm}^2} * 87.009,6 \text{ mm}^4} \quad (19)$$

$$= 0,029$$

Diese erläuterten Merkmale, Eigenschaften und Verhaltensweisen des vereinfachten Rohrs sowie die auch angeführten Anforderungen, Nutzungs- und Umgebungsbedingungen des beispielhaften Anwendungsszenarios stellen nur eine kleine Auswahl dar und sind daher nicht vollständig. Im Rechenbeispiel zum Anwendungsszenario Rohr sind die wesentlichen mathematischen Zusammenhänge mit Werten ausführlich erläutert. Selbst für dieses (scheinbar) sehr einfache Rohr lassen sich zahlreiche **weitere Merkmale, Eigenschaften und Verhaltensweisen** berechnen und beschreiben. So lassen sich u. a. folgende Produktmerkmale, -eigenschaften und -verhaltensweisen ergänzen (vgl. für Details [189], [270]):

- Mechanische Elemente: Streckgrenze, Bruchdehnung, -festigkeit, Härte, Warmfestigkeit
- Elektrische und magnetische Elemente: spezifischer Widerstand, Leitfähigkeit, Isolationsfähigkeit, thermoelektrische Eigenschaften
- Chemisch-physikalische Elemente: thermische Ausdehnung, Schmelzpunkt, Korrosionsbeständigkeit, Oxidationsbeständigkeit

Für die **Berechnung** von weiteren Merkmalen, Eigenschaften und Verhaltensweisen des Rohrs sei abschließend auf die entsprechende Literatur verwiesen, wie z. B. [226], [225], [189].

A.5 Begriffe, Prinzipien der Systemtheorie und eine Klassifikationssystematik von Systemen

Grundlegende Fachbegriffe zur Systemtheorie

Mit dem Begriff System sind eine Vielzahl weiterer **Begriffe** eng verknüpft. Dazu gehören nach HUBKA v. a. die Begriffe Menge, Zweck, Verhalten, Funktion, Struktur, Umgebung (Umwelt), In- und Output, Eigenschaften, Ausprägung, Zustand, Modell, Art, Aufgabentyp, Prozess, Transformation sowie Relation mit Eigenschaften und Arten [268], die auch für die vorliegende Arbeit von Relevanz sind. Daher werden diese Begriffe nach HUBKA definiert bzw. die entsprechenden Erläuterungen aus [268] im Folgenden teils wörtlich und verkürzt wiedergegeben, so dass an dieser Stelle für tiefergehende Erläuterungen insb. auf [268] (und auch [267]) verwiesen wird.

Eine **Menge** ist eine Zusammenfassung bestimmter Elemente zu einem Ganzen. Es kann zwischen einer endlichen und einer unendlichen Menge differenziert werden. Sind alle Elemente der Menge N in der Menge M enthalten, so ist N Teil-Menge von M . Die Gesamtheit M minus N (d. h. alle nicht zu N gehörenden Elemente von M) ist ihr Komplement. Die Vereinigungsmenge von M und N umfasst alle Elemente, die entweder zu M oder zu N gehören. Der Durchschnitt von M und N enthält alle Elemente, die sowohl zu M als auch zu N gehören.

Ein künstliches (technisches) System dient einem bestimmten **Zweck**, der durch ein bestimmtes Zielsystem beschrieben wird. Dabei besteht ein Zielsystem i. d. R. eine Menge von Zielen mit wechselseitigen Relationen. Ziele können in hierarchischer Beziehung i. S. v. Ober- und Unterzielen stehen. Unterziele können übergeordnete Ziele konkretisieren und als Mittel zur Erreichung von Oberzielen dienen.

Verhalten kann als Menge der zeitlich aufeinanderfolgenden Zustände eines Systems definiert werden. Die Zielsetzung für die Schaffung eines technischen Systems ist ein bestimmtes Verhalten.

Oft wird das zweckgebundene Verhalten eines Systems als seine **Funktion** (d. h. teleologische Funktion) bezeichnet. Dabei beschreibt die Funktion eine stabile, gewünschte Wirkungsfähigkeit, die aber nicht immer durch das Verhalten des Systems sichergestellt ist, da sich ein System auch nicht richtig verhalten kann. Folglich kann der Begriff Verhalten sinngemäß breiter als Funktion angewendet werden, da die Funktion eng mit der gewünschten Wirkung verbunden ist.

Die **Struktur** bezeichnet allgemein die innere Ordnung oder den Aufbau des Systems. Somit ist die Struktur die Menge der Elemente eines Systems und die Menge der, die Elemente miteinander verbindenden, Relationen.

Zur **Beziehung zwischen Struktur und Verhalten** lässt sich festhalten, dass das Verhalten des Systems durch die Struktur des Systems gegeben ist. Ein (relativ) geschlossenes System mit einer gegebenen Struktur hat grundsätzlich nur ein einziges Verhalten, da die Struktur das Verhalten determiniert. Das Verhalten legt die Struktur nicht eindeutig fest. Dasselbe Verhältnis (Funktion) kann durch unterschiedliche Strukturen verwirklicht werden. Dabei sind die Struktur und das Verhalten die wichtigsten, charakteristischen Eigenschaften eines Systems.

Die **Umgebung** (Umwelt) eines Systems S ist grundsätzlich alles, was nicht in das betreffende System einbezogen wird. Im Allgemeinen bezeichnet die Umgebung alles, was nicht zu der Gesamtheit aller in Betracht gezogenen Systeme gehört.

Ein **Input** (Eingangsgröße, Eingabe) stellt die äußere Relation zwischen Umwelt und System dar. In Abhängigkeit von der Art des Systems kann der Input Wirkung, Kopplung oder Zustandsgröße des Operanden sein, wobei Operanden allgemein alle Gegenstände der Änderung bezeichnen.

Ein **Output** (Ausgangsgröße, Ausgabe) stellt die äußere Relation zwischen System und Umwelt dar. In Abhängigkeit von der Art des Systems kann der Output Wirkung, Kopplung oder Zustandsgröße des Operanden sein.

Grundsätzlich lässt sich zu **In- und Output bemerken**, dass diese das Einzige sind, was ein System mit der Umwelt verbindet. Dabei können In- und Outputs sowohl nicht erwünschte Kopplungen (d. h. Störungen) als auch erwünschte Kopplungen stofflichen, energetischen oder informativen Charakters sein. Zudem sei angemerkt, dass manche Autoren In- und Output als Eingangselemente (Rezeptoren) respektive Ausgangselemente (Effektoren) auffassen, durch welche die Stimuli in das System i. S. v. Einwirkung der Umgebung kommen bzw. die Reaktionen (Antworten) als die Einwirkung des Systems auf die Umgebung eintreten. Für andere Autoren sind In- und Output, was vom System aufgenommen bzw. abgegeben wird.

Jedes System mit seinen Elementen und Relationen besitzt **Eigenschaften**, die dieses System innehat und die es genauer definieren, wie z. B. Volumen, Masse, Biegesteifigkeit und auch Eignung zur Herstellung, zum Transport, zur Umweltverträglichkeit sowie insb. auch die Fähigkeit, etwas zu tun, d. h. das Verhalten, wie z. B. Neigung, Durchbiegung, Längsdehnung (vgl. Kapitel 1.5 und Anhang A.4).

Dabei kann die **Ausprägung** (z. B. Wert, Qualität, Güter) einer Eigenschaft verschieden sein. So hat die Produkteigenschaft Masse m des Rohrs die Ausprägung 2,382,9 g (g als Einheit der Ausprägung) im Anwendungsszenario (vgl. Kapitel 1.5 und Anhang A.4). Bei einer Modifikation einer der der beiden oder beider Ausprägungen für das Volumen V oder der Dichte ρ verändert sich auch die Produkteigenschaft der Masse des Rohrs.

Die Gesamtheit der Ausprägungen aller Eigenschaften des Systems (z. B. Rohrsystem) zu einem bestimmten Zeitpunkt wird als der **Zustand** des Systems bezeichnet. Der Zustand kann anschaulich als ein „Vektor“ mit den einzelnen Eigenschaften als Vektorkomponenten dargestellt werden. Zwei Systemzustände können entweder gleich oder unterschiedlich (als Differenz bezeichnet) sein. Eine Differenz besteht entweder zwischen den Zuständen eines Systems oder beim Übergang des Systems von einem in einen anderen Zustand. Bei einem Übergang können die Unterschiede entweder differentiell (d. h. bei kontinuierlichen Übergängen) oder diskret sein.

Das **Modell** des Systems veranschaulicht die diversen Definitionen zu den Systembegriffen und deren jeweiligen Beziehungen. In Kapitel 1.4 sind detaillierte Erläuterungen zur Modelldefinition und zum Modellverständnis zu finden.

In Abhängigkeit verschiedener Kriterien lassen sich zahlreiche **Arten** von Systemen voneinander unterscheiden. Einige, wichtige Klassen sind in der Klassifikationssystematik von Systemen im folgenden Abschnitt zu finden.

Mit Systemen sind nach [268] die folgenden drei charakteristischen **Aufgabentypen** verbunden:

- Bei der Synthese werden das Verhalten und weitere Anforderungen als gegeben angesehen, so dass die Struktur, die die erforderlichen Anforderungen erfüllt, gesucht werden muss
- Bei der Analyse wird dagegen die Struktur als gegeben aufgefasst, so dass das Verhalten gesucht werden muss
- Beim sog. „Black Box Problem“ ist das System, dessen Struktur nicht oder nur teilweise bekannt ist, gegeben, so dass das Verhalten und möglicherweise auch die Struktur gesucht werden muss.

Allgemein bedeutet **Prozess** ein Vorgang, ein Geschehen, etwas sich im Zeitablauf Veränderndes. Neben natürlichen Prozessen erschafft der Mensch künstliche Prozesse, um die für ihn notwendigen oder erwünschten Änderungen (z. B. zur Bedürfnisbefriedigung) zustande bringen. Für den Menschen ist der Prozess der gezielten Änderung bestimmter Objekte lebenswichtig.

HUBKA bezeichnet diese künstlichen Prozesse als **Transformationen** (d. h. Operationen), in denen bestimmte Eigenschaften des Operanden (s. Definition In- und Output) dem Ziel (Zweck) entsprechende Änderungen erfahren, wodurch der gewünschte Zustand des Operanden erreicht wird; dabei ist oft die Mitwirkung des Menschen und meist technischer Mittel von Bedeutung [268]. Eine Änderung ist die Folge bestimmter Einwirkungen, die v. a. auf physikalischen, chemischen und biologischen Phänomenen basieren und mit einer Regel (z. B. Anleitung, Algorithmus, Technologie) beschrieben werden. An der Ausübung von erforderlichen Einwirkungen sind generell einige Operatoren beteiligt. Der Output der Operatoren sind ihre Auswirkungen auf den Operanden. Grundsätzlich wird der Begriff „Wirkung“ benutzt, ohne dabei zu unterscheiden, ob es sich um Aus- oder Einwirkung handelt. Die Wirkung der Operatoren verläuft als Stoff-, Energie- und Informationsfluss. Das allgemeine Modell des Transformationsprozesses nach HUBKA, welches in [268] detailliert erläutert wird, ist in Bild 97 dargestellt. ROPOHL ergänzt diesen Transformationsprozess um Raum und Zeit, da im Rahmen des Transformationsprozesses auch räumliche und zeitliche Faktoren zu berücksichtigen sind [458].

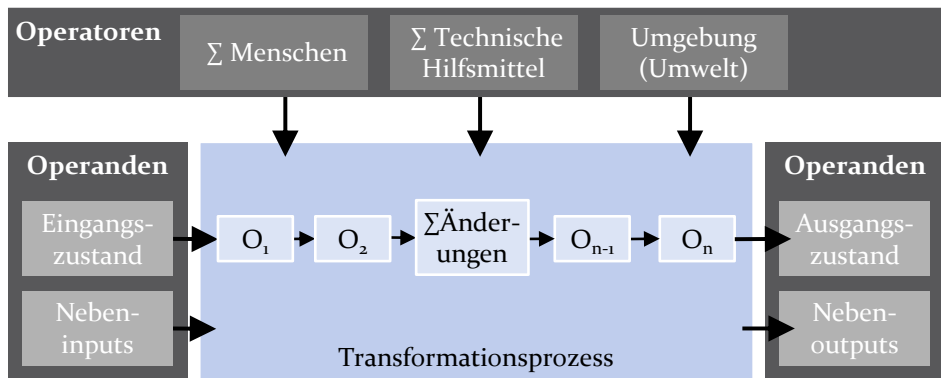


Bild 97: Vereinfachtes Modell des Transformationsprozesses (in Anlehnung an [268])

Relationen (synonym für Beziehung) in Systemen bezeichnen das sachlich oder bedeutungshafte voneinander abhängen oder aufeinander wirken von zwei oder mehreren Objekten (synonym für Elementen), die von abstrakter oder konkreter Art sein können. Relationen verbinden einzelne Elemente zu verschiedenen Systemen. Im Rahmen der (Produkt-)Entwicklung sind v. a. objektive, (mathematisch) exakte, beschreibbare Relationen wichtig, wobei der Autor dieser Arbeit auch die Berücksichtigung von subjektiven, (nur) qualitativ beschreibbaren Relationen als erforderlich empfindet.

Grundsätzlich kann die **Eigenschaft einer Relation** reflexiv, symmetrisch oder transitiv sein. Sofern alle der drei folgenden Eigenschaften bei einer Relation erfüllt sind, wird diese als Äquivalenzrelation bezeichnet:

- Reflexivität liegt vor, falls jedes Objekt äquivalent zu sich selbst ist
- Symmetrie liegt vor, falls ein Objekt einem zweiten äquivalent ist und so auch das zweite dem ersten
- Transitivität liegt vor, falls zwei Objekte einem dritten äquivalent sind und so auch beider untereinander äquivalent sind

Abschließend führt HUBKA noch Reihe von **Arten von Relationen** auf, die im Folgenden gekürzt wiedergegeben werden:

- Ähnlichkeit: Relationen mit gewissen gemeinsamen Eigenschaften, wobei das Ausmaß von vollständiger Gleichheit (Identität) bis zur entferntesten Ähnlichkeit geht. Dabei kann es sich um Ähnlichkeit z. B. in funktioneller oder struktureller Hinsicht handeln.
- Analogie: Relationen stimmen hinsichtlich wesentlicher Eigenschaften überein, v. a. in der Funktion und Struktur bei Systemen (vgl. Ähnlichkeit). Oft wird Analogie i. S. v. Ähnlichkeit gebraucht.
- Homomorphie: Relation zwischen zwei Systemen, bei der von jedem Bestandteil und jeder Relation des einen Systems auf einen Bestandteil und Relation im zweiten System geschlossen werden kann, jedoch nicht umgekehrt.
- Isomorphie: Relation zwischen zwei Systemen, bei der sich jedem Bestandteil des einen ein bestimmter Bestandteil des anderen Systems zuordnen lässt und umgekehrt (vgl. Symmetrie). Ferner, wenn jeder Relation zwischen den entsprechenden Bestandteilen die gleiche Relation des anderen Systems entspricht und umgekehrt.
- Äquivalenz des Objekts: Objekte heißen (sind) äquivalent, wenn zwischen ihnen eine Äquivalenzrelation besteht. Ähnlich wird von Äquivalenz bei Mengen oder Aussagen gesprochen.
- Identität: Relation zwischen Objekten (Sachverhalten), welche hinsichtlich gewisser Merkmale übereinstimmen. Bei absoluter (relativer) Identität müssen alle (einige) Merkmale übereinstimmen. Im Gegensatz zur Äquivalenz enthält die Identität nur die Reflexivität.
- Mathematische Funktion: Relation als mathematische Funktionen, d. h. eine gesetzmäßige Abhängigkeit von veränderlichen Variablen (vgl. $y = f(x)$). Mathematische Funktionen drücken eine genaue Relation zwischen x und y aus (vgl. determinierten Zusammenhang).

- Kausalität: Eine zweistellige asymmetrische Relation zwischen Ursache und Wirkung, wobei die Ursache der Wirkung vorausgeht. Dabei besteht eine strenge (d. h. determinierte, wenn-dann) oder abgeschwächte Form von kausaler Beziehung. Eine kausale Kette entsteht, wenn die Wirkung erneut als Ursache weiterer Wirkungen auftritt.
- Kopplung: Eine Kopplung liegt vor, wenn bestimmte Outputs des Elements (Systems) zugleich als Inputs eines Elementes (Systems) auftreten. Die Kopplung kann entweder in Serie, parallel, als Rückkopplung oder als Kombination angeordnet werden
- Ziel-Mittel Relation: Zweistellige asymmetrische Relation zwischen Zielsystem (oder Zweck, Aufgabe) und Mittel zur Verwirklichung der Ziele.
- Räumliche Relation: Relationen zur Beschreibung von gegenseitiger bzw. referenzieller Lage bzw. Anordnung der Elemente (der Relation) im Raum. Eine Wissenschaftsdisziplin, die sich mit räumlichen Relationen beschäftigt, ist z. B. die Topologie.
- Logische Relation: Logische Relationen sind Beziehungen zwischen Elementen, wie z. B. „A ist kleiner als B“ und „B ist größer als C“. Bekannte zweistellige, logische Verknüpfungen sind die Konjunktion (AND-Verknüpfung) und die Disjunktion (OR-Verknüpfung).
- Zeitliche Relation: Zeitliche Relationen werden in Abhängigkeit vom Zeitablauf, z. B. von Prozessen (Transformationen) entlang des Zeitstrahls angeordnet.

Abschließend lässt sich dieser Abschnitt zu grundlegenden **Fachbegriffen der Systemtheorie** mit den folgenden drei Aussagen zu Systemen nach HUBKA zusammenfassen [268]:

- Das Verhalten eines Systems ist durch die definierte Struktur des Systems festgelegt.
- Das (relativ) geschlossene System mit einer fixierten Struktur hat nur ein einziges Verhalten. Dabei determiniert die Struktur das Verhalten.
- Das Verhalten legt die Struktur eines Systems nicht eindeutig fest. Dasselbe Verhalten (Funktion) kann daher durch unterschiedliche Strukturen verwirklicht werden.

Entwicklungsstufen in der Systemtheorie

Weitere Wurzeln der modernen Systemtheorie sind in der Kybernetik [458], die seit 1948 v. a. von WIENER geprägt wird und Bereiche der Regelungs-, Steuerungs-, und Informationstheorie bei Maschinen und bei Lebewesen umfasst [607], in der Verwissenschaftlichung praktischen Problemlösens, die z. B. EHRENSPIEL & MEERKAMM anhand der Differenzierung zwischen intuitiven und rationalen Problemlösen sowie mittels Analyse- und Syntheseprobleme beschreiben [151], sowie in der strukturelle Denkweise der modernen Mathematik, die sich als Werkzeug der Systemtheorie anbietet [458] und die basierend auf der Mengenalgebra mit dem Konzept des Relationengebildes, also der definierten Menge von Elementen und der Menge von Relationen [524], den eingangs beschriebenen Unterschied zwischen Vielheit und Ganzheit mathematisch präzisiert [458].

Nach PULM hat die **Systemtheorie zwei Entwicklungsstufen** bzw. zwei Paradigmenwechsel vollzogen [444] (vgl. Bild 98). Die erste Entwicklungsstufe lässt sich vereinfachend mit dem Übergang von einem deutlich definierten, in sich existenten und geschlossenen System (Allgemeine Systemtheorie), das kontrolliert werden kann, hin zu einem selbststeuernden, offenen System, in das mit geeigneten Methoden steuernd eingegriffen werden kann, (Kybernetik 1. Ordnung) beschreiben. Nach der zweiten Entwicklungsstufe bzw. dem weiteren Paradigmenwechsel wird von einem selbstreferenziellen, autopoietischen und integrierenden System gesprochen, das sich selbst aus der Umwelt formieren und in dieser erhalten kann (Kybernetik 2. Ordnung). Gemäß dem Systemverständnis nach der Kybernetik 2. Ordnung ist Integration des Beobachters in das System erforderlich und lediglich Intervention bei schwer oder nicht vorhersagbaren Folgen ist möglich [444]. Infolge ihres Kerngedankens unterliegt die moderne Systemtheorie grundsätzlich der fortwährenden Entwicklung und wurde und wird daher aus unterschiedlichen Perspektiven diskutiert (vgl. u. a. [607], [425], [444], [458], [498]) und mit anderen Theorien aus diversen Disziplinen verknüpft.

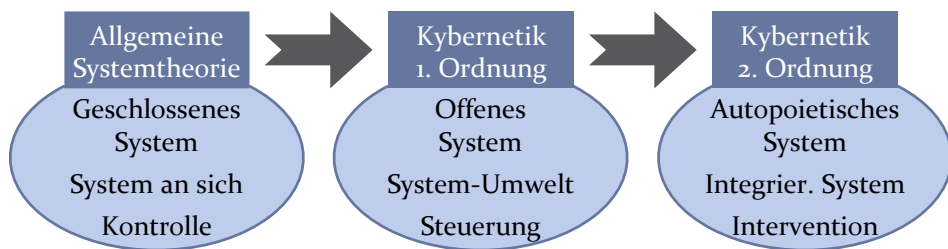


Bild 98: Entwicklungsstufen in der Systemtheorie (in Anlehnung an [444])

Tabellarische Klassifikationssystematik von Systemen

Nach diesem Abschnitt zu den zentralen Begriffsdefinitionen im Kontext der Systemtheorie wird in diesem Abschnitt eine Klassifikationssystematik von Systemen in tabellarischer Form für ein besseres Systemverständnis skizziert. Hierfür werden mit Hilfe unterschiedlicher Unterscheidungsmerkmale, Ausprägungsbereiche und ergänzenden Erläuterungen Systeme klassifiziert. Die tabellarische Klassifikationssystematik von Systemen in Bild 99 ist weitgehend aus [425] und [268] entnommen.

Unterscheidungsmerkmal		Ausprägungsbereich (gemäß Polaritätsansicht)		Erläuterungen
Seins-Ebene (Art der Elemente in Bezug auf Konkretheit)		Konkret (materiell)	Abstrakt (ideell)	Meist Mischformen (entspricht nicht: real - irreal)
Entstehung		Natürlich (von der Natur geschaffen)	Künstlich (vom Mensch geschaffen)	Künstliche, sozio-technische Systeme entwickeln natürliches Verhalten bei Realisierung
Beziehung zur Umwelt (Übersystem)		Offen (dynamisch)	Geschlossen (metastabil)	Konkrete Systeme sind nie ganz geschlossen, vom Betrachtungszeitraum abhängig (quasi geschlossen = metastabil)
Arten von Elementen		Objekt-systeme	Prozess-systeme	Objekt(systeme) i. S. v. Dinge, z. B. Rohr, Maschine Prozess(systeme) i. S. v. Operationen, z. B. Entwickeln, Fertigen
Dominanz von Elementarten (bei konkreten Systemen)		Unbelebt Technisch	Belebt Human-sozial	Wesentliche Ausprägungen sind anthropotechnische Systeme sozio-technische Systeme
Bestimmtheit von Output und/oder Zustand		Deterministisch	Stochastisch (Probabilistisch)	Deterministisch: eindeutig und in Sätzen prognostizierbar Stochastisch: nur mit Wahrscheinlichkeitswerten prognostizierbar
Abhängigkeit des Outputs		Kombinatorisch	Sequentiell	Kombinatorisch, d. h. Output abhängig nur vom Input; Sequentiell, d. h. Output abhängig vom Input und anderen Größen
Komplexität (Strukturierung)	Elementanzahl	Gering	Groß	Beliebig viele Zwischenwerte
	Varietät Elementarten	Eindisziplinär	Multidisziplinär	
	Relationen Anzahl	Gering	Groß	Beliebig viele Zwischenwerte
	Konnektivität Relationen Arten	Einheitlich	Unterschiedlich	

Unterscheidungsmerkmal		Ausprägungsbereich (gemäß Polaritätsansicht)		Erläuterungen
Funktion (Wirkung)	Bei abstrakten Systemen	Aussagend	Klassifizierend	Aussagend sind Gleichungssysteme, philosophische Systeme u. ä.
	Bei konkreten Systemen	Materiell Energetisch	Informationell	Materie/Energie-Kontinuum Meist alle drei Übertragungsarten vorliegend
Charakteristika der Systemfunktion		Linear Ohne Verzögerung	Nicht linear Mit Verzögerung	Approximation: abschnittsweises Linearisieren
Charakteristikum der Systemzustände		Diskret	Stetig	Fließender Übergang bei praktischer Analyse
Zustandsänderung		Statisch	Dynamisch	Statisches heißt Zustand in der Zeit nicht veränderlich Dynamisches heißt Zustand in der Zeit veränderlich
Variabilität (Zeitabhängigkeit)	passiv Der Funktion aktiv	Invariabel	Variabel Funktionsadaptiv	Variabel i. S. v. wählbar; funktionsadaptiv i. S. v. funktionsdynamisch; jeweils abhängig von Betrachtung, d. h. Zeitraum und Genauigkeit
	passiv Der Struktur aktiv	Inflexibel	Elastisch Selbstorganisierend	Elastisch i. S. v. adaptierbar, modular aufgebaute Systeme; Selbstorganisierend i. S. v. strukturdynamisch, lernfähige Systeme
Komplexität der Struktur		Einfach	Äußerst komplex	Einfach, z. B. Rohr, Schraube Komplex, z. B. Pkw, Flugzeug Sehr komplex, z. B. vollautom. Fabrik Äußerst komplex, z. B. digital. Gehirn
Verhalten gegenüber Störungen		Instabil	(Meta-)stabil Fließgleichgewicht	Abhängig vom Beobachtungszeitraum Ultrastabilität Multistabilität
Zielorientiertheit (Finalität)		Programmiertheit	Zielstrebigkeit	Programmiertheit bei unbelebten Systemen (Normenvorgabe) Zielstrebigkeit bei belebten Systemen (Wertungsprozess)

Bild 99: Klassifikationssystematik von Systemen (nach [425], [268])

Wesentliche Prinzipien der Systemtheorie

Die in Kapitel 2.1.1 erläuterten Grundlagen zur Systemtheorie und zum Begriff System werden in diesem Abschnitt um die von PULM in [444] zusammengestellten **Prinzipien der Systemtheorie** ergänzt, welche auch dieser Arbeit zugrunde liegen (vgl. auch [75]). Dazu werden analog nach PULM ausgewählte Prinzipien der modernen Systemtheorie skizziert, die auch zur wissenschaftlichen, methodischen und praktischen Reflektion der PE sowie zum besseren Verständnis (technischer) Systeme dienen. Dabei werden die Prinzipien der Systemtheorie erstens in sechs Systementstehungsstufen gegliedert (vgl. Bild 100), zweitens übergeordneten Prinzipien (vgl. Tabelle 7) zugeordnet und drittens den Prinzipien der klassischen Systemtechnik gegenübergestellt (vgl. Tabelle 8) [444]. Dabei betont PULM, wie auch der Autor dieser Arbeit, dass das über allem stehende Prinzip die Komplexität selbst ist [444].

Entstehung	
Prinzip	Systemtheorie
Grenzbildung	
Differenz (Bezug, Abgrenzung, Relativität) Ein System bildet eine Einheit, indem es sich von seiner Umwelt abgrenzt und intern qualitativ stärkere Strukturen als zur Umwelt und zu anderen Systemen besitzt.	Interdisziplinäre Wissenschaft
Zentralisierung (interne Kontingenzschließung, Fokus) Die Elemente des Systems richten sich gleich aus, wodurch sich einzelne Systeme selbst einschränken, aber durch Verfolgung gleicher Ziele mehr Möglichkeiten erreichen (emergente Eigenschaften), vgl. Permanentmagnet.	Ausrichtung auf nicht dogmatische Ansätze
Ressourcengewinnung	
Spezialisierung (Funktionale Differenzierung, Zweck) Um im Gesamtsystem zu bestehen, müssen sich Systeme auf eine bestimmte Funktion spezialisieren, die den Zweck des Systems darstellt und für das System nützlich ist.	Prinzipien Austausch zwischen Wissenschaften
Kontingenzschließung (Handlungsoption) Da für das System viele Handlungsoptionen bestehen, muss das System seine Möglichkeiten einschränken und sich für eine Alternative entscheiden. Darauf wird die Handlung ausgerichtet, so dass Unsicherheiten beseitigt werden.	Allgemeine Beschreibung von Entwicklungsprinzipien

Entstehung	
Prinzip	Systemtheorie
Strukturierung	
Doppelte Kontingenzschließung (Strukturelles Driften) Es gibt praktisch keine a priori Umwelt, sondern die Systeme passen sich gegenseitig an und schränken ihre Möglichkeiten ein. Die Frage ist wie sich eine abgeschlossene Einheit von Systemen ohne externen Bezug gegenseitig zu einem funktionsfähigen Ganzen konstituieren kann.	Disziplinäre Systemtheorien, Fokussierung der Disziplinen
Strukturierung (Dekomposition, Integration) Die Interaktion zwischen den Systemen ordnet sich, um die Funktionserfüllung dauerhaft sicherzustellen. Gleichzeitig werden Funktionen intern weiter ausdifferenziert. Diese Strukturen verfestigen sich im Laufe der Zeit zunehmend und führen so zu einer Stabilität des Systems.	Klare Definition systemtheoretischer Fragestellungen
Prozesssteuerung	
Prozesssteuerung (Elementaraktionen, Prozessmuster) Bei der Strukturierung wird der Austausch von z. B. Energie, Informationen, Stoffen zwischen den Teilsystemen fixiert. Um flexibel auf äußere Änderungen reagieren zu können, liegt der Fokus auf der Prozesssteuerung, wobei die Prozesse nicht vollständig beschrieben sind, sondern sich aus Prozessmustern und Elementaraktionen zusammensetzen.	Festlegung des Austauschs disziplinärer, aber systemtheoretisch beschriebener Prinzipien
Kommunikation (Medien, Austausch) Die Prozesssteuerung führt zur Bildung von Medien und Kommunikationen, die letztlich das System konstituieren. Die Medien erlauben eine Vereinfachung der Handlung oder Auswahl weiterer Handlungsoptionen. Die einzelnen Medien erfüllen unterschiedliche Funktionen.	Prinzipien (Erkenntnisse) aller Bereiche als Medien
Zeitbezug (Flussorientierung, Dialektik) Kurzlebige Kommunikationen bedingen, dass das System nur in der Zeit existent ist und sich ständig selbst erhalten muss. Gleichzeitig führt der Zeitbezug dazu, dass jedes System auf einem vorherigen System oder Systemzustand basiert und sich ständig weiterentwickelt. Statische, endgültig entwickelte Systeme oder Lösungen existieren nicht.	Betrachtung der Systemtheorie als sich entwickelndes, ungenaues Erklärungsmodell

Entstehung	
Prinzip	System- theorie
Reflexion	
Reflexivität (Selbstreferenz) Um wirklich flexibel auf die Umwelt reagieren zu können, sind Selbstreferenz und Reflexivität notwendig, d. h. eine Betrachtung des Systems durch sich selbst. Es existiert eine weitere Dialektik zwischen Kontingenzbeschränkung, um handlungsfähig zu sein, und Kontingenzerhalt, um auf sich ändernde Umweltbedingungen reagieren zu können.	s. diese gesamte Spalte und vgl. Selbst-ähnlichkeit
Subjektivität (Wahrnehmung, Konstruktivismus) Die Wahrnehmung des eigenen Selbst als auch der Umwelt ist subjektiv, d. h. hauptsächlich vom betrachtenden System abhängig. Dies bedingt zum einen, dass Systeme sich selbst nicht objektiv betrachten können, zum anderen, dass ein objektiver Eingriff in ein komplexes System nicht möglich ist. Zudem ist Subjektivität Großteiles auch kulturbedingt und bezieht sich somit nicht nur auf das Individuum.	Systemtheorie ist subjektives Erklärungsmodell
Genese	
Evolution (Vermehrung, Vererbung, Selektion) Zum Erhalt der Art des Systems und zur Sinnkonstitution ist die Vermehrung des Systems notwendig. Vermehrung und Selektion bedingen sich gegenseitig, d. h. ohne Selektion führt die Vermehrung ins Unendliche, ohne Vermehrung löst die Selektion das System auf. Dieser Prozess kann als Kommunikation (der Erbinformation) gesehen werden.	Entwicklung weiterer Systemtheorien
Pluralismus (Parallelität, Widersprüche, Inkonsistenz) Die kontinuierliche Entstehung neuer Systeme führt dazu, dass auch widersprüchliche, inkonsistente Systeme nebeneinander bestehen können. Die Systemtheorie als Differenztheorie versucht gerade, die unmöglich eindeutige Hierarchisierung der Welt durch andere Ansätze zu begreifen, d. h. Systeme werden nicht mehr (nur) hierarchisch zerlegt, sondern über Differenzkriterien spezifiziert und können sich überschneiden.	Widersprüchliche Systemtheorien möglich (s. reduktionistisch vs. holistisch)

Bild 100: Prinzipien der Systemtheorie (nach [444])

Tabelle 7: Übergeordnete Prinzipien der Systemtheorie (nach [444])

Prinzip	Systemtheorie
Selbstähnlichkeit (Ebenen, Analogien) Im Sinne der Systemtheorie folgen Systeme auf verschiedenen Ebenen den gleichen Prinzipien und Prozessen. Dieser Punkt bezieht sich v. a. auf die Rekapitulation der Phylogenese durch die Ontogenese, d. h. der Entwicklung der Systeme selbst.	Systemtheorie folgt systemtheoretischer Entwicklung
Emergenz (Teil – Ganzes) Das Zusammenspiel vieler gleicher oder verschiedener Elemente kann zu neuen Eigenschaften auf höherer Systemebene führen. Dabei ist es von einer Ebene aus schwer bis unmöglich, die emergenten Eigenschaften der übergeordneten Ebene zu erkennen. Das Optimum für das Gesamtsystem kann sich vom Optimum für die Teilsysteme unterscheiden.	Systemtheorie ergibt sich aus der Vielzahl unabhängiger Prinzipien
Unvollständig (unendlicher Regress, Rekursivität) Die ständige Dynamik und reflexive Neubildung von Systemen führen zur Tatsache, dass jedes System an sich unvollständig ist. Besonders durch die Reflexivität kann ein System sich nie ganz selbst enthalten. Im Umkehrschluss ermöglicht die Unvollständigkeit den immer fortlaufenden Evolutionsprozess.	Vollständige Systemtheorie als vollständige Weiterklärung nicht möglich
Unbestimmtheit (Indeterminismus, Unvorhersehbarkeit) Die Entwicklung von Systemen ist trotz ihrer Determiniertheit unbestimmt, da es eben der Entwicklung bedarf, das Ergebnis „herauszufinden“. In Bezug auf Handlungen bedeutet dies, dass auch deren Ergebnis unbestimmt ist.	Entwicklung der Systemtheorie offen

Abschließend sei ergänzend auf die aufschlussreichen Erkenntnisse aus der **Verknüpfung der Systemtheorie mit dem Konstruktivismus** hingewiesen (vgl. [277]). Dabei beleuchtet der Konstruktivismus insb. das Erschaffen bzw. Modellieren von Systemen durch das Wahrnehmen bzw. Erkennen des Modellierers aus unterschiedlichen Blickwinkeln (z. B. methodischer Erlanger Konstruktivismus vs. wahrnehmungsbiologischer Radikaler Konstruktivismus) beleuchtet. Dies ist auch für diese Arbeit wichtig zu berücksichtigen, da schon allein in Folge einer systematischen Modellierung von komplexen Systeme Erkenntnisse gewonnen und aufgrund dessen ein besseres Systemverständnis realisiert wird.

Tabelle 8: Gegenüberstellung der Prinzipien der Systemtechnik und der Systemtheorie (nach [444])

Systemtechnik	Systemtheorie
Hierarchiebildung	Strukturierung
Systemgrenze / Systemgrenzenverschiebung	Differenz / Abgrenzung
Abstraktion / Konkretisierung	Kontingenzschließung
Trennung Analyse – Synthese	Wahrnehmung und Handlung
Variation	Genese
Bewertung (Bewertungsmethoden)	Selektion
Vorgehen vom Groben zum Detail	---
Einflussanalyse	---
---	Autopoiesis

Die Allgemeine **Systemtheorie ist disziplinenübergreifend** und demzufolge eine von den verschiedenen Wissenschaftsbereichen unabhängige Theorie [Ropholaz2009]. Da die Anwendung der Systemtheorie stets im Kontext einer bestimmten Disziplin erfolgt [75], erfährt diese im Rahmen ihrer Anwendung je nach Disziplin spezifische Ausprägungen [444], wie z. B. in der Organisationstheorie (vgl. [65]) und der Theorie technischer Systeme (vgl. Kapitel 2.1.2 und [421]). Mit dem Systembegriff sind eine Vielzahl weiterer Begriffe (z. B. Struktur, Verhalten, Zustand) eng verknüpft, die kurz im Anhang A.5 erklärt werden. Zudem sind für ein besseres Systemverständnis in diesem Anhang A.5 auch Prinzipien der Systemtheorie sowie eine Klassifikationssystematik von Systemen zu finden.

A.6 Bedeutung, Bestandteile und Klassifikation technischer Systeme

Zur Vertiefung der **wissenschaftlichen Grundlagen bzgl. technischer Systeme** dienen die folgenden Abschnitte. Im ersten Abschnitt erfolgt die Beschreibung der Bedeutung von Systemtechnik, danach der Erläuterung der Bestandteile von technischen Systemen und abschließend noch die Darstellung der Klassifikation von technischen Systemen.

Bedeutung der Systemtechnik

HUBKA erklärt ein **technisches System** als ein aus einer gewissen Menge von Elementen bestehendes und mit Relationen zwischen diesen Elementen nach definierten Regeln geordnetes Ganzes [268]. Dabei erklärt er, dass nicht alle Elemente eines Systems Relationen haben müssen. Analog zum hierarchischen Konzept (vgl. Bild 7) erläutert er die Relativität der Begriffe System und Element, nachdem ein System auch als Element eines übergeordneten Systems sein kann und ein Element wiederum als ein System aus untergeordneten Elementen angesehen werden kann. Auch die zwei weiteren Konzepte eines Systems (vgl. Bild 7) lassen sich nach ihm analog auf technische Systeme anwenden [268]. Er interpretiert die geplante, zielgerichtete Transformation bzw. Umwandlung von Input in Output durch erzwungene Naturphänomene als „technische Funktion“ des technischen Systems. Dabei betont er, dass es sich bei den Ein- und Ausgangsgrößen technischer Systeme um Stoffe (Masse), Energie und Informationen handelt [268] (vgl. [267] und [458]).

Die Gründe für das Entstehen sowie auch für die steigende **Bedeutung der Systemtechnik** sind vielfältig. Im Folgenden werden ausgewählte Aspekte in Anlehnung an [425], [458], [237], [239], [64] [151] zusammenfassend skizziert:

- Die zunehmende **Wissens- und Informationsflut** sowie die stetig steigenden Spezialgebiete in den einzelnen Wissenschaften führen einerseits dazu, dass sich Spezialisten aufgrund ihrer jeweiligen fachspezifischen Sprachen kaum mehr austauschen können, und andererseits dazu, dass Generalisten das ganze System in seiner Ganzheit kaum begreifen können (vgl. Kapitel 1.4). Die Systemtechnik kann mit ihrem ganzheitlichen Systemverständnis und Denken als fachübergreifendes Verständigungsmittel dienen und die interdisziplinäre Nutzung von Wissen fördern.
- Die unüberschaubare **Vielfalt von Produkten**, die in den letzten Jahren bereits entwickelt und produziert wurden, sowie der Anstieg an verschiedenen Varianten und Versionen bestehender Produkte

sowie zusätzlicher Neuentwicklungen führt dazu, dass diese nur schwer verstanden und klassifiziert werden können. Die Systemtechnik kann mit ihrer übergreifenden Theorie und Systematik als ein Hilfsmittel zur Einordnung und zum besseren Verständnis technischer Systeme und Produkte dienen.

Die immer **komplexer werdenden Systeme**, in denen Produkte integriert sind und mit denen Produkte interagieren müssen, führen dazu, dass die Planung, Analyse, Bewertung und Optimierung solcher Systeme schwieriger werden (vgl. Kapitel 1.5). Die Systemtechnik kann mit ihrem Verständnis und ihren Konzepten von Systemen als ein Unterstützungsmittel für eine produktneutrale wie auch produktspezifisch flexibel adaptierte Arbeitsmethode dienen.

Bestandteile von technischen Systemen

Wie bereits in Kapitel 2.1.2 geschildert, handelt es sich bei **technischen Systemen** um geometrisch stoffliche Gebilde, die einen bestimmten Zweck erfüllen [151], [173] und eine Untergruppe der künstlichen Systeme darstellen [268]. Dabei wird die Erfüllung des Zwecks (d. h. die Funktion oder Aufgabe, vgl. Anhang A.5 für Details) des technischen Systems durch die technische Umsetzung z. B. physikalischer Effekte erreicht [173]. Dafür sind (System-)Elemente erforderlich, die mittels der technischen Umsetzung eines physikalischen Effekts die bezweckte Funktion realisieren [173]. Demzufolge werden diese Elemente in Modellen als Funktionsträger und in realen Produkten als Bauteile bezeichnet [315]. Die Umsetzung physikalischer Effekte wird an den sog. Wirkflächen der involvierten Funktionsträger erzwungen und dazu müssen die entsprechend benötigten Wirkflächen von Bauteilen miteinander durch Material in Kontakt stehen (z. B. um Kräfte, Momente zu leiten) [8]. Schlussendlich wird die Gestaltung technischer Systeme vor allem von der gewünschten Funktion beeinflusst.

Danach umfassen technische Systeme die folgenden Bestandteile (diese und ergänzende Systembegriffe sind im Anhang A.5 näher erläutert):

- Systemgrenze als Abgrenzung zwischen einem System und der Umwelt (in Bild 8 in Kapitel 2.1.1 als Strich-Punkt-Linie dargestellt)
- Umwelt bzw. -feld als Umgebung des Systems außerhalb dessen Grenze (d. h. alles was nicht zum betrachteten System gehört), wie z. B. (übergeordnete) Systeme („Supersysteme“)
- Element als unterschiedliche „Bausteine“ eines Systems, die auch Sub- bzw. Teilsysteme sein können (z. B. „Subsystem E“)

- Relationen (Beziehungen) als i. d. R. verschiedene Verbindungen zwischen den einzelnen Elementen und Systemen, wie z. B. Flussrelationen, hierarchische Relationen (bei technischen Systemen v. a. auch Funktions-, Lage-, Bewegungs- und Kräftebeziehungen [151] und [55])
- In- und Output (Ein- und Ausgangsgrößen) als Relationen des Systems zu seiner Umwelt, welche die Systemgrenze überschreiten

Aufgrund der obigen Definition eines technischen Systems lassen sich **Schlussfolgerungen** ziehen:

- Das technische Systemdenken (engl. Systems Thinking) ermöglicht es komplexe, technische Systeme besser zu verstehen [237]
- Das technische Systemdenken führt die drei in Kapitel 2.1.1 genannten Systemkonzepte bzw. Systembetrachtungen zusammen (vgl. auch [75])
- Das technische Systemdenken unterstützt bei der Planung sowie Entwicklung technischer Systeme (z. B. technischer Produkte) [151]
- Das technische Systemdenken ist eine hilfreiche Basis für den Modellerstellungsprozess (z. B. für Produktmodelle) in der PE und erleichtert dessen Detaillierung (vgl. auch [619])
- Das technische Systemdenken unterstützt bei der abstrakten Modellierung und bei der Fokussierung auf das Wesentliche [151]
- Das technische Systemdenken ermöglicht das frühzeitige Erkennen, Durchdenken und Simulieren von Eigenschaften und Verhaltensweisen technischer Systeme mit Hilfe graphisch und/oder mathematisch angefertigter Modelle [151].

Klassifikation von technischen Systemen

Die **Klassifikation technischer Systeme** kann nach vielen verschiedenen Gesichtspunkten erfolgen. So können nach HUBKA technische Systeme u. a. nach der Funktion (z. B. Verformen, Festhalten, Drehen), nach der Hauptumsetzungsart (d. h. Transformation von Stoffen, Energie und Informationen), nach dem Prinzip zur Ausführung von Arbeitswirkung (z. B. mechanisches, elektronisches, hydraulisches Prinzip), nach den Bedingungen des Verhaltens (z. B. Leistung, Geschwindigkeit, Beschleunigung) und nach dem Grad der Komplexität (z. B. Bauteil, Baugruppe, Anlage) klassifiziert werden [268]. KOLLER unterscheidet bei den Klassifikationsmerkmalen nach Branche (z. B. Maschinenbau, Elektrotechnik und Bauwesen), nach Zweck (i. S. v. Umsetzungsart), wobei er jeweils die energieumsetzenden, stoffumsetzenden und informationsumsetzenden Systeme weiter gliedert, nach Fachgebieten (d. h. biologische, chemische und physikalische

Systeme), wobei er die physikalischen Systeme weiter gliedert (z. B. mechanische, elektronische, hydraulische Systeme) und nach dem Grad der Komplexität (z. B. von Punkte über Bauteile bis hin zu Analysensystemen) [315]. EHRENSPIEL & MEERKAMMEN betonen auch die Vielzahl möglicher Klassifizierungsmerkmale und unterscheiden technische Systeme v. a. nach der Hauptumsetzungsart (d. h. Stoff, Energie, Information), nach Grad der Komplexität analog zu Koller (z. B. von Punkte über Bauteile bis hin zu Analysensystemen), nach den Modellierungsbegriffen technischer Systeme (d. h. v. a. die Bereiche Funktion durch funktionelle Lösungsmöglichkeiten, Physik durch physikalische Lösungsmöglichkeiten und Gestalt durch gestalterische Lösungsmöglichkeiten).

Der **Zweck von Klassifikationen technischer Systeme** ist es zum einen, einen Überblick zu geben sowie die Systeme zu charakterisieren und zu bewerten [268]. Zum anderen soll dadurch ihre Vielfalt dargestellt und ein besseres Verständnis der verschiedenen technischen Systeme erlangt werden [151]. Die für diese Arbeit relevanten Klassifikationen werden daher im Folgenden kurz beschrieben, und zwar erstens nach der Hauptumsatzart, zweitens nach den Modellierungsbegriffen bzw. -bereichen und drittens nach der Komplexität technischer Systeme.

Technische Systeme können den Zweck haben, Stoffe, Energien und/oder Daten bzw. Informationen umzusetzen, um eine Zustandsänderung eines Systems (vgl. Operand im Anhang A.5) zu bewirken [315], [151]. Somit ist die **Klassifikation nach der Hauptumsatzart** die erste der nachfolgend erläuterten Klassifikationsarten. Demzufolge lassen sich stoffumsetzende, energieumsetzende und informationsumsetzende Systeme voneinander unterscheiden [315] (s. Tabelle 9). Oft wirken jedoch alle drei Umsatzarten bei vielen technischen Systemen zusammen, um die gewünschte Zustandsänderung bzw. Hauptfunktion zu erfüllen [151]. Dabei wird nach FELDHOUSEN in der Konstruktionslehre je nach Hauptumsatz bzw. -fluss eines Systems zwischen Apparat (Hauptumsatz Stoff), Maschine (Hauptumsatz Energie) oder Gerät (Hauptumsatz Information) differenziert, wobei er auf die Abweichungen zur Definition sowie der vielfältigen Verwendung der drei Begriffe im Alltag (vgl. Bild 101) aufmerksam macht [173]. Tiefergehende Erläuterungen zu und Zusammenhänge zwischen stoff-, energie- und signalumsetzenden Produkten sind in [176] zu finden. Darin werden die drei Hauptumsatzarten wie folgt definiert [176]:

- Stoff (Materie): Im Allgemeinen, Gas, Flüssigkeit, Feststoff, z. B. auch Rohprodukt, Material, Bauteil, Endprodukt, Behandlungsobjekt, Prüfgegenstand, geprüfter oder behandelter Gegenstand

- Energie: Im Allgemeinen mechanische, thermische, elektrische, chemische, optische Energie, z. B. auch Kraft, Strom, Wärme
- Information: Im Allgemeinen Daten, Signale, z. B. auch Anzeige, Messgröße, Steuerimpuls.

In der Tabelle 9 werden die drei **Hauptumsatzarten in weitere Systeme** gegliedert, wobei auch KOLLER betont, dass neben diesen Systemen nach einer Umsatzart natürlich auch „hybride“ Systeme bestehen. Diese „hybriden“ Systeme verbinden (trennen) Stoff mit (von) Energie, Stoff mit (von) Information und Energie mit (von) Information sowie alle drei Umsatzarten miteinander (voneinander) [315].

Tabelle 9: Gliederung der drei Umsatzarten in elementare Systeme (nach [315])

Stoffumsetzende Systeme d. h. Stoff oder Stoffeigenschaften ... Systeme	Energieumsetzende Systeme d. h. Energie oder Energiekomponenten ... Systeme)	Informationsumsetzende Systeme d. h. Informationen bzw. Daten ... Systeme
wandelnde	wandelnde	verknüpfende
vergrößernde oder verkleinernde	vergrößernde oder verkleinernde	vervielfältigende
leitende oder isolierende	leitende oder isolierende	leitende oder isolierende
fügende oder lösende	richtungsändernde	umcodierende
sammelnde oder teilende	sammelnde oder teilende	speichernde
mischende oder trennende	mischende oder trennende	

Für eine Veranschaulichung werden in Bild 101 die drei **Hauptumsatzarten mit typischen Maschinenelemente** und Produkttypen (d. h. Apparate, Maschinen, Geräte) aufgezählt, die wiederum nach dem Umsatzprodukt klassifiziert sind [151]. In dieser Übersicht lassen sich viele Maschinenelemente der mechanischen Antriebstechnik zuordnen, die bei sich bewegendenden Produkten verwendet werden [151]. Dabei machen EHRLENSPIEL & MEERKAMMN darauf aufmerksam, dass auch Software ein „Maschinenelement“ ist, dass für die meisten antriebstechnischen Produkte von zunehmender Bedeutung ist [151].

Hauptumsatzart	Umsatzprodukt	Maschinenelement	Produkttyp
Stoffumsatz	Fester Stoff	Sieb, Filter Mischschaufel Drehmeißel	Sieb-, Filtermaschine Mischer Drehmaschine
	Flüssiger Stoff	Zahnrad, Kolben Umströmte Schaufel Rohrleitung	Zahnrad-/Kolbenpumpe Kreispumpe Trinkwasserversorgung
	Gasförmiger Stoff	Umströmte Schaufel Behälter Rohrleitung	Ventilator Gasflasche Pipeline für Erdgas
Energieumsatz	Mechan. Energie	Zahnrad, Welle, Lager, Kupplung	Getriebe
	Pneumat. Energie	Kolben Zahnrad	Verbrennungsmotor Zahnradpumpe/-motor
	Hydraul. E.	Umströmte Schaufel	Turbine, Verdichter
	Therm. E.	Rohrleitung	Wärmetauscher
	Elektr. E.	Elektromagnet	Elektromotor
Informationsumsatz	Energetische Information	Zahnrad	Indukt. Drehzahlmessgerät
		Kolben	Mehrwegeventil
		Umströmte Schaufel	Geschwindigkeitsmessgerät
		Rohrleitung	Pneumatische Steuerung
		Elektrische Schalter	Telegraf
		Software	Rechner
	Stoffliche Information	Zahnrad Kolben Letter	Ovalrad-Messgerät Mengenregler Schreibmaschine

Bild 101: Klassifikation technischer Systeme nach der Hauptumsatzart (in Anlehnung an [151])

Die zweite Klassifikationsart technischer Systeme, die in den nachfolgenden Abschnitten skizziert wird, erfolgt nach **Modellierungsbegriffen bzw. Abstraktionsebenen der Modellierung**. Dabei lassen sich je nach Literaturquelle verschieden viele und unterschiedliche Abstraktionsstufen technischer Systeme finden. So unterscheidet HUBKA zwischen folgenden

Abstraktionsstufen, die an Gliederungshierarchieebenen der Biologie angelehnt, sind: Reich, Stamm, Klasse, Ordnung, Familie, Gattung, Art. Durch die jeweilige Festlegung bestimmter Konstruktionsmerkmale werden die Auswahlmöglichkeiten von einer zur nächsten Abstraktionsstufe top-down immer mehr eingegrenzt und definiert (z. B. ausgehend vom Reich Maschinensystem, über den Stamm Werkzeugmaschine bis zur Drehmaschine) [268]. Ebenfalls top-down bzw. entlang der absteigenden, linken Seite des V-Modells (vgl. VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ [575] werden nach KLEINER & KRAMER gemäß dem RFLP-Ansatz, der v. a. die Entwicklung mechatronischer Produkte ganzheitlich unterstützt, die vier Abstraktionsebenen der Modellierung unterschieden: Requirement bzw. Anforderungsmodell, Functional bzw. Funktionsmodell, Logical bzw. Logisches Modell und Physical bzw. Physikalisches Modell (d. h. 3D-CAD-Modell) [303], [304].

Ein dritter Ansatz zur Klassifikation technischer Systeme nach Abstraktionsebenen, der auch in EHRENSPIEL & MEERKAMM dargestellt wird [151] und v. a. in Publikationen der Konstruktionsmethodik (vgl. u. a. [422], [461], [462], [460], [64]) sowie darauf aufbauende VDI-Richtlinien (vgl. u. a. [580], [581], [570]) zu finden ist, unterscheidet ausgehend von den Anforderungen die drei horizontal angeordneten Ebenen Funktion, Physik, Gestalt. Diese drei Ebenen werden in Bild 102 in Anlehnung an BLASS skizziert [64] und zum einen um die Produktionsebene (i. S. v. Fertigungs- und Montageebene) und zum anderen um ein vereinfachtes Beispiel ergänzt. Anschließend werden die vier **Abstraktionsebene Funktion, Physik, Gestalt, Produktion** in Form einer Pyramide in Anlehnung an EHRENSPIEL & MEERKAMM dargestellt (vgl. Bild 103). Darin sind ausgehend von z. B. einer Anforderung als erstes die Funktionsebene mit allen funktionellen Lösungsmöglichkeiten, als zweites die Physikebene mit allen prinzipiellen physikalischen (biologischen, chemischen) Lösungsmöglichkeiten, als drittes die Gestaltebene mit allen gestalterischen und stofflichen Lösungsmöglichkeiten und als viertes Produktionsebene mit allen fertigungs- und montagetechnischen Lösungsmöglichkeiten vom Abstrakten zum Konkreten zu durchzulaufen. Durch die Pyramidenform der hierarchischen Modellierung des technischen Systems wird die Zunahme von sinnvollen Lösungsmöglichkeiten sowie der Informationszuwachs mit zunehmender Konkretisierung zum Ausdruck gebracht [151].

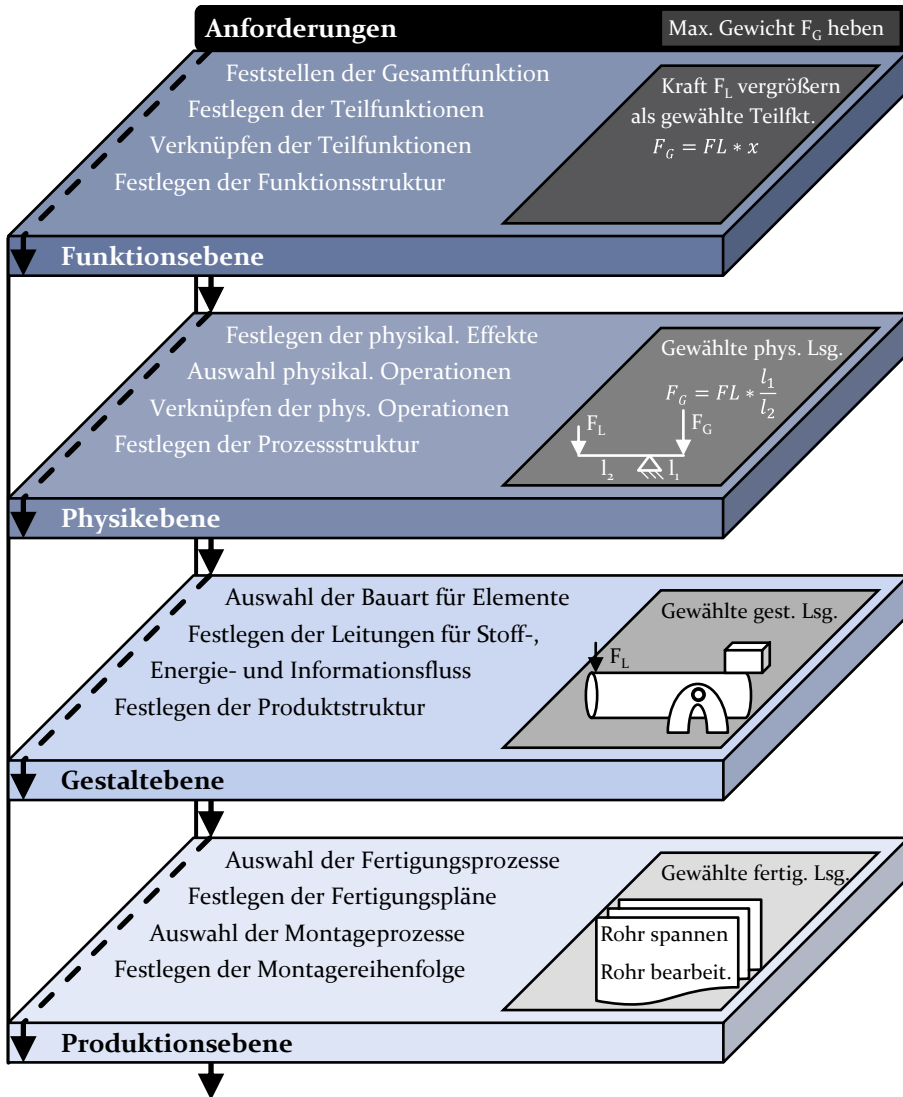


Bild 102: Erläuterung und Zusammenhang der Funktions-, Physik-, Gestalt- und Produktionsebene (in Anlehnung an [64], ergänzt um Beispiel)

Im **vereinfachten Beispiel**, das in Bild 102 rechts skizziert ist, kann die Anforderung, ein max. Gewicht F_G (z. B. in Form eines Quaders) heben zu können, durch die damit verknüpfte Gesamtfunktion „Gewicht heben“ realisiert werden. Diese Gesamtfunktion kann durch unterschiedlich festzulegende Teilfunktionen (z. B. Kraft F_L vergrößern) sowie durch verschiedene Verknüpfungen dieser Teilfunktionen zur Funktionsstruktur realisiert werden. Dabei sind die Prozesse der Erarbeitung einer zweckmäßigen Lösungsmenge und -auswahl schematisch als Dreiecksflächen gezeichnet.

Die Dreiecke in Bild 103 repräsentieren die erarbeitete Lösungsmenge und deren Schnittlinie mit der Funktionsebene die sinnvoll miteinander verknüpften funktionellen Lösungsmöglichkeiten, wovon eine Lösung zur weiteren Konkretisierung zu wählen ist (z. B. Kraft F_L vergrößern). Für gewählte Teilfunktion können auf der folgenden Ebene wieder viele verschiedene prinzipielle physikalische Lösungsmöglichkeiten in Betracht kommen – analog auch beim Übergang zur Gestalt- und Produktionsebene. Die zunehmende Anzahl an alternativen Lösungsmöglichkeiten auf den jeweils darunter liegenden Ebenen geht mit dem **Konkretisieren mittels Bewertung und Auswahl einer Lösungsmöglichkeit** einher, was in Bild 103 nur anhand einer einzigen Teilfunktion, prinzipiellen und gestalterischen Lösung skizziert ist. In der realen PE steigt die Komplexität dadurch, dass zum einen alle erforderlichen Teilfunktionen eines technischen Produkts zu beachten sind und zum anderen beim Übergang zwischen zwei Ebenen i. d. R. verschiedene Lösungsmöglichkeiten parallel so lange zu konkretisieren sind, bis eine Bewertung und Auswahl möglich ist [151].

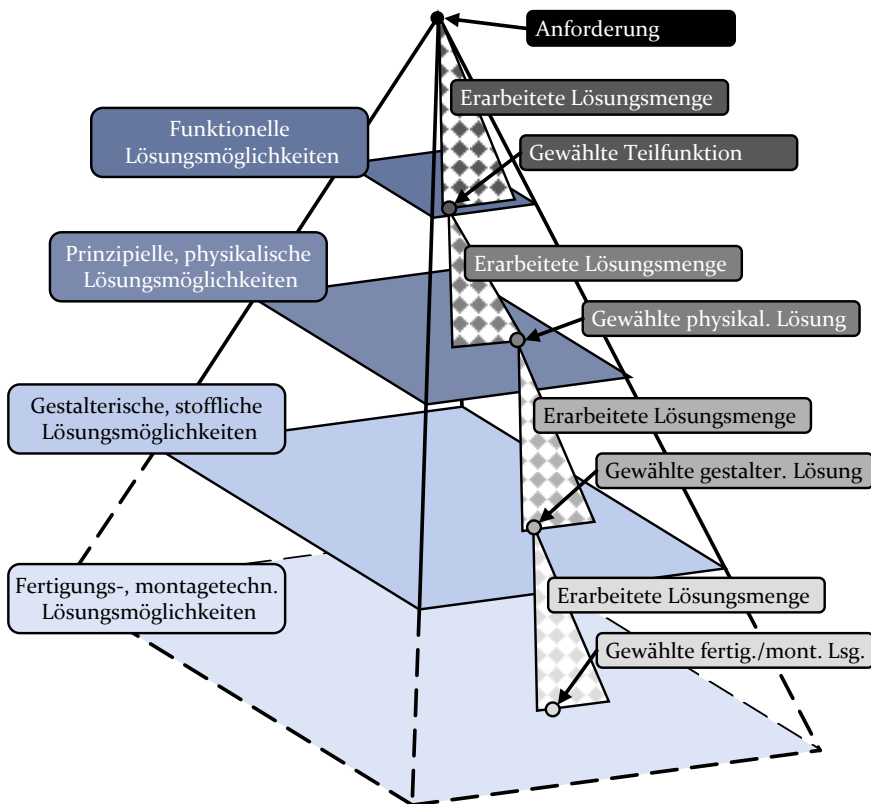


Bild 103: Klassifikation technischer Systeme in Funktions-, Physik-, Gestalt- und Produktionsebene (in Anlehnung an [151])

Die vierte und abschließend erläuterte Art der Klassifikation technischer Systeme erfolgt nach dem **Komplexitätsgrad der Elemente bzw. der Komplexitätsstufe**. Der Begriff Komplexität wird im folgenden Kapitel 2.2 ausführlich beleuchtet. Vereinfachend beschreibt KOLLER, dass ein technisches Produkt dann komplexer als ein anderes Produkt ist, wenn es sich aus weniger komplexen Produktelementen zusammensetzen lässt [315]. Daher erfolgt die Klassifizierung technischer Systeme nach KOLLER mit Hilfe von Komplexitätsstufen, und zwar ausgehend von einem einzigen Punkt über Bauteile und -gruppen bis zu komplexen, technischen Systemen (vgl. Bild 104). Dabei wird ein nicht weiter zerlegbares Produktelement eines technischen Systems oft auch als Teil, Bauteil, Einzelteil, Bauelement oder Komponente bezeichnet [315] (vgl. auch [151]). Im Gegensatz zu den jeweils zehn Komplexitätsstufen in [315] und [151], sind in Bild 104 elf Komplexitätsstufen mit ihren jeweiligen Produktelementen bzw. Teilsystemen zusammen mit Erläuterungen und Beispielen dargestellt. Dies liegt daran, dass KOLLER sowie EHRENSPIEL & MEERKAMM jeweils eine andere Komplexitätsstufe in zwei unterteilen und dies im Bild 104 zusammenfassend berücksichtigt wird.

Die **Gründe für die Klassifikation technischer Systeme nach deren Komplexitätsgrad** sind u. a. nach HUBKA vielfältig [268]:

- Komplexitätsstufe steht in Korrelation mit dem Komplexitätsgrad der Lösung von Entwicklungsaufgaben
- Komplexitätsstufe weist auf erforderliche Kompetenzen sowie den Spezialisierungsgrad des Entwicklers hin, wie z. B. beschäftigt sich ein Systemingenieur bzw. Komplexitätsingenieur (vgl. Kapitel 1.4) eher mit technischen Systemen und Anlagen, ein Projekt- bzw. Konstruktionsingenieur eher mit einzelnen Maschinen und Baugruppen und ein Fachexperte bzw. Detailkonstrukteur eher mit einem Maschinenelement und/oder einem Bauteil
- Komplexitätsstufe formiert eine Orientierung bzw. Struktur für die Arbeit des Entwicklers, d. h., wenn er auf einer bestimmten Komplexitätsstufe arbeitet, muss er lediglich die Eingliederung seiner Entwicklungsaufgabe bzw. seines Produktelements in die höhere Stufe kennen.

Komplexitätsstufe	Produkt-element, Teilsystem	Erläuterungen	Beispiele
1	Ecke, Spitze	Punkte (Punktfolgen, Splines), Schnittpunkte von Teilkanten, zu einer Spitze zulaufe Teiloberflächen	Teilecke, Nadelspitze
2	Kante, Grenze zw. Flächen	Flächenberandung, kanten-/ tangentialförmiger Übergang zweier Teiloberflächen	Schnittkanten, Bauteilkanten 1. & 2. Ordnung.
3	Teiloberfläche, Wirkfläche	Teile der Oberfläche eines Bauteils	Lagerauflagefläche, Zylinderlauflagefläche, Schneidfläche
4	Wirkflächenpaar	Zusammenwirkende Teiloberflächen zweier Teile	Lagerwellen- und -schalenflächen, Wälzflächenpaar
5	Teilkörper	Teilkörper aus denen ein Bauteil zusammengesetzt werden kann	Zylinder, Kegelstumpf, Senkung, Fase, Eindrehung
6	Bauteil, Werkstück	Nicht weiter demontierbares Teil eines technischen Produkts	Innen-/Außenring, Kugel, Schraube, Bolzen, Feder (1 Teil)
7	Maschinenelement	Eigenständiges funktionsfähiges Subsystem aus mind. zwei Bauteilen	Wälzlager, Chip Spanneinrichtung, (~ 10 Teile)
8	Baugruppe	Technisches System mit einigen Bauteilen und/oder Maschinenelementen	Getriebemotorführung, Vorschubeinheit (~ 100 Teile)
9	Maschine, Gerät, Apparat	Technisches System zur Realisierung mind. eines bestimmten Stoff-, Energie-, oder Informationsprozesses	Werkzeugmaschine, Wasserturbine, Computer, Kopiermaschine (~ 1.000 Teile)
10	Anlage, Aggregate	Technisches System, das aus mehreren Maschinen, Geräten und/oder Apparaten besteht	Walzwerkanlage, Notstromaggregat, Super-Computer, Automobil (~ 10.000 Teile)
11	Technisches System	Komplexes, technisches System, das aus vielen, einzelnen techn. Systemen besteht	Flugsystem mit Flughafen, Flugzeuge, -sicherung; (> 100.000 Teile)

Bild 104: Klassifikation technischer Systeme nach Komplexitätsgrad bzw. -stufen (nach [315] und ergänzt in Anlehnung an [151])

Aus der **zusammenfassenden Betrachtung Klassifikation technischer Systeme** lässt sich festhalten, dass sich die Komplexität technischer Systeme u. a. erstens aus einer Vielzahl an wirkenden, oft wechselseitig miteinander in Beziehung stehenden Stoff-, Energie- und Informationsumsätzen, zweitens den vielen, auszuwählenden und zueinander in Relation stehenden Lösungsmöglichkeiten – ausgehend von den Anforderungen – auf Funktions-, Physik-, Gestalt- und Produktionsebene sowie drittens der mit der Komplexitätsstufe zunehmenden Anzahl und Vielfalt an Teilen mit deren jeweiligen Relationen ableiten lässt. Dies wird am Vergleich eines früheren mit einem heutigen Pkw deutlich (vgl. [151]: Ein damals rein mechanisch wirkender Pkw (abgesehen vom Motor) ist deutlicher weniger komplex als ein heutiger, hybrider bzw. elektrifizierter Pkw, der zudem auch eine Vielzahl an elektrischen, elektronischen, hydrostatischen, hydrodynamischen sowie weiteren Software- und Serviceelemente (z. B. Assistenzsysteme, Entertainmentsysteme) aufweist, die darüber hinaus häufig miteinander verknüpft sind.

A.7 Grundlagen zum Komplexitätsverständnis

Präzisierung des Komplexitätsverständnisses

Dieser Abschnitt basiert v. a. auf den zwei zusammenhängenden Zeitschriftenartikeln [451] und [452] zum Komplexitätsmanagement von REIß, so dass nachfolgend auf ein Referenzieren der beiden Artikel verzichtet wird. Ziel des Absatzes ist eine Präzisierung des Verständnisses für Komplexität. Die Differenzierung zwischen einer **Komplexität von Elementen sowie von Relationen** geht auf die systemtheoretische Denkweise zurück (vgl. Kapitel 2.1.1). Dabei besteht die größere Herausforderung für das Komplexitätsmanagement im Hinblick auf die Relationen, die auch u. a. als Beziehungen, Verbindungen, Vernetzung, Interdependenzen, Wechselwirkungen, Links bezeichnet werden.

Die beiden linken Dimensionen bzw. Spalten in Bild 10 in Kapitel 2.2.1, die **Vielzahl und Vielfalt** stellen die „traditionellen“ Dimensionen der Komplexität dar (auch als vereinfachte Komplexität in Kapitel 2.2 bezeichnet). Die **Vielzahl und Vielfalt** an Elementen und Relationen äußern sich einerseits in komplizierten Mengengerüsten, wie z. B. Produkten, Kunden, Mitarbeiter, Standorte und Umlaufvermögen, und andererseits in komplizierten Zeitgerüsten, wie z. B. Durchlaufzeiten, Transportzeiten, zeitlicher Kapazitätsinanspruchnahme. Im Umgang mit dieser Komplexität, die zum einen auch als „konjunktiven“ Komplexität (AND-Verknüpfung) oder Abstimmungskomplexität bezeichnet wird, sind v. a. Abstimmungsprobleme zu lösen.

Die beiden rechten Dimensionen bzw. Spalten in Bild 10 in Kapitel 2.2.1, die **Veränderlichkeit und Vieldeutigkeit** stellen die „modernen“ Dimensionen der Komplexität dar, die sich durch die kybernetischen Phänomene der aktiven Variation und reaktiven Adaptivität an wechselnde Konstellationen (vgl. u. a. Evolution, Flexibilität, Chaos, Risiko, Regelkreise) charakterisieren lassen. Die Handhabung dieser Komplexität, die auch als „disjunktive“ Komplexität (OR-Verknüpfung) oder Anpassungskomplexität benannt wird, erfordert fortwährendes Anpassungsmanagements.

Die in Bild 10 in Kapitel 2.2.1 verwendeten Begriffe sind im Folgenden nur kurz erläutert worden. Daher werden in der folgenden Tabelle 10 die Begriffe **Vielzahl, Vielfalt, Veränderlichkeit und Vieldeutigkeit** jeweils mit ihrem Bezug zu den Systemelementen und -relationen nach REIß diskutiert [451]. Diese Diskussion der Begriffe dient v. a. für deren Konkretisierung sowie für ein **grundlegendes Verständnis der verschiedenen Facetten von Komplexität**.

Tabelle 10: Diskussion der vier Dimensionen der Komplexität (inhaltlich in Anlehnung an [451])

	Diskussion des Begriffs
Vielzahl	Vielzahl beschreibt die Facette von Komplexität, die sich durch eine schwer überschaubare Anzahl von Elementen (z. B. Anzahl von Produktmerkmale, Anforderungen, Zielen) sowie von Relationen (z. B. geometrischen Relationen, Einflussbeziehungen) ausdrückt.
Anzahl, Größe der Elemente	Anzahl, GröÙte bezeichnet dabei Menge, Volumen, Häufigkeit, Breite, Tiefe, Länge, Dauer und weitere MaßgröÙen für die Dimensionierung eines Systems.
Kopplungsgrad der Relationen	Der Kopplungsgrad gibt rein quantitativ die Dichte eines materiellen, energetischen und informatorischen Geflechts an Relationen wieder.
Vielfalt	Vielfalt drückt die Verschiedenartigkeit in einem System aus. Dabei reicht das Spektrum von wenig komplexen homogenen Systemen bis hin zu außerordentlich pluralistischen, heterogenen Systemen.
Diversität der Elemente	Diversität der Elemente lässt sich mittels von Dispersionskennzahlen operationalisieren. Hierbei reicht das Spektrum von der Spannweite über die Varianz bis hin zu Diversifikationskennzahlen. Auffallende Abweichungen vom Durchschnitt (i. S. v. Ausreißer, Exoten, abweichendes Verhalten, außerordentliche hohe Aufwendungen und Erlöse) wirken hier komplexitätssteigernd.
Divergenz der Relationen	Divergenz liegt vor, wenn im System gegenläufige Strömungen bestehen. So werden u. a. Material-, Energie- und Stoffflüsse in mehreren, entgegengesetzten Richtungen (wechselseitige Beziehungen, reziproke Interdependenz, Netzwerk- statt Dienstwegkommunikation) ebenso wie zweigleisige Planung nach dem Gegenstromprinzip (bottom-up und top-down) oder auch wechselseitige Abstimmung zwischen Funktionsbereichen (z. B. fertigungsgerechte Konstruktion) als komplex i. S. v. schwer koordinierbar eingestuft. Auch das Wirken gegensätzlicher Kräfte im Gefolge eines Interessenkonflikts (i. S. v. Macht und Gegenmacht, Promotoren und Opponenten, Dissens, Dialektik, Disharmonie, Dissonanzen, Inkompatibilitäten, Antinomie, Konkurrenz, Rivalität) ist komplexitätsstiftend. Ferner kommt es aufgrund von Gegensätzlichkeiten zu sog. Misfits (d. h. mangelnde Stimmigkeit), die sich oft in Ungleichgewichten niederschlagen. Hierzu zählen bspw. Verletzungen des organisatorischen Kongruenzprinzips.
Veränderlichkeit	Veränderlichkeit gibt das komplexe Zeitverhalten von komplexen Systemen wieder. Zudem wird mit der Veränderlichkeit ausgedrückt, wie sich das u. a. in der Vieldeutigkeit implizierte Veränderungspotential konkret im Zeitverlauf aktualisiert.

	Diskussion des Begriffs
Dynamik der Elemente	Dynamik äußert sich in kontinuierlichen Wachstums- bzw. Schrumpfungsmustern. Zu den fundamentalen Wachstumsmustern zählen das proportionale Wachstum mit konstanten Wachstumsraten sowie die progressive und degressive Wachstumsdynamik. Mit gängigen Zeitmustern operiert die XYZ-Analyse bzgl. der Materialbedarfsermittlung. Signifikant komplexer sind die diskontinuierlich-unstetigen Muster, bei denen u. a. Sprünge, Brüche und Umkehrungen auftreten.
Chaos der Relationen	Chaos soll hier Unschärfe beim Zusammenspiel mehrerer Größen im Zeitablauf bezeichnen. So ist es quasi unmöglich, bei der chaotischen Lagerung (d. h. keine zeitlich konstanten, „festen“ Lagerplätze), der flexiblen Fertigung ohne ordnende Losgrößenbildung, bei einem wechselnden Personaleinsatz oder bei der selbstorganisatorischen Zusammenarbeit von Mitgliedern in einem teilautonomen Entwicklungsteam echte Regelmäßigkeiten festzustellen. Diese Unberechenbarkeit bzw. Labilität zeigt sich allerdings nur bei einer kurzfristigen Detailbetrachtung durch ein kleines Zeitfenster. Grundsätzlich funktionieren solche Systeme nach Regeln (z. B. FCFS-Abfolge für die Maschinenbelegung) bzw. Spielregeln (z. B. Regeln des fairen Wettbewerbs). Sie legen aber jeweils nur den Rahmen fest und lassen sich als Invarianzen im Systemverhalten auch nur über einen längeren Betrachtungszeitraum rekonstruieren. Im Falle der wechselnden Zuordnungen von Personal zu Stellen (z. B. handelt es sich oft nicht um ein Vagabundieren, sondern um ein mehr oder weniger systematisches Rotieren).
Vieldeutigkeit	Vieldeutigkeit bezeichnet die Schwierigkeit sich ein „klares“ Bild vom betreffenden System zu machen. Verantwortlich für derartige Unbestimmtheiten, Unsicherheiten, Risiken, Ambiguitäten, Ambivalenzen sind meist Defizite im Wissen über das System (z. B. Ignoranz, Intransparenz einer dualen Organisationsstruktur), in Ausnahmen auch die stochastische Beschaffenheit des Systems selbst. Systemelemente besitzen keine prägnante Gestalt. Ziele sind nur vage und damit mehrdeutig formuliert, eine eindeutige Kursbestimmung lässt sich nicht rekonstruieren, bei Abhängigkeiten fällt die Mustererkennung schwer.
Freiheitsgrade der Elemente	Anhand der vorhandenen Freiheitsgrade (z. B. Optionen in einem Software-Menü) lässt sich angeben, in welchen alternativen Erscheinungsformen ein Element auftreten kann. Insb. in der PE haben Elemente sehr viele Freiheitsgrade. Ferner operiert die Entscheidungstheorie mit derartigen Alternativen- und Zustandsräumen, je nach Wissensstand mit oder ohne Wahrscheinlichkeiten.
Unschärfe der Relationen	Vieldeutige Relationen zeichnen sich durch Unschärfe aus. Unschärfe Zuordnungsrelationen zwischen Kostenträgern einerseits und den in Anspruch genommenen Ressourcen andererseits entstehen durch Ressourcen-Pooling und provozieren das heikle Gemeinkostenproblem. Unklare Attribuierungen, Anstrengungs-Results- bzw.

Diskussion des Begriffs

Resultats-Gratifikations-Erwartungen beeinträchtigen die Motivation und die Herausbildung von unternehmerischem Kosten- und Erlösbewusstsein. Ein Beispiel für unscharfe Wirkungsrelationen stellt die Synergieformel ($2 + 2 = 5$) dar. Im Falle von Lag- bzw. Lead-Strukturen (z. B. bei Markov-Prozessen) kommt die Vieldeutigkeit dadurch zustande, dass im Prinzip zwar klare Abhängigkeiten bestehen, diese sich aber nicht in ein und derselben Periode, sondern nur zeitlich verzögert nachvollziehen lassen. Unschärfen resultieren zudem aus einer Überlagerung mehrerer Muster, bspw. konjunkturellen, strukturellen, saisonalen, exzeptionellen und zufälligen Veränderungen im Auftragseingang. Im Unterschied zu einseitig-mechanistischen Ursache-Wirkungsbeziehungen sind wechselseitig-zirkuläre Abhängigkeiten naturgemäß auch schlechter berechenbar, wie z. B. bei diffizilen Abhängigkeitsverhältnissen zwischen PE und Produktionsanlagenentwicklung bei zeitlich parallelierter Zusammenarbeit (Simultaneous Engineering).

Dieser tabellarische Überblick über die vier Dimensionen der Komplexität zeigt, dass das Komplexitätsverständnis selbst sehr vielschichtig und verflochten ist. Dabei kristallisieren sich Begriffe wie Vielzahl, Vielfalt, Veränderlichkeit und Vieldeutigkeit von Elementen (Elementkomplexität) und von Relationen (Relationenkomplexität) bzw. deren Interaktion heraus (vgl. [66]). Abschließend sind in Tabelle 11 unterschiedliche **Begriffsbestimmungen von Komplexität** von einigen Wissenschaftlern nach EMBST zusammengestellt. Daraus wird deutlich, dass – bis auf die Vieldeutigkeit – die meisten Dimensionen der Komplexität von diesen Autoren verwendet werden. Dabei sei angemerkt, dass die Dimension „Interaktion“ bei Reiß integraler Bestandteil seiner vier Dimensionen ist.

Tabelle 11: Begriffsbestimmungen von Komplexität verschiedener (gemäß [159])

Dimension	Feess [171]	Börner [70]	Dörner [133]	Luhmann [361]	Reiss [451]	Schuh [491]	Ulrich/Probst [565]
Vielzahl	x	x	x	x	x	x	x
Vielfalt	x	x	x	x	x	x	x
Veränderlichkeit	x			x	x	x	x
Vieldeutigkeit					x	x	

Dimension	Feess [171]	Börner [70]	Dörner [133]	Luhmann [361]	Reiss [451]	Schuh [491]	Ulrich/Probst [565]
Interaktion	x	x	x	x	x	x	x

Kennzeichen und Umgang mit komplexen Problemsituationen

Nach ULRICH & PROBST liegen mehr Problemstellungen (äußerst) komplexe Systeme zugrunde als dies den meisten Menschen bewusst ist [565]. Dies liegt nach den beiden Wissenschaftlern v. a. daran, dass Menschen das Vorliegen von komplexen Systemen nicht wahrnehmen (wollen), da dies oft unangenehm ist und unsicher macht [565]. Der **Umgang und das Lösen komplexer Situation** erfordert im Vergleich zu einfachen Situationen nicht nur andere Denkweisen, sondern auch geeignete Modellierungs- und Problemlösungsmethoden, wie in Tabelle 12 dargestellt wird.

Tabelle 12: Gegenüberstellung einfacher und komplexer Situationen (nach [565])

	Einfache Situation	Komplexe Situation
Charakteristik	Wenige, gleichartige Elemente Geringe Vernetztheit Wenig Verhaltensmöglichkeiten der Elemente Determinierte, stabile Wirkungsverläufe	Viele, verschiedene Elemente Starke Vernetztheit Viele, verschiedene Verhaltensmöglichkeiten der Elemente Viele, veränderliche Wirkungsverläufe
Erfassbarkeit	Vollständig analysierbar Quantifizierbares Verhalten Prognostizierbar, d. h. analytisch erklärbar und d. h. Sicherheit erreichbar	Beschränkt analysierbar Beschränkt quantifizierbare Verhaltensmuster Erkennbar, d. h. synthetisch verstehbar und d. h. Unsicherheit reduzierbar
Geeignete Denkweise	Kausalanalytisches Denken	Ganzheitliches Denken
Geeigneter Modellierungsansatz	Vorbild: „Maschine“ Systemtyp: Triviales System	Vorbild: „Ökosystem“ Systemtyp: Nicht trivia. System

	Einfache Situation	Komplexe Situation
Geeignete Problemlösungsmethoden	„Exakte“, quantitative Methoden Algorithmen	„Nicht exakte“, qualitative Methoden Heuristiken
Faktische Beeinflussbarkeit	Konstruierbar Beherrschbar mit „Restrisiko“	Beschränkt gestaltbar und lenkbar „Kultivierbar“

Typische Konflikte im Umgang mit Komplexität

Typische **Konfliktpotentiale** in Folge des Interessengegensatzes zwischen der Markt- und Umsatzorientierung einerseits und der Kostenorientierung andererseits sind in Tabelle 13 gegenübergestellt. Dieser wird am besten dadurch entschieden, dass nur diejenigen neuen Produktvarianten in das Produktportfolio aufgenommen werden, die die prognostizierte Stückkostenerhöhung durch einen entsprechend höheren Preis überkompensieren.

Tabelle 13: Typische Konflikte aufgrund des Interessengegensatzes zwischen der Markt- und Kostenorientierung (weitgehend entnommen aus [506])

Konflikt	Markt-/Umsatzorientierung	Kostenorientierung
Insb. Koordinationskonflikt	Typischer Kommentar der Vertriebs-/Marketingseite	Typischer Kommentar der Entwicklungs-/Produktionsseite
Kapazitätsplanung und langfristige Umsatzprognose	„Warum reicht unsere Kapazität nicht aus?“	„Warum wurden uns keine genauen Umsatzvorhersagen vorgegeben?“
Produktionsplan und kurzfristige Umsatzprognose	„Wir müssen schneller reagieren können. Unsere Zeitvorgaben sind lächerlich.“	„Wir brauchen realistische Lieferzusagen und Umsatzprognosen, die sich nicht so schnell ändern wie das Wetter.“
Auslieferung und Vertrieb	„Warum haben wir nie die Ware auf Lager, die wir brauchen?“	„Wir können nicht alles im Lager haben.“
Qualitätskontrolle	„Warum können wir keine vernünftige Qualität zu vernünftigen Kosten produzieren?“	„Warum müssen wir immer Sonderanfertigungen anbieten, die schwer herstellbar sind und dem Kunden kaum Nutzen bieten?“
Angebotsbreite	„Unsere Kunden verlangen Vielseitigkeit.“	„Die Baureihe hat zu viele Varianten und ist damit unrentabel.“
Kostenkontrolle	„Unsere Kosten sind derart hoch, dass wir draußen am Markt nicht konkurrieren können.“	„Sofortige Lieferung, breitestes Typenprogramm innerhalb einer Baureihe, schnellste Reaktion auf Änderungswünsche und hohe Qualität zu niedrigen Kosten – was da von uns verlangt wird, ist ein Ding der Unmöglichkeit.“
Einführung neuer Produkte	„Neue und v. a. innovative Produkte sind unser Lebenselixier!“	„Überflüssige Programmerweiterungen verursachen unnötige Kosten.“

Definition des Begriffs Komplexität aus verschiedenen Sichten

Aus **etymologischer Sicht** geht komplex, die Grundform von Komplexität, auf das lateinische Substantiv *complexus* zurück, welches mit Umfassung, Umschließung, Verknüpfung übersetzen lässt [137]. Mit komplex ist somit umfassend, umschlungen oder verwoben i. S. v. vielschichtig, verknüpft und miteinander zusammenhängend oder ineinandergreifend gemeint [480]. Dies kann nach HEYLIGHEN auch als grundlegende Dualität zwischen Teilen, die gleichzeitig verschieden bzw. eindeutig unterscheidbar und gleichzeitig auch miteinander zusammenhängen bzw. verbunden sind, interpretiert werden [257]. Demnach würde die Komplexität eines Systems zum einen mit der Anzahl an unterschiedlichen Elementen und zum anderen mit der Anzahl an Verbindungen zwischen diesen ansteigen [257].

Aus **systemtheoretischer Sicht** (vgl. Kapitel 2.1.1) wird Komplexität in einem System bzw. eines Systems mit strukturellen Merkmalen (vgl. [425]) bzw. bestimmten Eigenschaften (vgl. [91]) erläutert. So besteht zum Beispiel nach SIMON ein komplexes System aus vielen Elementen, die miteinander auf nicht einfacher Art und Weise in Beziehung stehen [509], wobei bereits in [302] zwischen der Vielzahl und Vielfalt der zwischen den Elementen bestehenden Relationen unterschieden wird. In der Definition des Komplexitätsbegriffs nach KLABUNDE werden die beiden zuvor skizzierten Merkmale Varietät, d. h. Anzahl und Art der Elemente in einem System, und Konnektivität, d. h. Anzahl und Art der Relationen zwischen den Systemelementen, um das dritte Merkmal Dynamik, d. h. die Unbestimmbarkeit und Unvorhersehbarkeit komplexer Systeme (vgl. für Details v. a. [1]), ergänzt [360] und [301]. Dabei ergänzt REIß wiederum die beiden traditionellen Dimensionen der Vielzahl und Vielfalt (vgl. Varietät und Konnektivität, die dazu eigentlich orthogonal stehen) um die zwei Dimensionen Vieldeutigkeit und Veränderlichkeit (vgl. Dynamik) von Elementen und Relationen [451]. Dabei versteht er unter Vieldeutigkeit die Schwierigkeiten sich ein klares Bild vom System zu machen, wobei er dies vorwiegend auf mangelndes Wissen über das System zurückführt (z. B. unsichere, unbestimmte ambivalente Elemente und/oder unscharfe zugeordnete, wirkende Relationen) [451]. Mit Veränderlichkeit drückt er vor allem die Anzahl und Art der Veränderungen über einen Zeitablauf aus [451]. Dieses Verständnis komplexer Systeme erweiterten und konkretisieren u. a. DÖRNER und NEGELE z. B. im Hinblick auf jeweils verschiedene Aspekte der Dynamik, (In-)Transparenz und Zustandsbetrachtungen von Systemen [133], [401].

Auch aus **Produktentwicklungssicht** gibt es eine Vielzahl verschiedener Sichtweisen des Begriffs Komplexität. Wesentliche Forschungsarbeiten, die sich grundlegend und ausführlich mit der Erläuterung des Begriffs Komplexität aus unterschiedlichen Perspektiven beschäftigt haben, sind u. a. [151], [354] [355], [401], [327], [378], [491], [322], [120], [84], [2].

A.8 Arten, Treiber und Grade von Komplexität

In diesem Anhang werden **unterschiedlichen Komplexitätsarten aus verschiedenen Sichten dazugehörige Komplexitätstreiber und Grade von Komplexität** beleuchtet. So werden in einem ersten Abschnitt die externen von den internen Komplexitätsarten differenziert. Danach werden für die sechs Komplexitätsarten, die in der vorliegenden Arbeit im Fokus stehen, die zentralen Problemstellungen und Herausforderungen beschrieben. In den darauffolgenden sechs Abschnitten werden die sechs Komplexitätsarten jeweils detaillierter vorgestellt. Dabei sind die Komplexitätsarten in der folgenden Reihenfolge angeführt: Anforderungskomplexität, Produktkomplexität, Prozesskomplexität, Wissenskomplexität (inkl. Informations- und Datenkomplexität), Organisationskomplexität und abschließend die Komplexitätskosten. Diese Erläuterungen dienen für ein besseres Verständnis der Komplexitätsarten an sich und für deren vielen, wechselseitigen Relationen. Zudem werden dadurch die große Bedeutung sowie die Herausforderungen im Umgang mit komplexen Systemen und damit des Komplexitätsmanagements in der PE verdeutlicht. Anschließend werden die wesentlichen Treiber skizziert und die im Kapitel 2.2.2 vorgestellten Grade der Komplexität konkretisiert und um Beispiele ergänzt.

Externe und interne Komplexitätsarten

Aufgrund des intensiven Wettbewerbs und der durch VUCA geprägten Rahmenbedingungen nimmt die Komplexität in nahezu allen Bereichen der Entwicklung komplexer Systeme, wie z. B. technischer Produkte, zu (vgl. Kapitel 1.1). Dabei sind die Bedeutung (vgl. Kapitel 1), der Grad (vgl. Kapitel 2.2.2) und auch die Art der Komplexität stark abhängig u. a. von dem Betrachtungsfokus, der Perspektive und der (hieratischen) Ebene der Betrachtung des komplexen Systems (vgl. [360]). Daher ist es naheliegend, dass auch in der Literatur **viele verschiedene Arten von Komplexität** zu finden sind. So unterscheidet z. B. FRIEDRICH im Rahmen der Entwicklung mechatronischer Systeme in statische und dynamische Komplexität, wobei er im Hinblick auf die statische Komplexität die zwei Aspekte der technischen Produktkomplexität sowie der interdisziplinären Zusammenarbeit betont, und mit Bezug zur dynamischen Komplexität die beiden Herausforderungen in Folge von Veränderung (d. h. des Produkts, seiner Elemente und Strukturen sowie der verfügbaren Technologien) und durch Wiederverwendung (v. a. von Varianten, Modulen) hervorhebt [195]. Eine weitere Unterscheidung ist z. B. in [69] und [135] zu finden, nach der Komplexität in der Entwicklung in einerseits technische Komplexität, die sich durch die Zahl der Elemente und den Charakter der Beziehungen charakterisieren

lässt, und andererseits organisatorische Komplexität, die durch die Zahl der Beziehungen und die zeitliche Veränderung gekennzeichnet ist, differenzieren lässt. Nach Auffassung von GLEICH kann aus Perspektive des Produktlebenszyklus zwischen den folgenden Arten der Komplexität differenziert werden: Marktkomplexität, Produktkomplexität, Prozesskomplexität, Informationstechnik (IT)-Struktur-Komplexität, Organisationskomplexität, Infrastrukturkomplexität und Sozialkomplexität [496]. Davon etwas abweichend unterscheiden DANILOVIC & BROWNING (aufbauend auf [90]) aus PE-Sicht zwischen fünf verschiedenen Teilbereichen der Komplexität, die als Produkt, Prozess, Organisation, Arbeitsmittel und Ziel/Rahmenbedingungen bezeichnet werden [119].

Die grundsätzliche **Differenzierung zwischen externer und interner Komplexität** (s. Bild 105) ist in den Ingenieurwissenschaften (vgl. u. a. [355], [151], [491], [322], [401]) sowie auch in den Managementwissenschaften (vgl. u. a. [610], [480], [159], [457]) am weitesten verbreitet (vgl. auch [91]). Dabei werden diese zwei unterschiedlichen Komplexitätsarten je nach wissenschaftlicher Arbeit auch als exogene und endogene Komplexität bezeichnet (s. z. B. [66], [300]). Externe bzw. exogene Komplexitätsarten, wie z. B. Markt-, Informations-, Gesetzes- und Normenkomplexität) wirken von außen auf ein Unternehmen und interne bzw. endogene Arten von Komplexität, wie z. B. Produkt-, Prozess-, Wissens- und Organisationskomplexität beschreiben die unternehmensinterne Komplexität, die grundsätzlich (leichter) zu beeinflussen ist (vgl. Bild 105 [355], [151]).

Demnach kann nach PONN aus Sicht der PE, die sowohl ein Unternehmensprozess als auch eine Organisationseinheit ist, zwischen **unternehmens-externen und -internen Einflussbereich** differenziert werden [437] (vgl. Bild 105). Neben den bereits angeführten, externen Komplexitätsarten, wie Märkte bzw. Kunden, sind im Bild 105 z. B. auch Lieferanten, Konkurrenten und Kooperationspartner aufgrund des zunehmenden globalen Wettbewerbs sowie Politik, Gesellschaft und weitere, heterogene Interessensgruppen aufgrund des steigenden ökologischen Bewusstseins als externe Einflussbereiche auf die PE dargestellt [437]. Zum internen Einflussbereich aus Sicht der PE gehören u. a. mit der Entwicklung in Kooperation stehende Menschen, Organisationseinheiten und das Unternehmen selbst sowie auch die im Rahmen der PE eingesetzten Methoden und Werkzeuge [437]. Der nicht immer trennscharfe Übergang zwischen dem externen und internen Einflussbereich wird z. B. mit Blick auf die Branchen- und Technologiekomplexität, die bspw. von Entwicklungszyklen beeinflusst werden, in Bild 105 skizziert [437].

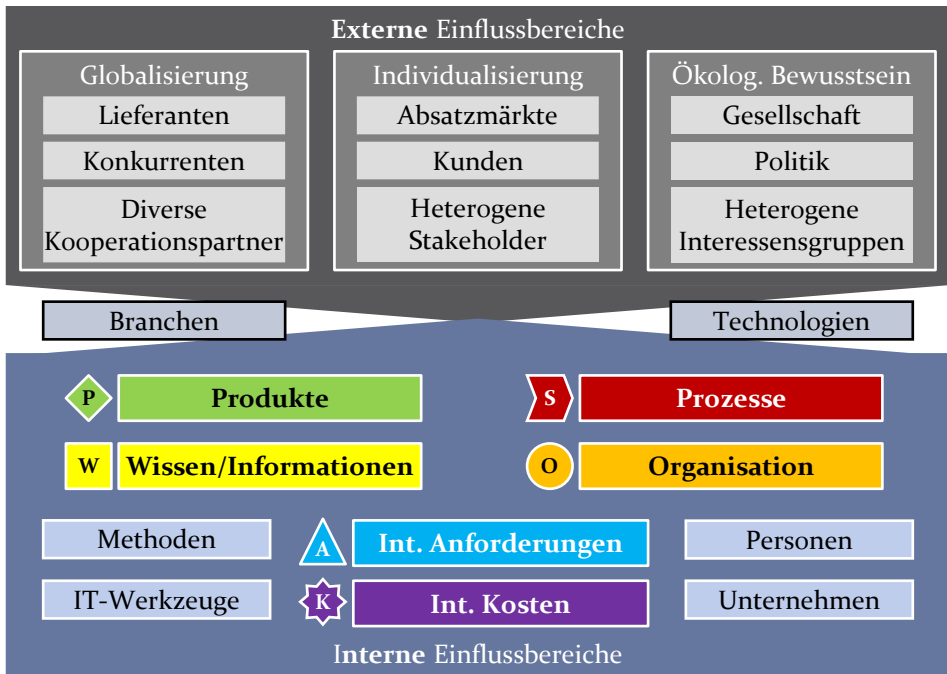


Bild 105: Externe und interne Einflussbereiche der PE (mit Inhalten aus [437])

Das Bild 105 eignet sich auch gut zur **Verortung der sechs Domänen**, die in dieser Arbeit im Fokus stehen und daher farblich hervorgehoben sind (vgl. Kapitel 1 und für Details Anhang A.8). Während die vier Arten der Komplexität Produkt, Prozess, Wissen und Organisation nahezu vollständig im internen Einflussbereich der PE liegen, sind die Anforderungs- und die Kostenkomplexität teilweise dem internen und teilweise dem externen Einflussbereich zuzuordnen.

Herausforderungen aufgrund der sechs Komplexitätsarten

Als Ursache für diese vier internen Komplexitätsdomänen kann insb. die steigende **Anforderungskomplexität** bzw. Spezifikationskomplexität angesehen werden. Diese beruht einerseits auf der zunehmenden Individualisierung und Diversifizierung aufgrund kundenspezifischer Anforderungen (vgl. u. a. [354], [175]) und andererseits auf der steigenden Anzahl an unternehmensexternen Gesetzen, Normen und Einflüssen sowie unternehmensinternen (DfX-)Richtlinien, Anweisungen und Vorgaben (vgl. u. a. [588], [36], [151]). Die Herausforderung für den Entwickler liegt hierbei darin, Produkte zu entwickeln, die diese zahlreichen Anforderungen und ihre komplexen Interdependenzen erfüllen (vgl. [374]).

Die zunehmende **Produktkomplexität** ist v. a. auch eine Folge der wachsenden Anforderungskomplexität. Diese Produktkomplexität wird durch die steigende Vielfalt an Produktfunktionen und -eigenschaften sichtbar. Diese Produktvielfalt führt zu einer zunehmenden Anzahl an Varianten auf Produkt-, Baugruppen- und Komponentenebene (vgl. u. a. [603]). Dabei wirkt sich eine deutlich ansteigende (Varianten-)Anzahl von Komponenten und Baugruppen in komplexen Produkten auch stark auf alle Bereiche der Wertschöpfungskette in produzierenden Unternehmen aus [141]. Eine große Herausforderung für Produktentwickler liegt folglich darin, die richtigen Produktmerkmale (bzw. Merkmalsausprägungen) festzulegen, die möglichst optimal zur Realisierung der erwünschten Produkteigenschaften beitragen. Eine weitere Herausforderung für die PE ist, dass aufgrund der Produktkomplexität bereits kleine Änderungen (z. B. eines Merkmals), zu einer großen Anzahl an Änderungen der zugehörigen Eigenschaften des Produkts führen können [P13]. Diese nachträglichen Änderungen und damit einhergehende beträchtliche Zusatzkosten und Verzögerungen, v. a. in späten Phasen des PEPs, lassen sich jedoch i. d. R. vermeiden [613].

Die steigende **Prozesskomplexität** ist oft ein anforderungs- und/oder produktinduziertes Phänomen, da es viele Wechselwirkungen zum einen zur Anforderungsdomäne (z. B. Änderungsmanagementprozesse) und zum anderen zur Produktdomäne (z. B. Konstruktions- und Simulationsprozesse) gibt. Daher hat eine steigende Produktkomplexität i. d. R. auch eine Zunahme der Prozesskomplexität zur Folge [326]. Dabei resultiert die Höhe der Prozesskomplexität v. a. aus der Anzahl und dem Aufbau der Schnittstellen, dem Vernetzungs- und Standardisierungsgrad der Prozesse [480] sowie aus der hohen Dynamik und dem größeren Zeitdruck [301] in Folge kürzerer Innovations- und Produktlebenszyklen (s. für weitere Definitionen [326]). Daher lässt sich die PE gemäß der Geschäftsprozessklassifizierung nach EPPLER ET AL. neben der starken Wissensintensität insb. auch durch eine sehr hohe Prozesskomplexität charakterisieren [164]. Da der PEP durch eine steigende Komplexität geprägt ist, die sich u. a. aus der Vielfalt vieler Einflüsse des Entwicklungsvorhabens [588], der Abhängigkeiten der Entwicklungsprozesse (z. B. bei Simultaneous Engineering) bzw. Teilaufgaben aufgrund Arbeitsteilung [310], vieler Unsicherheiten und auch Anforderungen resultiert [550], steht der Entwickler vor großen Koordinations- und Steuerungsherausforderungen.

Die zunehmende **Wissenskomplexität** ist oft ein Ergebnis der steigenden Anforderungs-, Produkt- und Prozesskomplexität und v. a. der (externen) Daten- und Informationskomplexität. Aufgrund der steigenden Informations- und Wissensflut, der sich verkürzenden Wissenshalbwertszeiten

[439] sowie einer (oft notwendigen) steigenden Wissensintensität [164] ergeben sich beträchtliche Herausforderungen für Produktentwickler. So steigt einerseits das (neue) Wissen exponentiell an und andererseits nimmt das (vorhandene) Wissen aufgrund der sog. Wissenshalbwertszeit (d. h. der Zeitspanne, in der 50 Prozent des erworbenen Wissens überholt ist) in ähnlicher Geschwindigkeit ab, was z. B. bedeutet, dass das in der Erstausbildung erworbene Wissen nach rund 25 Jahren fast völlig veraltet ist [85]. Daher stehen Entwickler zunehmend vor der Herausforderung die benötigten Informationen und das erforderliche Wissen über z. B. Anforderungen, Produkte, Prozesse aus einer Wissensflut (z. B. aus E-Mails, Datenbanken, Online- und Offline-Archiven) extrahieren zu müssen [289] (vgl. auch die beiden Studien bzgl. weiterer Probleme beim Umgang mit Wissen [93] [584]). Parallel dazu nimmt der Entwicklungsfortschritt stetig zu (z. B. durch Integration verschiedener Technologiebereichen [470] und Spezialgebiete [151]). Mit der daraus resultierenden Wissenskomplexität sehen sich Entwickler immer mehr konfrontiert und müssen demnach nicht nur über Kompetenz zur Wissensentwicklung sondern auch zum Wissensmanagement verfügen [470].

Die steigende **Organisationskomplexität** ist auch v. a. auf die zuvor erläuterten Komplexitätsarten zurückzuführen [326]. So haben empirische Analysen gezeigt, dass mit hoher Produkt- und Prozesskomplexität und insb. vielen involvierten Spezialisten der Wissenstransfer schwieriger wird [310]. Daher wird die Zusammenarbeit von Spezialisten mit heterogenem Fach- und Erfahrungswissen in interdisziplinären Teams für die Entwicklung der komplexer werdenden Produkte immer herausfordernder [470]. Gründe hierfür sind, dass sich die Spezialisten untereinander nur sehr schwer verstehen, da sie für ihre jeweiligen Fachgebiete eigene Modelle und Sprachen verwenden [151], und die Integration dieser Spezialfächer immer schwieriger wird [149], [150]. So nehmen die Herausforderungen für Entwickler mit steigender Wissenskomplexität zu, da das benötigte Wissen zur richtigen Zeit dem richtigen Empfänger im Entwicklungsteam zur Verfügung stehen muss. Dies führt sowohl zu einem steigenden Bedarf und damit einem erhöhten Koordinations- und Abstimmungsaufwand in Entwicklungsteams, die häufig in globalen, dezentralen und flexiblen Organisationsstrukturen mit einem hohen Selbstorganisationsgrad zusammenarbeiten [328]. Aufgrund einer häufigen (Neu-)Bildung von Teams und der sich erhöhenden Anzahl an interdisziplinären Teammitgliedern in Entwicklungsprojekten sowie der Kooperation mit Lieferanten- und Kundenteams wird die Zusammenarbeit mit der Anzahl an teaminternen und -externen Personen immer schwieriger [328]. Entwicklungsteams sind sowohl

funktional als auch fachlich heterogener und die dadurch notwendige Arbeitsteilung geht mit einer hohen Interdependenz der Teammitglieder einher, was zusätzliche Koordinationsprobleme bedeutet.

Die zunehmenden Herausforderungen beim **Kostenmanagement** (s. Komplexitätskosten nachfolgend und Auswirkungen von Komplexität in Anhang A.9) im PEP ergeben sich in Folge der bereits zuvor skizzierten Komplexitätsarten [550] als auch durch den Wandel der Märkte (vgl. Bild 1 in Kapitel 1.1). Diese haben sich von einem Verkäufer- über einen Käufer- zu einem sog. Erlebnismarkt weiterentwickelt (d. h. Unternehmen müssen individuelle Kundenbedürfnisse mit Produkten mit möglichst geringen Preisen – und damit niedrigen (Selbst-)Kosten – befriedigen) [297]. Somit stehen Entwickler grundsätzlich vor der Herausforderung kostengünstige und innovative bzw. leistungsfähige Produkte zu entwickeln [152]. Dabei ist eine Schwierigkeit, dass – v. a. in Wachstumsphasen – mit dem Ziel der individuellen Bedürfnisbefriedigung der Kunden die Produktkomplexität stark steigt (z. B. durch hohe Variantenanzahl) und dies i. d. R. zu einer überproportionalen Kostenzunahme führt [221]. Besonders in gesättigten Märkten müssen diese innovativen und oft auch komplexeren Produkte zunehmend effektiver entwickelt, kostengünstiger angeboten (z. B. durch effizienten Ressourceneinsatz) sowie in einer kürzen Entwicklungszeit realisiert werden [175]. Nach EHRENSPIEL ET AL. resultieren hohe Produktkosten immer aus falschen Entscheidungen während des PEPs, die häufig auf Basis ungenügender Lösungsanalysen von Entwicklern getroffen werden [152]. Gründe für ungenügende Analysen sind v. a. eine hohe Wissenskomplexität (z. B. schlechte Wissensverfügbarkeit) sowie eine hohe Organisationskomplexität (z. B. ungeeignete Teamorganisation) [152].

Anforderungskomplexität

An die Entwicklung komplexer, technischer Systeme werden nach HUBKA Anforderungen gestellt, welche mit den realisierten Produkteigenschaften verglichen werden [268]. Dabei entsprechen die Anforderungen am Anfang des PEP den Zielvorstellungen für das zu entwickelnde Produkt [77] und den technischen Entwicklungszielen bzw. geforderten Produkteigenschaften [437]. Nach BREIING & KNOSALA können zu Anforderungen auch Rand- und Nutzungsbedingungen gehören, unter welchen die Funktionalität eines technischen Systems gewährleistet werden muss [77]. Somit umfassen Anforderungen i. w. S. einerseits aus externer Sicht die von Kunden gestellten Forderungen und Wünschen und gesetzlichen und normativen Vorgaben sowie andererseits aus interner Sicht die vom Unternehmen geforderten Vorgaben und Randbedingungen (vgl. z. B. interne Richtlinien und

vorhandenen Ressourcen in Entwicklung, Produktmanagement, Produktion) (vgl. [323], [437]). Zudem zählen zu den wichtigsten impliziten Anforderungen nach BREIING & KNOSALA die sog. Design for X (DfX) Gerechtheiten [77], die nach BAUER in wechselseitigen Relationen zueinanderstehen [36]. Die große Vielzahl und Vielfalt zu berücksichtigten DfX-Aspekten mit zahlreichen, möglichen Hilfsmitteln, wie z. B. Richtlinien, Methoden und IT-Werkzeugen, führt zu einer beträchtlichen Komplexität des Design for X [36]. Darüber hinaus sind auch viele und verschiedene Relationen, wie z. B. konkurrierende, komplementäre und indifferente, Beziehungen, zwischen Anforderungen zu beachten [36], [437]. Dadurch wird die **strukturelle Anforderungskomplexität** ersichtlich.

Da im Zeitablauf des PEP die Anforderungen zur Fokussierung von Entwicklungsprozessen nicht nur zu priorisieren sind, sondern diese entlang der Funktions-, Physik-, Gestalt- und Produktionsebene (s. Bild 102 in Anhang A.6) auch immer weiter zu konkretisieren sind (vgl. u. a. [437]), verändern sich neben den Anforderungen auch die Relationen zwischen diesen (Veränderlichkeit). Da trotz der meist umfangreichen Spezifikationen (vgl. Lasten- und Pflichtenheft) viele Anforderungen noch einige Freiheitsgrade haben (müssen) und die Relationen zwischen diesen v. a. am Anfang der Entwicklung noch teils unscharf und manchmal auch noch unbekannt sind, kann auch von einer Vieldeutigkeit bezüglich der Anforderungen ausgegangen werden. Demgemäß wird die **funktionale Anforderungskomplexität** deutlich. Darüber hinaus sei angemerkt, dass diese zunehmende Anforderungskomplexität auch ein Resultat der oft zahlreichen und vielfältigen Kunden bzw. beteiligten Stakeholder an der PE ist (vgl. z. B. Ausführungen zur Kundenkomplexität in [3]). Daher ist für eine kontinuierliche Weiterentwicklung von Anforderungen über den gesamten Entwicklungsprozess auch eine strukturierte Vorgehensweise erforderlich (vgl. u. a. [437], [S31]).

Produktkomplexität

Die Erfüllung der Kundenanforderungen mit dem vorhandenen Produktportfolio und in den bekannten oder noch zu erschließenden Märkten zu wachsen ist neben der Entwicklung neuartiger, innovativer Produkte für bekannte/neue Märkte ein zentrales Ziel von Unternehmen (vgl. Kapitel 1). Ein Produkt kann in Bezug auf eine erwartete Bedürfnisbefriedigung bei einem Nutzer (Nachfrager) als eine von einem Unternehmen (Anbieter) gebündelte Menge an (Produkt-)Eigenschaften verstanden werden, das i. d. R. zu einem gewissen Preis getauscht wird, welcher in Folge der Präferenzbewertung durch Kunden und des Zusammenspiels von Angebots-

und Nachfragebedingungen entsteht [82]. Durch die starke Kundenorientierung und der damit einhergehenden Erweiterung des Produktportfolios mit kundenspezifischen – oft individuellen Lösungen – haben v. a. die zu entwickelnden Produkte immer mehr und vielfältigere Funktionen, wie z. B. Hardware- und Softwarekomponenten, und werden somit zunehmend komplexer (vgl. [401]). Neben der Entwicklung von mechanischen und mechatronischen Produkten werden zunehmend auch (digitale) Services wichtiger, wodurch die Komplexität dieser Produkte im Sinne von hybriden Leistungsbündel (d. h. Produkt mit Sach- und Dienstleistungsanteilen) weiter steigt (vgl. u. a. [92], [94]). Daraus resultiert die steigende **strukturelle Produktkomplexität** (vgl. auch technische Systeme in Bild 104 die sich durch die jeweils entsprechenden, miteinander zusammenhängenden Bestandteile der Produktkomplexität auf Funktions-, Physik-, Gestalt-, und Produktionsebene ausdrücken lassen (vgl. Bild 102 und Bild 103 in Anhang A.6).

SCHUH & RIESENER, die Produktkomplexität prinzipiell aus einer externen Sicht (i. S. v. Anforderungskomplexität) und einer internen Sicht (d. h. i. e. S. von Produktkomplexität) betrachten, erläutern die Auswirkungen von Projekten zur Kostenoptimierung, von Lieferantenwechseln und von global verteilten Produktionsnetzwerken auf die Produktkomplexität [493]. Diese Auswirkungen, die i. d. R. durch Unternehmensziele getriggert werden, führen neben den sich (häufig) ändernden Anforderungen, wie z. B. von Kunden und verändernden Gesetzen, zu einer verändernden Funktionsvielfalt. Diese Funktionsvielfalt führt zu einer zunehmenden Vielzahl und Vielfalt auf Endprodukt- sowie ebenfalls auf Baugruppen- und Bauteilebene (vgl. Bild 104) [603], was starke Effekte auf alle Bereiche der Wertschöpfungskette (vgl. Bild 103) [141] sowie auch auf weitere Komplexitätsarten (z. B. Prozess- und Organisationskomplexität) innerhalb des Unternehmens zur Folge hat [529]. Dies führt auch zusammen mit den sich verkürzenden Produktlebenszykluszeiten zu Produkten, die einer hohen Veränderlichkeit unterliegen, sowie zu diversen Transparenzproblemen im Hinblick auf die diversen Produktelemente und deren Relationen (Vieldeutigkeit) [301], wodurch sich auch die zunehmende **funktionale Produktkomplexität** begründen lässt.

Für die Produktkomplexität ist der **Produktingenieur** (vgl. Bild 3 in Kapitel 1.4), der vergleichbar mit dem in [151] beschriebenen (klassischen) Entwicklungskonstrukteur ist, hautverantwortlich. Vertiefende Erläuterungen zu verschiedenen Rollen und Berufsbildern insb. in der Konstruktion und Fertigungsvorbereitung sind in [151] zu finden.

Prozesskomplexität

Mit der Zunahme der Produktkomplexität geht auch ein entsprechender Anstieg der **strukturellen Prozesskomplexität** einher (vgl. u. a. [160], [326], [497]), da die Prozesskomplexität meist als ein produktinduziertes Phänomen angesehen werden kann (z. B. Prozess des Anforderungsmanagements) [3]. In Folge dieser Wechselwirkung zwischen Produkt- und Prozesskomplexität führen die zunehmend komplexer werdenden Produkte (z. B. mehr Varianten) zu entsprechend komplexeren Prozessen, die in [26] als vielfaltsinduzierte Komplexität in Entwicklungsprozessen bezeichnet wird. Daher lässt sich die Prozesskomplexität durch eine starke Vernetzung verschiedener Prozesselemente mit vielen internen wie auch externen Schnittstellen charakterisieren [443] (vgl. auch Simultaneous Engineering und Concurrent Engineering in [151] und am Beispiel der Automobilindustrie [236]). Zudem ist – im Vergleich zu allen anderen Unternehmensprozessen – die Intensität der (vielfaltsinduzierten) Prozesskomplexität in der PE am höchsten [26], woraus meist ein Höchstmaß an Koordinationskomplexität resultiert [3].

Diese hohe Koordinationskomplexität von Entwicklungsprozessen liegt u. a. auch an der hohen Dynamik der vielen, verschiedenen Prozesselemente und deren oft wechselseitigen Abhängigkeiten sowie der häufig mehrdimensionalen Zielerwartung (z. B. bzgl. Qualität, Kosten und Zeit) an Prozessen begründet (vgl. [443]). Zudem lässt sich in der PE ein Anstieg zum einen der Interaktionshäufigkeit zwischen den beteiligten Abteilungen auch aufgrund verkürzter Produktlebens- und Technologiezyklen feststellen zum anderen der Anpassungshäufigkeit der Entwicklungsprozesse erkennen [301]. Deshalb ist es z. B. für flexible Anpassungs- und Abstimmungsprozesse erforderlich, dass diese Prozesse gewisse Freiheitsgrade aufweisen und die Beziehungen zwischen diesen eine gewisse Unschärfe haben (vgl. v. a. agile PE i. w. S. in [487] und für physische PE in [366]). Demnach wird die **funktionale Prozesskomplexität** unverkennbar.

Der **Prozessingenieur** (vgl. Bild 3 in Kapitel 1.4) trägt v. a. die Verantwortung für die Prozesskomplexität. Hilfreiche Beschreibungen von zentralen Rollen des Prozessmanagements und der Prozessgestaltung und -modellierung sind v. a. in [201] und [332] (s. auch [46], [261], [473], [485], [510]) zu finden.

Wissenskomplexität

Das für die PE erforderliche Wissen bzw. Informationen über bspw. Kunden, Anforderungen, Produkte und Prozesse aus diversen Wissensquellen

(z. B. PDM- und ERP-Systemen, Datenbanken, Internet) zu beschaffen, kann als **strukturelle Wissenskomplexität** bezeichnet werden (s. Diplomarbeit des Autors für Details [P6]). Die Vielzahl und Vielfalt der Wissens- und Informationselemente wird insb. auch vor dem Hintergrund einer steigenden Wissens- und Informationsflut deutlich, wobei nach [85] und [121] auch die steigende Wissenshalbwertszeit (d. h. die Zeitdauer, in der 50% des erarbeiteten Wissens überholt ist) zu beachten ist. So lässt sich besonders auch in der PE eine zunehmende Vielzahl und Vielfalt an neu erzeugten Daten und Informationen feststellen, wobei auch durch die steigende Produktkomplexität ein höherer Grad an Vernetzung dieser Daten erforderlich wird [341].

Die Bedeutung und Auswirkung der Veränderlichkeit und Vieldeutigkeit von Wissen in der PE wird bspw. auch durch eine US-amerikanische Studie deutlich, nach der Ingenieure zwar (lediglich) zehn Prozent ihrer durchschnittlichen Arbeitszeit mit der Informationssuche, aber fast 40% mit dem Erfragen von Wissen bei Fachexperten und rund 45% mit der Wissensinterpretation und -adaption beschäftigt sind [275] (s. [251], [P6] für Details). Daher wird die Fähigkeit Wissen nicht nur zu schaffen, sondern auch verschiedene Wissensarten neu zu kombinieren bzw. in Relation zu setzen, kontextspezifische und effizient zu verwalten sowie in andere Arten des Wissens zu transformieren, immer wichtiger zur Entwicklung von Produkten, die die gegenwärtigen und zukünftigen Bedürfnisse der Kunden befriedigen [409]. So ist für die Entwicklung neuer Produkte sowie für die Lösung schwieriger Probleme insb. Faktenwissen (z. B. Grundkenntnisse über Mechanik, Physik), Methodenwissen (z. B. Entwicklungs- und Berechnungsmethoden) und heuristische Kompetenz (z. B. zielgerichtete Kreativität, Steuerungsfähigkeit) erforderlich [151], [549]. Da v. a. implizites Wissen (z. B. von Experten) zum einen i. d. R. veränderlich ist bzw. sich aufgrund neuer Erkenntnisse weiterentwickeln kann und zum anderen meist unscharfe und unsichere Beziehungen zwischen diversen Wissensselementen vorliegt, kann auch von einer **funktionalen Wissenskomplexität** in der Entwicklung ausgegangen werden.

Der **Wissensingenieur** (vgl. Bild 3 in Kapitel 1.4) ist für das Wissensmanagement und damit für den Umgang der Wissenskomplexität verantwortlich. Für ausführliche Beschreibungen von Rollen mit deren spezifischen Kompetenzen, Aufgaben, Verantwortlichkeiten und Beziehungen im Wissensmanagement – v. a. die des Wissensingenieurs – sei an dieser Stelle auf die Arbeiten des Autors [P7], [P20], [P24] verwiesen (s. für weitergehende Erläuterungen [59], [345], [409], [439], [408], [501], [577], [588]).

Organisationskomplexität

In Folge der zuvor beschriebenen Komplexitätsarten und zusätzlich auch aufgrund der Befriedigung regionaler Anforderungen von Kunden und Ländern bzw. Regionen sowie kultureller Anpassungen steigt ebenfalls die Organisationskomplexität in der PE [602]. Im organisatorischen Kontext ergibt sich die **strukturelle Organisationskomplexität** aus der Anzahl und Vielfalt an Personen und Organisationseinheiten sowie deren verschiedenen Interaktionen innerhalb und außerhalb des Unternehmens. Werden z. B. Personen verschiedenen Organisationseinheiten zugeordnet, ergeben sich entsprechend der gewählten Strukturierungsform (z. B. hierarchisch oder funktional) unterschiedliche Organisationsformen (z. B. Linienorganisation oder Team-/Projektstrukturen) [602]. Dabei lassen sich den Personen bzw. Organisationsstrukturen (d. h. der Aufbaustruktur) z. B. Methoden und Werkzeuge (vgl. Bild 105) zuordnen [602]. Infolgedessen steigt v. a. die Varietät mit der Anzahl und Vielfalt (z. B. Fachabteilungen, Bildung, Landeskulturen) der Personen, die sich für eine effektive sowie effiziente Zusammenarbeit untereinander abstimmen müssen bzw. zu führen sind. Die Arbeitsteilung erfordert v. a. bei vernetzten Abläufen der PE eine verstärkte Kooperation und Koordination zwischen den verteilten Organisationseinheiten [602]. Dabei ist bspw. zwischen der Abstimmung und Koordination von Personen auf gleichen und auf verschiedenen Hierarchiestufen zu unterscheiden, wobei mit steigender Anzahl an Hierarchiestufen die Organisationskomplexität steigt. Neben den hierarchischen, disziplinarischen Weisungsbefugnissen gibt es eine Vielzahl weiterer und verschiedener Verbindungen (z. B. Wissens- und Informationsaustausch, Zugehörigkeit zu einem interdisziplinären Projekt) sowie Kommunikationsbeziehungen zwischen Personen (Konnektivität). Bei einem bestimmten Grad von Komplexität, wie z. B. Produkt- und Wissenskomplexität, ergeben sich oft netzwerkartige Strukturen von kooperierenden Unternehmen, die nicht grundsätzlich von Vorteil für die PE Innovationen sind [81].

Die Veränderlichkeit von Personen und Verbindungen zwischen diesen hat einen Einfluss auf die **funktionalen Organisationskomplexität**. Hierfür sind häufige Veränderungen von Bezugspersonen, mehrfache Wechsel von (weisungsberechtigten) Vorgesetzten sowie zunehmender Wandel von Organisationsstrukturen anschauliche Beispiele. So hat eine Reorganisation tiefgreifende und umfassende Veränderungen der bestehenden Organisationsstruktur zur Folge [517]. Reorganisationen sind oft eine Reaktion der Unternehmensführung auf die Umweltdynamik, also häufige Änderungen der Märkte und der Unternehmensumwelt, und auf die Unsicherheit, die sich auf betriebliche Planungen bezieht und durch Informationsdefizite

entsteht [517]. Daher ist eine Wandlungsfähigkeit von Organisationen zwingend erforderlich und somit sollten organisatorische Veränderungen fester Bestandteil der Entwicklung von Organisationen sein [29], wie z. B. im Rahmen von Internationalisierungs- und Dezentralisierungsbestrebungen des Unternehmens [468]. Damit einher geht auch eine der Kernbotschaften zur agilen Organisation, nach der Organisationen oft unter einer zu geringen Anpassungsfähigkeit leiden und eine hohe Anpassungsfähigkeit insb. in einer VUCA-Welt erforderlich ist [471]. Zusätzlich zur steigenden Dynamik in den meisten Organisationsformen nimmt auch die Vieldeutigkeit zu, die sich bspw. aus mehr Freiheit, Entfaltungsmöglichkeiten sowie selbstständiges Arbeiten von Mitarbeitern in Folge von flacheren Hierarchien ergibt (vgl. u. a. [191], [287], [144]). Dies geht oft auch mit teilweise unklaren Kommunikationsbeziehungen und Machtstrukturen in inter- und multidisziplinären Projektarbeiten einher. Insb. beim Lösen von Schadenfällen und Entwicklungsproblemen ist aufgrund der vielen vieldeutigen, ungewissen und unbekannten eine Zusammenarbeit zwischen vielen, verschiedenen internen und externen Personen erforderlich. Dies zeigt auch eine empirische Studie, nach welcher Entwickler zwar ca. 70% ihrer Arbeitszeit allein arbeiten, aber bei Konstruktionsproblemen bzw. kritischen Situationen in fast 90% mit ihren Kollegen kommunizieren und kooperieren [190]. Dabei treten nach FRANKENBERGER bereits bei relativ einfachen Entwicklungsaufgaben Probleme in der Zusammenarbeit auf [190]. Schwierigkeiten in der Kooperation mit externen Partnern werden auch in einer Untersuchung von EHRENSPIEL als ein großes Defizit thematisiert [147].

Zudem sei darauf hingewiesen, dass in [567] das Berufsbild des Organisators beschrieben wird, welches vergleichbar mit der Rolle des **Organisationsingenieurs** (vgl. Bild 3 in Kapitel 1.4) ist. Vertiefende Literatur ist in [29], [344], [558], [602], [605] zu finden.

Komplexitätskosten

Eine Folge der zuvor skizzierten Komplexitätsarten sind steigende Kosten, die häufig als Vielfalts-, Varianten- oder **Komplexitätskosten** bezeichnet werden [38], [598]. Da Komplexität (s. Kapitel 2.2.2) und infolgedessen auch die damit zusammenhängenden Komplexitätskosten schwer messbar sind [512], werden nach BATTENFELD Komplexitätskosten mittels einer relativen Betrachtungsweise wie folgt definiert: Komplexitätskosten sind diejenigen monetären Aufwendungen, die durch den Komplexitätsanstieg durch eine zusätzliche Variante verursacht werden [35]. Dagegen wird in [34] explizit zwischen Komplexitäts- und Variantenkosten differenziert.

Demnach sind Variantenkosten diejenigen Kosten, die u. a. durch die Entwicklung, Produktion und Verkauf von nicht Standardprodukte entstehen (vgl. hohe Ähnlichkeit zur Definition gemäß [35]), und Komplexitätskosten diejenigen Kosten, die durch jede Art von Komplexität entstehen und damit die Variantenkosten umfassen (vgl. auch [375]). Neben diesen beiden Begriffsdefinitionen von Komplexitätskosten lassen sich weitere Begriffsbestimmungen und somit keinen Konsens in der Forschung (vgl. [71], [91] für Details). In dieser Arbeit wird der Definition von Komplexitätskosten gemäß RATHNOW gefolgt, nach der Komplexitätskosten alle Kosten umfassen, die durch die interne Komplexität für das Unternehmen verursacht werden [447].

In Folge der externen Komplexität (vgl. Bild 105), wie individueller Kundenwünsche und globaler Wettbewerbsdruck, versuchen Unternehmen durch die zunehmende Ausdifferenzierung ihres Produktportfolios mit steigender Variantenvielfalt immer kleinere Nischen abzudecken [71]. Die damit einhergehende Spirale steigender interner Komplexität und der daraus resultierenden Komplexitätskosten ist für Unternehmen langfristig erfolgswirtschaftlich gefährlich, da dem Komplexitätskostenanstieg in geringerer Höhe Erlössteigerung gegenüberstehen (s. [71], [176] für Details und [206] für Beispiele). Dabei bestehen die (Selbst)Kosten eines Unternehmens je nach Sichtweise aus direkten und indirekten Kosten bzw. Einzel- und Gemeinkosten, die jeweils fixe und variable Kostenarten umfassen [152]. In fast allen diesen Kostenarten können sich (anteilig) auch Komplexitätskosten verbergen [206]. Während direkte Komplexitätskosten die Gemeinkosten proportional zum Produkt erhöhen, treten indirekte Komplexitätskosten v. a. in Form von Opportunitätskosten auf, falls Ressourcen im Unternehmen nicht bestmöglich genutzt werden können [264] (z. B. Verringerung von Skaleneffekten aufgrund geringerer Losgrößen [301]). Opportunitätskosten entstehen bspw. dadurch, dass ein großer Teil der Entwicklungsstunden für kundenspezifische Anpassungen (vgl. Variantenkonstruktion) anstatt für echte Innovationen (vgl. Neukonstruktion) aufgewendet wird [66]. Direkte Komplexitätskosten können weiter in einmalige (z. B. Entwicklung und Konstruktion zusätzlicher Teile, neue Variantenmerkmale [598]) und dauerhafte direkte Komplexitätskosten (z. B. Pflege und Verwaltung der Produkt- und Preisdokumentation [598]) gegliedert werden [264]. Aufgrund der Vielzahl und Vielfalt von Komplexitätskostenarten und deren Relationen untereinander und mit anderen Kostenarten kann daher auch von steigenden **strukturellen Komplexitätskosten** gesprochen werden.

Die **funktionalen Komplexitätskosten** lassen sich weitgehend mit den vier Eigenschaften von Komplexitätskosten nach RATHNOW erläutern [447]. Diese Eigenschaften von Komplexitätskosten werden in Bild 106 überblickartig dargestellt.

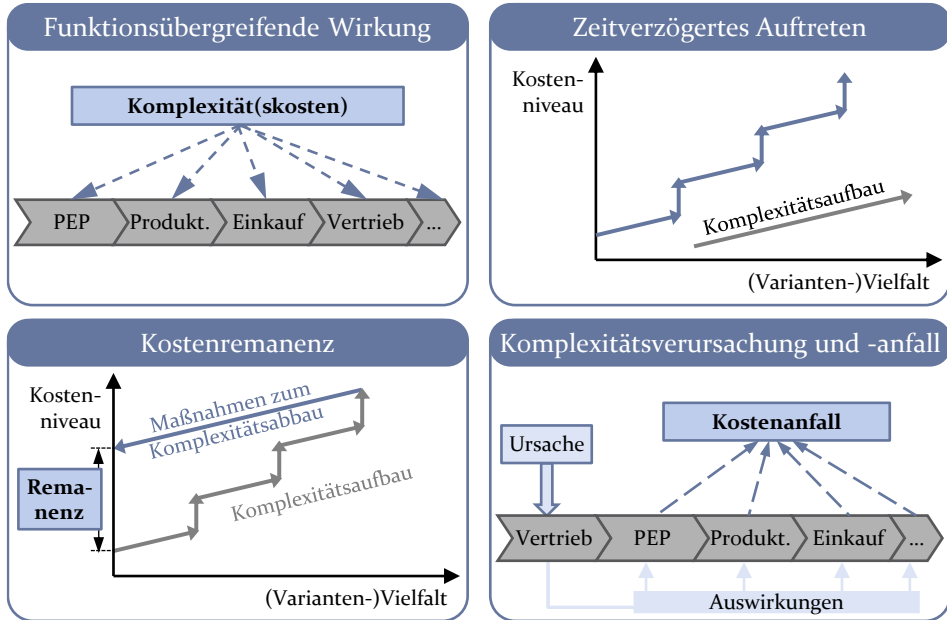


Bild 106: Eigenschaften von Komplexitätskosten (in Anlehnung an [447] und z. T. ergänzt nach [206])

Diese vier Eigenschaften von Komplexitätskosten nach RATHNOW werden nachfolgend kurz zusammengefasst [447] (vgl. auch [206], [598] für Details und Beispiele):

- Funktions- bzw. lebensphasenübergreifende Wirkung: Komplexitätskosten aufgrund vielfalts- bzw. produktvarianzinduzierter Komplexität haben Auswirkungen auf fast alle Funktionsbereiche und auf den gesamten Produktlebenszyklus. Entscheidungen bzgl. neuer Varianten werden i. d. R. zu Beginn des PEPs getroffen (vgl. Kapitel 1.2), was zum einen zur Folge hat, dass die verschiedenen anderen Funktionsbereiche von den potentiellen Auswirkungen der zusätzlichen Varianz noch nicht betroffen sind bzw. diese oft nur schwer abschätzen können, was zum anderen aber das größte Potential hat, auf die Lebenszykluskosten Einfluss zu nehmen (vgl. Bild 108) [179]. Zudem wird durch die Zuordnung der Auswirkungen in einzelnen Kostenstellen wird die Gesamtwirkung der

Komplexität im Unternehmen häufig nicht wahrgenommen oder unterschätzt [206] (vgl. auch [151]).

- Zeitverzögertes Auftreten: Komplexitätskosten weisen ein zeitverzögertes Auftreten. Während bei geringer Veränderung der Vielfalt der auftretende Mehraufwand von den verfügbaren Ressourcen noch mitgetragen werden kann, sind ab einer bestimmten bzw. kritischen Anzahl zusätzlicher Varianten zusätzliche Ressourcen erforderlich (z. B. Maschinen, Mitarbeiter). Dies ist mit einem sprunghaften Anstieg der Kosten verbunden.
- Kostenremanenz: Komplexitätskosten verhalten sich bei Komplexitätsaufbau und anschließender Komplexitätsreduktion remanent (asymmetrisch dynamisches Verhalten). Dies liegt daran, dass zuvor aufgebaut Ressourcen (z. B. Maschinen) aufgrund gestiegener Produktvielfalt bei einer Reduzierung der Komplexität meist nicht gleichermaßen abgebaut werden können und infolgedessen das ursprüngliche Kostenniveau i. d. R. nicht erreicht werden kann.
- Auseinanderfall von Komplexitätsverursachung und Kostenanfall: Komplexitätskosten lassen dadurch charakterisieren, dass ihrer Verursacher und Kostenträger i. d. R. nicht identisch sind. So kann bspw. eine vom Vertrieb gewünschte Produktvariante Kosten verursachen, die in der PE, der Produktion und im Einkauf zu (zusätzlichen) Komplexitätskosten führen.

Wie bereits im Zitat von MENCKEN [385] auf den Punkt gebracht, gibt es zwar für jedes komplexe Problem eine einfache Lösung, die aber i. d. R. falsch bzw. nicht zielführend ist. So sind auch die verschiedenen Komplexitätsarten aufgrund ihrer Kosten- und Nutzendimension für Unternehmen – v. a. in der PE – ein für den Unternehmenserfolg relevanter und **komplexer Zielkonflikt** und oft ein wesentliches Dilemma der Unternehmensführung [66]. Dieser Zielkonflikt ergibt sich durch die meist konträren Ziele der Funktionen Vertrieb und Marketing auf der einen und Entwicklung, Produktion und Einkauf auf der anderen Seite (vgl. [381]):

- Vertrieb- und Marketingfunktionen verfolgen durch ihre Markt- und Umsatzorientierung das Ziel möglichst viele Kundenbedürfnisse durch eine hohe Produktvielfalt zu befriedigen, woraus i. d. R. eine Erhöhung der internen Komplexität folgt. Daher ist die wesentliche Zielgröße dieser beiden Funktionen i. d. R. der Umsatz und dessen Steigerung sowie der Marktanteil.
- Entwicklungs-, Produktions- und Einkaufsfunktionen verfolgen durch ihre Kostenorientierung das Ziel möglichst geringe Kosten

bzw. Stückkosten durch standardisierte Produkte zu realisieren, woraus i.d.R. eine Reduktion der internen Komplexität folgt. Daher ist die wesentliche Zielgröße dieser Funktionen i. d. R. die Kosten und deren Senkung sowie Durchlaufzeit und die Produktqualität.

Dieser meist komplexe Zielkonflikt bzw. klassische Interessengegensatz zwischen Umsatzorientierung des Vertriebs- und Marketings einerseits und Kostenorientierung der Entwicklung, Produktion und des Einkaufs andererseits kann dadurch entschieden werden, dass nur diejenigen neuen Produktvarianten in das Produktportfolio integriert werden, bei welchen nachgewiesen ist (z. B. durch Marktforschung), dass der geplante Anstieg der Stückkosten bzw. Kosten durch eine entsprechende Erhöhung des Preises überkompensiert wird [381]. Somit erfordert die **Entscheidung bzgl. der (kosten)optimalen Produktvielfalt** stets eine exakte Abwägung der mit einer zusätzlichen Produktvariante realisierbaren Mehrerlöse und der zusätzlich verursachten Kosten [382] (s. Bild 107). Dabei ist v. a. auch die Zunahme der Komplexitätskosten in den indirekten Bereichen zu berücksichtigen, die sich nur schwer den einzelnen (neuen) Produktvarianten zuordnen lassen [598]. Diese daher oft fehlende Allokation der Komplexitätskosten führt häufig zu Intransparenz bzgl. der Auswirkungen wachsender interner Komplexität [264]. Demgemäß geht mit der optimalen Produktvielfalt bzw. dem Produktdifferenzierungsgrad auch der Komplexitätsgrad (s. Kapitel 2.2.2) einher.

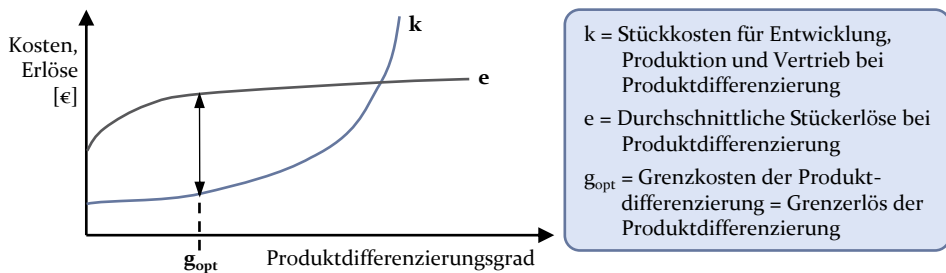


Bild 107: Kostenoptimale Produktvielfalt (in Anlehnung an [30], [382])

Am Ende der Wertschöpfungskette ist es der Vertrieb, der die Einführung kundenindividueller Produktvarianten oder Verpackungen zu verantworten hat. Demnach dürfen individualisierte Produkte nur dann angeboten werden, wenn der Kunde bereit ist die zusätzlichen Kosten der zunehmenden Komplexität zu tragen [441].

Treiber der Komplexität

Aufbauend auf der Differenzierung zwischen den externen (exogenen) und den internen (endogenen) Komplexitätsarten, liegt der Fokus im Folgenden auf den vielfältigen Ursachen von Komplexität, den **Komplexitätstreibern**. Der Unterscheidung gemäß [47] zwischen verschiedenen, symptomatischen Komplexitätsphänomenen und entsprechenden Komplexitätstreibern folgend, wird in dieser Arbeit in Anlehnung an [432] unter Komplexitätstreiber ein Phänomen verstanden, das durch einen bestimmten Beitrag die Komplexität in einem (Teil-)System beeinflusst.

Grundsätzlich wird zwischen externen und internen Komplexitätstreibern differenziert, wobei zwischen diesen beiden Arten jeweils enge dynamische Wechselwirkungen und Interaktionen bestehen. Dabei wirken externe Treiber von außen auf das Unternehmen bzw. die PE ein und lassen sich nicht oder nur sehr schwierig und wenig beeinflussen [480]. Die interne Komplexität resultiert häufig aus der extern induzierten Komplexität [91]. Die unternehmensinterne Komplexität kann in **korrelierte und autonome Komplexität** unterteilt werden [66] (s. auch [614], [91]):

- Die korrelierte Komplexität wird unmittelbar durch die externe Komplexität (v. a. Marktkomplexität) beeinflusst. Demnach liegt häufig eine direkte Korrelation zwischen der steigenden externen Komplexität (z. B. Kundenkomplexität, Wettbewerb) und den unternehmensinternen, korrelierenden Komplexitätstreibern vor (z. B. Kundenstrukturkomplexität hat einen entscheidenden Einfluss auf das CRM-System). So werden durch externe Komplexitätstreiber, wie z. B. Anforderungskomplexität, die Entscheidungen bzgl. zukünftiger Produkte und Varianten signifikant beeinflusst, wodurch die Produktkomplexität als wesentlicher Treiber der korrelierenden Komplexität angesehen werden kann (vgl. auch [216]). Außerdem kann eine Modifikation der Wertschöpfungstiefe eines Unternehmens bei seinen Lieferanten zu einer Veränderung der Komplexität führen. Daher können sowohl interne als externe Treiber Ursachen der korrelierten Komplexität sein.
- Die autonome Komplexität wird i. d. R. nicht durch die externe Komplexität beeinflusst, sondern von innerbetrieblichen Faktoren, wie z. B. der Organisationsstruktur (Organisationskomplexität) und der Prozessstruktur (Prozesskomplexität), bestimmt. Dabei bestehen auch Beziehungen zur korrelierenden Komplexität. So kann bspw. der steigenden Produktindividualisierung mittels einer steigenden Variantenanzahl begegnet werden, was i. d. R. zu einer

steigenden Prozess- und Organisationskomplexität führt. Dabei ist die prinzipiell zunehmende Komplexität mit Hilfe einer möglichst geschickten Modularisierung und Standardisierung von Produkten und Prozessen überproportional zu reduzieren.

Da somit die autonome und korrelierte Komplexität wechselseitig zueinander in Beziehung stehen, dient deren differenzierte Betrachtung aus wissenschaftlicher Perspektive v. a. deren formalen, logischen Abgrenzung [66]. In Bild 12 ist ein **Überblick über wesentliche externe und interne Komplexitätstreiber** zu finden, wobei auch zwischen den korrelierten und autonomen Komplexitätstreibern unterschieden wird. Dabei besteht v. a. eine direkte oder zumindest indirekte Korrelation zwischen der Marktkomplexität, wie bspw. der Kunden-, Wettbewerbs- und Beschaffungskomplexität, auf der einen Seite und den internen, korrelierten Komplexitätstreibern, wie bspw. Anforderungskomplexität, Ziel- und Kostenkomplexität (inkl. Komplexitätskosten), Portfolio- und Produktkomplexität sowie weiteren Komplexitäten (z. B. Kundenkomplexität, Technologiekomplexität) auf der anderen Seite. Die im Vergleich zu den korrelierenden Komplexitätstreiber weniger von den externen Komplexitätstreibern abhängigen autonomen Komplexitätstreiber sind u. a. die Prozess-, Wissens- und Organisationskomplexität und weitere Komplexitäten (vgl. Bild 12 in Kapitel 2.2.2). In diesem Bild 12 sind die im Fokus dieser Arbeit stehenden Domänen mit den entsprechenden Farben eingefärbt.

Die Wechselwirkungen zwischen externen Komplexitätstreibern, wie der Gesellschafts- und Marktkomplexität, einerseits und der Produktkomplexität andererseits werden u. a. von MARTIN & ISHII in einigen Publikationen unter dem Begriff Design for Variety ausführlich aufgezeigt (vgl. u. a. [370], [371] und für Details [369]). Demnach gehören zu den relevanten externen Treibern von Produktkomplexität u. a. wechselnde Performancebedürfnisse (z. B. Gewicht, Größe, Design), neue Umweltauflagen, verändernde Regularien und fallende Preise [371]. Die dadurch steigende Produktvarianz führt zu höheren Lagerbeständen und -kosten (vgl. [370] und für Details [369]). Weitere Erläuterungen zu den einzelnen Komplexitätstreibern und deren Wechselwirkungen sind u. a. in [91], [480], [66], [216], [300], [614], [84] zu finden.

Grade der Komplexität

Im Rahmen der Begriffserklärung von Komplexität wird häufig auch der **Begriff der Kompliziertheit** verwendet (z. B. im Zusammenhang mit Komplexitätsgraden). Dabei wird Komplexität in der Alltagssprache bzw. umgangssprachlich häufig mit undurchschaubar, schwierig, unerklärlich,

unverständlich und auch mit kompliziert assoziiert [300], [91]. Jedoch sind diese Adjektive im Rahmen des Komplexitätsmanagement keinesfalls als Synonyme zu verwenden, wie auch das vorhergehende Kapitel 2.2.1 gezeigt hat. Allerdings wird auch der Begriff Kompliziertheit, der im Folgenden im Hinblick auf komplizierte Systeme noch einmal aufgegriffen wird, in der Literatur nicht einheitlich definiert und verwendet (vgl. u. a. [442]).

So differenzieren EHRENSPIEL & MEERKAMM zwischen Komplexität und Kompliziertheit, indem sie Kompliziertheit als Maß für die **subjektive Schwierigkeit** beim Umgang mit (komplexen) technischen Systemen (nach dem sehr vereinfachten, technischen Komplexitätsdefinition, vgl. Kapitel 2.2.1 und Bild 9) beschreiben [151]. Diese bedeutet, dass identische Systeme von verschiedenen Personen als unterschiedlich kompliziert (i. S. v. schwierig, herausfordernd) wahrgenommen werden, wie z. B. die Reparatur eines technischen Produkts (s. Bild 104) durch einen Laien im Vergleich zu einem Experten [151].

Dagegen unterscheiden andere Forscher zwischen Kompliziertheit und Komplexität auf eine andere Art und Weise (s. u. a. [478], [180], [91], [105], [565], [479]). Demnach geht die **Komplexität über die Kompliziertheit hinaus** [91], d. h., dass komplexe Systeme auch alle Aspekte komplizierter Systeme erfüllen [478]. Dies bedeutet nach ULRICH & PROBST, dass etwas komplex ist, falls ein System nicht nur in seiner Zusammensetzung kompliziert ist (vgl. sehr vereinfachte, technische Komplexitätsdefinition, vgl. Kapitel 2.2.1 und Bild 9) sondern auch durch Dynamik (i. S. v. Zustandsänderungen) gekennzeichnet ist [565]. Demzufolge sind Lösungen und Prozesse in komplexen Systemen meist schwer oder nicht vorhersagbar [478], was u. a. an der vorliegenden Unbestimmtheit und Unsicherheit von Elementen und/oder Relationen liegen kann [480]. Im Gegensatz dazu haben komplizierte Systeme bei bekannten Ausgangsbedingungen und Systemvariablen nur einen – vereinfacht ausgedrückt – linearen Zusammenhang und sind somit vorhersagbar [480]. Daher lässt sich schlussfolgern, dass für das Management von komplexen Systemen ein anderes Verständnis als bei komplizierten Systemen erforderlich ist (vgl. auch [480]).

Aus diesem Grund ist es sinnvoll, für das zu managende, komplexe System im Vorfeld eine Klassifikation vorzunehmen [480] (vgl. Anhang A.5) und Rolle des Komplexitätsingenieurs in Kapitel 1.4). Zu diesem Zweck kann das von ULRICH & PROBST entwickelte Kategorisierungsschema verwendet werden, mit dem die Zusammenhänge und Unterschiede von einfachen, komplizierten, relativ komplexen und äußerst komplexen Systemen deutlich werden [565]. Diese Einordnung komplexer Systeme nach ULRICH & PROBST

[565] (s. Bild 11) ist in der Literatur oft referenziert (vgl. [491], [480], [91]) und wird auch zur **Verortung komplexer Systeme in dieser Arbeit** herangezogen. Mit dieser Einordnung werden nach SCHOENEBERG die vier Abstufungen einfache, komplizierte, relativ komplexe und äußerst komplexe Systeme aus wissenschaftlicher Sicht voneinander abgegrenzt [480], womit auch ein zunehmender Grad der Komplexität einher geht. Wie in Bild 11 dargestellt, sind mit der Abszissenachse die beiden Komplexitätsdimensionen Vielzahl und Vielfalt und mit der Ordinatenachse die Komplexitätsdimension Veränderlichkeit (Dynamik) berücksichtigt (vgl. Bild 10), wobei die Vieldeutigkeit in dieser Klassifikation weitgehend nicht einbezogen wird. Im Hinblick auf die horizontale Achse steht die Art der Zusammensetzung und damit die Kompliziertheit des Systems im Fokus, d. h. zum einen Anzahl und Verschiedenheit der Elemente und zum anderen Anzahl und Verschiedenheit der Relationen zwischen den Elementen (vgl. [565] und Bild 9). In Bezug auf die vertikale Achse steht die Veränderlichkeit im Zeitverlauf (in Bild 10 durch die Uhr symbolisiert) und damit auch die Komplexität im Fokus, d. h. einerseits die Vielfalt der Verhaltensmöglichkeiten der Elemente und andererseits die Veränderlichkeit der Wirkungsflächen zwischen den Elementen (vgl. [565]). Dabei sind die Übergänge zwischen den vier Stufen der Komplexität bzw. den Komplexitätsgraden fließend [480].

Einfache Systeme lassen sich durch eine geringe Anzahl und Vielfalt an Relationen charakterisieren, wie z. B. durch vier Produktelemente und mit vier Relationen (vgl. Bild 11). Für ein einfaches System kann die Berechnung des Volumens des Rohrs (vgl. Kapitel 1.5) als Beispiel angeführt werden. So lässt sich mit der entsprechenden mathematischen Funktion (s. Anhang A.4) mit den drei Produktmerkmalen Außen- und Innendurchmesser sowie Länge das Rohrvolumen berechnen. In Bild 11 sind diese drei Produktmerkmale, die jeweils eine unidirektionale Relation mit der Produkteigenschaft Volumen haben, dargestellt. Zudem ist auch die bidirektionale Abhängigkeit zwischen Innen- und Außendurchmesser gezeigt. Da sowohl die zur Berechnung des Rohrvolumens erforderlichen Produktmerkmale (lediglich Veränderungen der Ausprägungen, nicht der Verhaltensmöglichkeiten denkbar) als auch deren mathematisch beschreibbare Relation zur Berechnung des Rohrvolumens (sehr geringe Veränderlichkeit der Wirkungsverläufe zwischen den vier Elementen) als sehr stabil angesehen werden kann, handelt es sich um ein einfaches System.

Komplizierte Systeme sind im Gegensatz zu einfachen Systemen durch eine hohe Vielzahl und Vielfalt an Elementen und Relationen sowie ähnlich zu einfachen Systemen durch eine geringe Veränderlichkeit dieser im

Zeitverlauf charakterisierbar. In Bild 11 sind beispielhaft das Produktmerkmal Dichte sowie die beiden Produkteigenschaften Masse und Wandstärke aus der Produktdomäne ergänzt. Darüber hinaus ist das System – gemäß dem Anwendungsszenario in Kapitel 1.5 – um je zwei ausgewählte Prozesselemente (3D-CAD-Modell modellieren, Werkstoff auswählen), Wissensselemente (Fertigungszeichnung, 3D-CAD-Modell), Organisationselemente (Konstrukteur, Einkäufer) sowie zwei Anforderungen (maximales Gewicht, erforderliche Wandstärke) und zwei Kostenelemente (Konstruktionskosten, Einkaufskosten) erweitert worden. Außerdem lässt sich in Bild 11 erkennen, dass sowohl die Anzahl als auch die Vielfalt an Relationen zugenommen hat. So haben beispielsweise die drei Produktmerkmale einen unterschiedlich starken Einfluss auf das Produktvolumen, weshalb die entsprechenden Relationen mit einer unterschiedlich dicken Relation dargestellt sind. Der jeweils „positive“ Einfluss des Produktvolumens und der Dichte auf die Masse des Rohrs wird mit grünen, gerichteten Relationen skizziert. Die zeitlich-logische Abhängigkeit zwischen den Schritten „3D-CAD-Modell modellieren“ und „Werkstoff auswählen“ ist durch eine unidirektionale Relation dargestellt. Analog zum einfachen ist auch beim komplizierten System die Veränderlichkeit im Zeitverlauf gering.

Relativ komplexe Systeme unterscheiden sich von einfachen Systemen durch die Veränderlichkeit im Zeitverlauf, womit einerseits die Vielfalt der Verhaltensmöglichkeiten der Elemente und andererseits die Veränderlichkeit der Wirkungsverläufe zwischen den Elementen (d. h. der Relationen) gemeint sind. Dabei wird die Veränderlichkeit (Dynamik) im Zeitverlauf in Bild 11 durch die schematische Uhrzeit symbolisiert. Im Unterschied zum Beispiel für das einfache System können sich bei einem relativ komplexen System die Ausprägungen der drei Produktmerkmale (i. S. v. Veränderlichkeit im Zeitverlauf) und damit auch das Produktverhalten ändern. Zudem kann z. B. das Produktmerkmal Außendurchmesser „verschwinden“, da dies nicht mehr vom Konstrukteur zu modellieren ist, und das Produktmerkmal Wandstärke „dazu kommen“ (vgl. Bild 11). Ein (verkürztes) Beispiel für eine Veränderlichkeit der „Wirkungsverläufe“ zwischen Elementen ist zum Beispiel, dass für die Modellierung des Rohrs nicht mehr Außen- und Innendurchmesser zu definieren sind, sondern nur noch die Ausprägung des Produktmerkmals Wandstärke vom Konstrukteur festzulegen ist, da in einem zugrunde liegenden, parametrisierten CAD-Modell die Ausprägung für den Innendurchmesser als konstanter Wert determiniert und die Ausprägung des Außendurchmessers das Resultat aus der Ausprägung des Innendurchmessers plus des zu bestimmenden Wertes für die Ausprägung der Wandstärke ist. Grundsätzlich ist es auch denkbar,

dass die Rohrlänge zu einem der beiden Durchmesser oder der Wandstärke in Beziehung gesetzt wird.

Äußerst komplexe Systeme sind wie relativ komplexe Systeme durch eine hohe zeitliche Dynamik als auch wie komplizierte Systeme durch eine hohe Vielzahl und Vielfalt von Elementen und Relationen gekennzeichnet (s. Bild 11). So sind in Bild 11 die hohe Veränderlichkeit der Elemente und Relationen im Zeitverlauf durch die Uhr sowie die Vielzahl und Vielfalt an Elementen und Relationen zum einen durch sechs Produktelemente und jeweils zwei Elemente der fünf weiteren Domänen und zum anderen durch viele, verschiedene Relationen zwischen Elementen dargestellt.

A.9 Auswirkungen von Komplexität auf Kosten

In diesem Anhang wird der **Zusammenhang zwischen Komplexität und Kosten** kurz erläutert. Dazu werden im ersten Abschnitt die Grundlagen zu Kostenfestlegung und Kostenrealisierung in der PE skizziert. Danach werden die Auswirkungen der steigenden Komplexität dargestellt. Abschließend werden insb. die Auswirkungen der Komplexität in der Entwicklung erörtert.

Grundlagen zu Kostenfestlegung und Kostenrealisierung

Um den **Unternehmensgewinn** zu steigern, können zum einen der Erlös erhöht oder die (Selbst-)Kosten reduziert werden [297]. Die Erhöhung des Erlöses kann u. a. durch bedarfsgerechtere, innovativere Produkte, durch die Verbesserung des Vertriebs und Kundenservices sowie durch eine Verkürzung von Lieferzeit und Erhöhung der Produktqualität erfolgen (vgl. auch [152]). Die Reduzierung der (Selbst-)Kosten kann sowohl kurz- bis mittelfristig durch die Rationalisierung des PEPs als auch mittel- bis langfristig durch die Entwicklung kostengünstiger Produkte realisiert werden [152]. Zur Rationalisierung des PEPs gehören Maßnahmen zur effizienteren Gestaltung der betrieblichen Abläufe zur Herstellung eines vorgegebenen Produkts. Dagegen sind zur Entwicklung grundsätzlich kostengünstiger Produktkonzepte Methoden einzusetzen, die zur Verringerung der Teilevielfalt und zur fertigungs-, montage- und materialkostengünstigen Gestaltung von Produkten dienen.

Die gesamten **Lebenslaufkosten** eines Produktes setzen sich aus den anfallenden fixen und variablen Einzel- und Gemeinkosten zusammen und hinter diesen vier Kostenarten verbergen sich jeweils anteilig die sog. Komplexitätskosten, welche aus der internen Vielfalt bzw. Komplexität resultieren (vgl. u. a. [206], Anhang A.8 und Anhang A.9 für Details). Die Lebenslaufkosten lassen sich auch in die Kosten für das Unternehmen (Selbstkosten) und die Kosten für Produktnutzer aufgliedern [152]. Durch Analysen (vgl. u. a. [83], [146]) ist belegt, dass Entwickler durch die Festlegung eines wesentlichen Anteils der Lebenslaufkosten in hohem Maße die Wirtschaftlichkeit der Produkte sowohl für das Unternehmen selbst als auch für den Nutzer beeinflusst [569]. So werden in der PE zwar nur rund 9% der Selbstkosten verursacht aber rund 70% der Selbstkosten festgelegt (vgl. Bild 108). Daher können diese festgelegten Kosten durch Entscheidungen in der PE noch beeinflusst werden (Entscheidungsrelevanz) [152]. Da fehlerhafte Entscheidungen in den frühen Phasen der PE nur mit sehr hohem Aufwand in späteren Phasen korrigiert werden können (vgl. Fehlerfrüherkennung i. S. v. „Rule of Ten“ [450] und Änderungszeitpunkte im

PEP [21], [22]) ist die hohe Kostenbeeinflussbarkeit während der Entwicklung für den Produkterfolg besonders wichtig [152]. Zudem verdeutlicht WARTZACK auch weitere, maßgebliche Einflussbereiche des Entwicklers (z. B. Herstellbarkeit, Sicherheit, Recyclingfähigkeit) und der Festlegung der entsprechenden Produkteigenschaften im PEP [588]. Daraus lässt sich folgern, dass Entwickler einen sehr hohen Einfluss und damit eine große Verantwortung für den Produkterfolg und damit für den Gewinn eines Unternehmens haben.

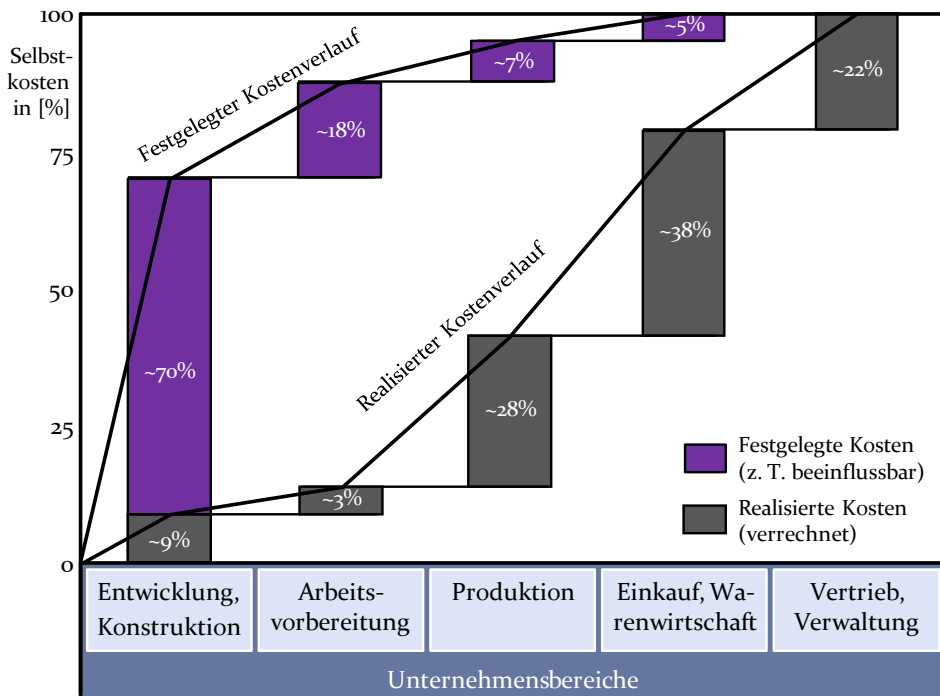


Bild 108: Kostenfestlegung und Kostenrealisierung in Unternehmensbereichen (in Anlehnung an [152], ähnlich zu [569], [352] und [151], vgl. auch [582])

Auswirkungen der steigenden Komplexität

Bei stagnierenden oder sogar sinkenden Absatzmengen in bekannten bzw. bereits bearbeiteten Märkten versuchen Unternehmen oft weitere Marktsegmente zu erschließen (s. [397], [497] für Details). Dabei ist die Stoßrichtung der Unternehmen je nach Ausgangsposition von Volumen- in Nischenmärkten oder umgekehrt. Mit der Penetration neuer Märkte verfolgen Unternehmen die Stabilisierung oder Erhöhung des Umsatzes sowie Auslastung freier Kapazitäten [497]. Dabei ist die Erwartungshaltung von Unternehmen, dass durch die zusätzlichen Produktvarianten (z. T. auch Sonderausstattungen, Exoten) hohe Deckungsbeiträge realisiert werden [497].

Dies ist aber nur möglich, falls der höhere Preis der zusätzlichen Varianten über den jeweils varianzinduzierten Kosten (s. Komplexitätskosten in Anhang A.16), die u. a. aufgrund der weiteren, differierenden Anforderungen entstehen, liegt [497]. Häufig werden jedoch einerseits die geplanten Volumeneffekte (engl. Economies of Scale) durch die Steigerung der Variantenanzahl und -vielfalt überschätzt und andererseits die sich auf der zunehmenden Komplexität ergebenden Effekte (v. a. Komplexitätskosten) unterschätzt (vgl. [286]). Daher ist das Finden der **kostenoptimalen Produktvielfalt** im Rahmen des Komplexitätsmanagement eine zentrale Herausforderung. Im Bild 109 wird der Zusammenhang zwischen den Kosten, die von der Menge und von der Komplexität abhängig sind, mit den Stückkosten und der Produktvielfalt in Bezug gesetzt, um die kostenoptimale Produktvielfalt zu bestimmen.

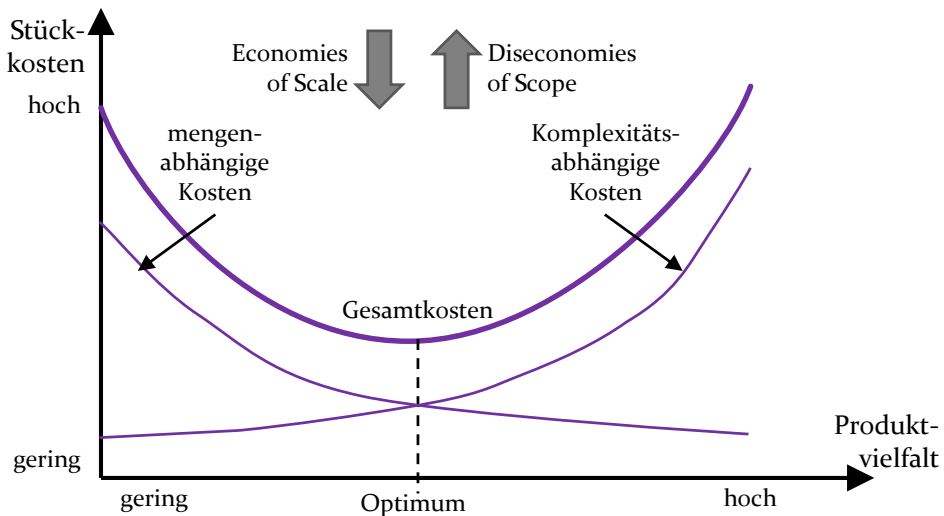


Bild 109: Kostenoptimale Produktvielfalt (in Anlehnung an [286])

Allerdings werden durch die zusätzlichen Produktvarianten zum einen oft nur geringe Absatzsteigerungen realisierbar (vgl. Verflachung der Häufigkeitsverteilung in Bild 110 sowie [381] für Details) und zum anderen häufig sehr hohe **Komplexitätskosten** verursacht, die zur Zunahme sowohl der Einzel- als auch der Gemeinkosten führen. Sofern diese Zusatzkosten (i. S. v. Komplexitätskosten) mittels Steigerungen der Preise an den Markt weitergegeben werden können, führt dies zu einer sinkenden Wettbewerbsfähigkeit [397]. Andernfalls wird durch die Komplexitätskosten entweder der Gewinn reduziert oder schlimmstenfalls Verlust realisiert.

Die Wechselwirkungen zwischen Produktvarianten, Komplexitätskosten und Preisen werden in Bild 110 schematisch dargestellt. Nach SCHUH haben

die meisten produzierende Unternehmen früher ihren Umsatz i. d. R. mit einem überschaubaren Produktportfolio (i. S. v. geringe Anzahl und Vielfalt von Produktvarianten bzw. „Standard“-Produkten) realisiert, wodurch hohe Skaleneffekte aufgrund standardisierter Produkte und Prozesse möglich sind [492]. Bei vielen Unternehmen ist das Produktportfolio über die Zeit bis zur Gegenwart bzw. heute sehr breit gefächelter, heterogen und exotenreich geworden (vgl. in Bild 110 breiter werdende Gauß-Verteilung), was v. a. durch Sättigungserscheinungen am Markt und in Folge steigenden Wettbewerbsdrucks oft weiter beschleunigt wird [492]. Dadurch steigen folglich auch die i. d. R. **nicht verursachungsgerecht erfassten Komplexitätskosten** (oft nur Cost-Plus-Ansatz), die den verschiedenen Varianten auch nicht entsprechend zugeordnet werden (können) [381], [492]. Somit sind die wirklichen Vollkosten und damit die Marge einer Variante nicht nachvollziehbar und eine ziel(kosten)gerichtete Optimierung des Portfolios nicht möglich [492]. Dies ist aus den folgenden beiden Gründen für Unternehmen von Nachteil [489], [492] (vgl. in Bild 110):

- Die Exoten erzielen zwar etwas höhere Preise als die Standardprodukte, verursachen jedoch deutlich mehr bzw. überproportional höhere Kosten. Beispiele für nicht einfach erfassbare und somit schwer verursachungsgerecht zuordenbare Komplexitätskosten sind bspw. geringere Lerneffekte, höherer Abstimmungsbedarf und erforderlicher Werkzeugkosten. Da die angenommene Kostenkurve (d. h. traditionell verrechnete Kosten) oft zu flach ist, macht das Unternehmen – i. d. R. trotz höhere Deckungsbeiträge – mit den exotischen Produktvarianten Verlust.
- Die Standardprodukte müssen oft zu einem höheren Preis verkauft werden, um die Verluste mit den Exoten zu kompensieren und dadurch insgesamt einen Unternehmensgewinn zu erreichen. Die Folge dieser (teilweisen) Quersubventionierung ist ein Wettbewerbsnachteil gegenüber Konkurrenten, die auf Standardprodukte bzw. ein möglichst modulares Produktportfolio fokussieren und daher zu signifikant geringeren Preisen ihre Produktvarianten gewinnbringend verkaufen können. Dadurch wird die Wettbewerbsfähigkeit im volumengrößten Markt- bzw. Kundensegment und letztlich auch insgesamt niedriger.

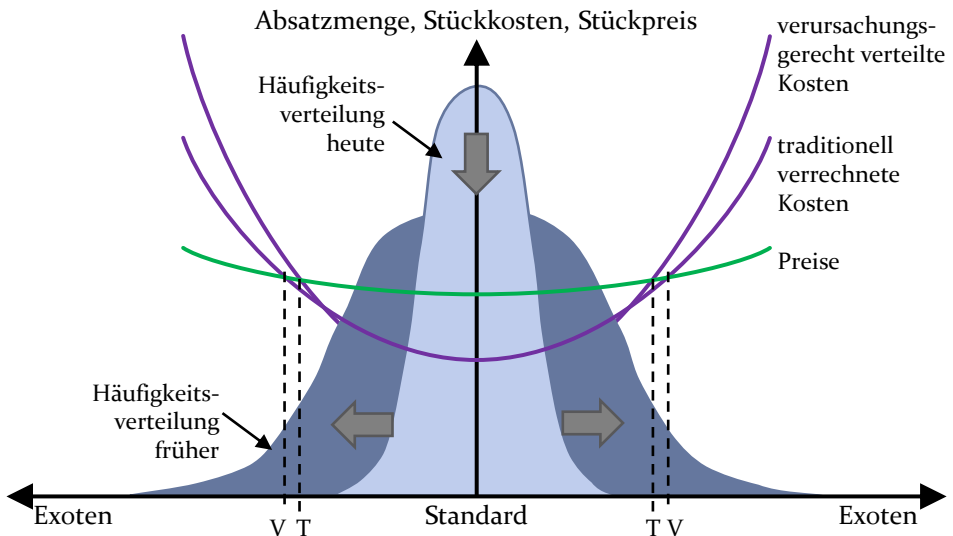


Bild 110: Wechselwirkungen zwischen Produktvarianten, Komplexitätskosten und Preisen (in Anlehnung an [489], [490] und z. T. ergänzt gemäß [381])

Eine wesentliche Folge, der schwierig erfassbaren und v. a. häufig fehlenden verursachungsgerechten Zuordnung von Komplexitätskosten zu Produktvarianten, ist die auf nicht systematische und zudem z. T. auch auf falsch interpretierte Kundenorientierung begründete Entwicklung von neuen Produktvarianten [497]. Außerdem führt die mangelnde Transparenz über Ursachen und Wirkungen sowie deren Beziehungen dazu, dass oft nur an der Reduzierung der Symptome von Komplexität gearbeitet wird und die **Ursachen der Komplexität** nicht erkannt sowie nicht reduziert, vermieden oder beherrscht werden [494].

Indikatoren für ein nicht ausreichendes Komplexitätsmanagement von Produkt- und Prozessvielfalt sind im sog. schwachstellenbasierten Checklistenverfahren von RATHNOW zur Ermittlung des Handlungsbedarfs definiert [447]. Demnach führt eine nicht ausreichende Entscheidungsgrundlage zu einem ausufernden Produktportfolio, das wiederum Probleme bei der Beherrschung der Vielfalt zur Folge hat (z. B. überproportional steigende Gemeinkosten) [447], [497]. Aus diesen Aspekten erfolgt gemäß dem Verfahren die Ableitung des Handlungsbedarfes, d. h. v. a. die Analyse der Ursachen der Komplexität sowie die daraus zu folgernden, zielgerichteten Maßnahmen [447] (vgl. [497] und in Bild 111).

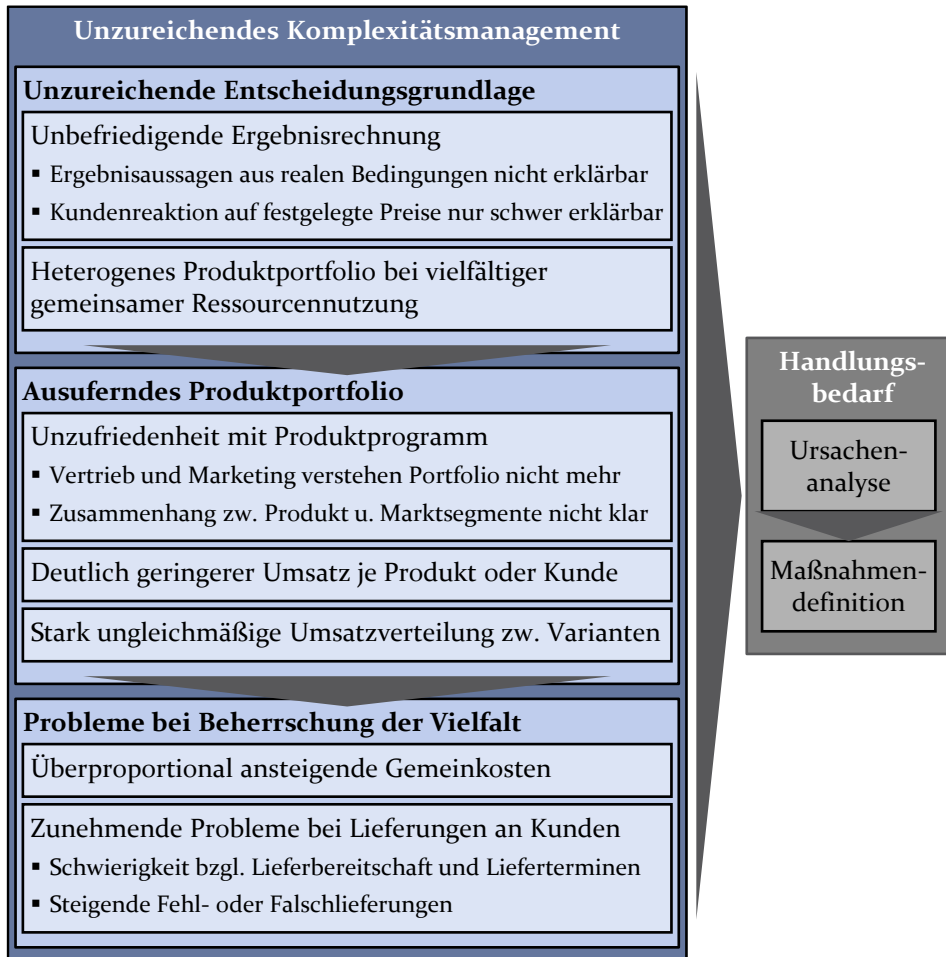


Bild 111: Ableitung des Handlungsbedarfs im Umgang mit der Komplexität (in Anlehnung an [447] und auch [497])

Auswirkungen der Komplexität in der Entwicklung

Die externen und internen Treiber sowie Ursachen der Komplexität haben viele Auswirkungen auf alle Funktionsbereiche eines Unternehmens. In diesem Abschnitt werden kurz die Auswirkungen der Komplexität in der Entwicklung in Anlehnung an WEISER [598] erläutert. Nach ihr lassen die **Komplexitätseffekte in den Lebenszyklusphase eines Produkts** unterscheiden und daher werden diese in [598] je nach Funktionsbereich getrennt voneinander erläutert. Somit sei für weitere Komplexitätseffekte in den Unternehmensbereichen Einkauf, Produktion (und Logistik), Vertrieb (und Marketing) auf die Zusammenstellung aus unterschiedlichen Literaturquellen auf [598] verwiesen.

In der PE fallen zwar kaum Kosten an, jedoch werden in diesem Funktionsbereich rund 70% der gesamten Produktkosten festgelegt (s. Bild 108). Daher ist es besonders in der PE wichtig auf das Komplexitätsmanagement, wie z. B. mit der geeigneten Kombination von Strategien (vgl. Kapitel 2.2.3), proaktiv für die nachfolgenden Unternehmensbereiche zu achten. Allerdings hat die zunehmende Komplexität breites viele, vielfältige Effekte auf die PE selbst. Dabei sind die wesentlichen **Auswirkungen der Komplexität in der PE** (v. a. i. S. v. neuen Varianten) gemäß einer Literaturstudie nach WEISER [598], die vom Autor dieser Arbeit ergänzt wurde:

- Zunehmender Entwicklungsaufwand, z. B. Konstruktion neuer Teile (z. T. für jeden neuen Auftrag), Simulationen und Versuche
- Erschwerte Priorisierung von Projekten, z. B. auch zwischen Neuentwicklungen und Konstruktionen für Aufträge
- Problematischerer Entwicklungskoordination und erhöhter Abstimmungsbedarf aufgrund geringerer Transparenz führt z. B. zu erhöhten Komplexitätskosten und Risiko von Parallelentwicklungen
- Problem der horizontalen und vertikalen Integration mit zahlreichen Neu- und Änderungskonstruktionen
- Zunahme der Varianten und Produktänderungen
- Schwierigeres Anforderungs- und Änderungsmanagement
- Zusätzliche technische Unterlagen und Produktdokumentationen
- Zunahme zusätzlicher neuer BoMs, Stücklisten und Stammdaten
- Aufwändigere Normung, Typung und Standardisierung
- Zusätzliche Musterteile, Prototypen, Arbeitspläne und Werkzeuge

A.10 Strategien und Ziele des Komplexitätsmanagements

In diesem Anhang werden ein **Überblick über die Strategien und über die Ziele des Komplexitätsmanagements** gegeben. So erfolgt zuerst eine ausführlichere Erläuterung der in Kapitel 2.2.3 skizzierten Strategien des Komplexitätsmanagements. Anschließend werden aufbauen auf den Strategien die Ziele des Komplexitätsmanagement erläutert.

Überblick über die Strategien des Komplexitätsmanagements

Ähnlich wie für die Definition von Komplexität (vgl. Kapitel 2.2.1) lassen sich auch für die **Begriffsbestimmung von Komplexitätsmanagement** eine Vielzahl verschiedener Erklärungen finden (vgl. [491]). An dieser Stelle wird auf wissenschaftliche Herleitung des Begriffs verzichtet (s. hierfür z. B. [491]) und stattdessen nachfolgend der Fokus auf die Strategien des Komplexitätsmanagements gelegt. In dieser Arbeit umfasst Komplexitätsmanagement – in Anlehnung an und Modifikation von [491] – die Planung, Entwicklung und Steuerung der unternehmensinternen Komplexität (bzgl. Anforderungen, Produkte, Prozesse, Wissen, Organisation und Kosten). Dabei wird durch das Komplexitätsmanagement in der PE das Ziel verfolgt, zum einen den Kundennutzen zu maximieren und zum anderen gleichzeitig den Unternehmenserfolg zu maximieren [491]. Dazu sind alle relevanten externen und v. a. internen Komplexitätsarten sowie alle Auswirkungen von Komplexität auf unterschiedlichen Unternehmensreife (s. u. a. Kapitel 1 und Anhang A.9) zu beachten. Hierfür sind die folgenden Aufgaben des Komplexitätsmanagements nach KRAUSE relevant: Erfassung und transparente Darstellung von Komplexität (d. h. Darstellung des Systemmodells), Bewertung und Bestimmung des optimalen Komplexitätsgrads (d. h. Festlegung der Zielsetzung) sowie Reduktion, Beherrschung und Vermeidung von Komplexität (d. h. Entwicklung von Strategien zur Systembeeinflussung) [322].

Neben der zunehmenden Individualisierung der Kundenbedürfnisse (Kundenkomplexität) nehmen im Hinblick auf die externe Komplexität auch weitere Komplexitätsarten zu (vgl. Anhang A.8). Dies führt i. d. R. zu einer wachsenden unternehmensinternen Komplexität und erschwert daher den **Umgang mit Komplexität**. Zur Handhabung von Komplexität lassen sich in der Literatur viele grundlegende Strategien finden (s. für einen Überblick u. a. [491], [497]), die auch als Basis- und Normstrategien bezeichnet werden. So wird nach WENZEL, der vom Axiom „komplexe Probleme erfordern komplexe Lösungen“ ausgeht und v. a. die Perspektive aus der

Entwicklungsorganisation einnimmt, zwischen den sechs grundlegenden Prinzipien zum Umgang mit Komplexität unterschieden [602]: Wachstum und Vernetzung, funktionale Differenzierung/ Dekomposition, Prozesse und Synchronisation, Autonomie und Selbststeuerung, Integration, Steuerung. Dabei betont er, dass diese Prinzipien in dieser Reihenfolge den Fortschritt „organisierter Komplexität“ in der PE beschreiben [602]. Dagegen unterscheiden SCHATZ ET. AL. im Rahmen ihres Konzepts zur Einführung der sog. Komplexitätsbewirtschaftung mit den drei Phasen Selektieren, Erkennen & Bewerten und Orchestrieren in der Phase des Orchestrierens zwischen den folgenden fünf grundlegenden Strategien des Komplexitätsmanagements: Umgang mit Komplexität, Reduzierung der Komplexität, Vermeidung von Komplexität, Bepreisung von Komplexität, Generierung von Komplexität [468]. Dabei betonen die drei Wissenschaftler, dass diese Strategien und deren jeweilige Kombinationen ein anforderungsgerechtes Komplexitätsmanagement erlauben [468]. Im Umgang mit Komplexität sind die vier grundsätzlichen, abstrakten Ziele bzw. Zielrichtungen des Komplexitätsmanagements – Dimensionieren, Differenzieren, Spezifizieren, Stabilisieren – jeweils bzgl. der Systemelemente und Systemrelationen möglich.

In dieser Arbeit wird zwischen den drei grundsätzlichen **Strategien der Komplexitätsreduktion, -beherrschung und -vermeidung** differenziert (s. Bild 13 in Kapitel 2.2.3). Diese drei Strategien des Komplexitätsmanagement sind auch in der Literatur sehr stark verbreitet (vgl. [3], [610], [480], [206]).

Die **Komplexitätsreduktion** hat die Zielsetzung, die bereits vorhandene Komplexität zu verringern. Komplexitätsreduzierende Maßnahmen auf systemstruktureller Ebene erfolgen grundsätzlich durch Reduktion der Vielzahl und der Vielfalt der Systemelemente und deren Relationen [480]. Zudem kann insb. durch Modularisierung eine Variantenreduktion erfolgen [480] und dadurch auch ganz grundsätzlich die internen Komplexitätsarten reduziert werden. Viele Ansätze verfolgen die Komplexitätsreduzierung, was jedoch in einigen Fällen mit einem negativen Kosten/Nutzen-Verhältnis verbunden sein kann. Oft ist eine bestimmte Komplexität auch erforderlich, um Kundenanforderungen ausreichend zu erfüllen [353]. Methodische Ansätze zur Komplexitätsreduktion sind u. a. (vgl. [322], [611]):

- Reduzierung der Anzahl an kombinierbaren Elementen, wie bspw. Produktelemente (z. B. Modulvarianten), Prozesselemente (z. B. Vorgänge) und Wissens-elemente (z. B. CAD-/CAM-Datenmodelle)

- Bereinigung des Portfolios von unattraktiven Produkten und nur Bereitstellung von oft nachgefragten und profitablen Produkten (z. B. mit der ABC-Analyse, vgl. für Details [441])
- Reduktion der vielfältigen, fall- und auch stufenweisen Kombinierbarkeit, d. h. Definition fester Elementkombinationen (z. B. sog. Paketangebote) mit Hilfe von z. B. automatisierten Constraints
- Klassifikation von Elementen und Ermöglichung von klassenweisen und produktlinienübergreifenden Kombinationen, wie z. B. der Zusammenfassung von Bauteilgruppen und Entwicklung im gleichen Team bzw. Produktion in der gleichen Fertigungsinsel
- Abgrenzung und gleiche Definition von funktionsübergreifenden Sachverhalten, wie z. B. gleiche Strukturierung und gemeinsame Pflege mehrgliedriger BoM-Strukturen in Vertrieb (order-BoM), Entwicklung (engineering-BoM), Produktion bzw. Einkauf (manufacturing-BoM) und Service (service-BoM)
- Reduktion der Vielfalt an Rohstoffen und Halbzeugen mit Hilfe von Simultaneous Engineering im Variantenentwicklungsprozess (vgl. für Details [610])

Die **Komplexitätsbeherrschung** hat das Ziel, die entweder „optimale“ und erforderliche oder nicht reduzierbare und vermeidbare Komplexität möglichst effektiv und effizient zu managen. Nach [611] und [480] ist das Hauptziel der Komplexitätsbeherrschung mit der nicht vermeidbaren internen Komplexität als Folge der externen Komplexität möglichst effizient umzugehen. Daher dienen Maßnahmen der Komplexitätsbeherrschung v. a. zur Effizienzverbesserung [480]. Darüber hinaus kann eine erfolgreiche Komplexitätsbeherrschung zu einem Wettbewerbsvorteil führen, weil allein anhand des Produkts keine Rückschlüsse auf das Know-how der effizienten Komplexitätsbeherrschung gezogen werden können und folglich die Imitierbarkeit durch Wettbewerber äußerst schwierig ist [615]. Aus diesen Gründen dient eine gewisse Komplexität auch als Schutz vor Produktplagiaten [615]. Grundsätzliche methodische Ansätze zur Komplexitätsbeherrschung sind u. a. (vgl. [322], [611]):

- Anpassung der bestehenden, organisationalen Strukturen, wie z. B. Anpassung der Auftragsabwicklung
- Verbesserung der Rahmenbedingungen entlang der ganzen Wertschöpfungsprozesskette, z. B. Gestaltung flexibler Schnittstellen
- Optimierung der IT-Systeme i. S. v. eines kontextspezifischen Optimums zwischen Standardisierung und individueller Anpassung, da zu wenig Standardisierung vermeidbaren Aufwand durch

Doppeltätigkeiten verursacht und zu viel Standardisierung suboptimale Kompromisse bzgl. IT-Werkzeugen erzwingt

- Schaffung eines guten Verständnisses bei allen Personen für die Probleme und Auswirkungen von Komplexität als Basis für einen bewussten Umgang mit komplexen technischen Systemen
- Bereitstellung von zweckdienlichen Werkzeugen mit geringem Bedienungsanfang und deren Integration in das Arbeitsumfeld.

Die **Komplexitätsvermeidung** hat die Zielsetzung, die Entstehung von Komplexität präventiv zu verhindern. Zur Komplexitätsvermeidung sind somit einerseits mögliche Komplexitätsbedarfe vor deren Entstehung im Hinblick auf die diversen Domänen zur Steigerung der Effektivität zu antizipieren [480] und andererseits mögliche Komplexitätstreiber bereits im Produktplanungs- und spätestens im Entwicklungsprozess zur Steigerung der Effizienz zu verhindern [611]. Demzufolge hat diese Strategie den längsten Umsetzungs- und Wirkhorizont [611]. Dabei steht eine fortwährende Komplexitätsvermeidung aus systemtheoretischer Perspektive im Widerspruch zu ASHBY's Postulat einer notwendigen Varietät, gemäß der die Ausprägung der externen mit der internen Systemkomplexität in enger Wechselwirkung steht [20]. Daher kann Komplexität nur bis zu einem gewissen Grad vermieden werden, um den Unternehmenserfolg zu realisieren [480]. Grundsätzliche methodische Ansätze zur Komplexitätsbeherrschung sind u. a. (vgl. [322], [611], [480]):

- Schaffung eines Informationsvorsprungs gegenüber dem Wettbewerb durch Foresight- und Szenarioanalysen und sog. Vorfeldmarketings
- Marktsegmentierung und Wertanalyse (Value Management) der Produkte und Prozesse bereits ab der Produktplanung
- Modularisierung von Produkt-, Prozess-, Dokumenten-, IT- und Organisationsstrukturen, z. B. Abstimmung und Trennung von Systemelementen mit hohen und niedrigen Veränderungsfrequenzen (vgl. u. a. [317] zu modularen Produktarchitekturen, [611] zu Plattformstrategien)

Vorbeugung durch präventives Variantenmanagement bzgl. eines zukünftigen Anpassungsbedarfs, d. h. Verwirklichung einer hohen externen Varianz bei einer möglichst geringen internen Varianz

Überblick über die Ziele des Komplexitätsmanagements

Dieser und die nachfolgenden sechs Abschnitte basieren stark v. a. auf dem Artikel [452] zum Komplexitätsmanagement von REIß, so dass auf ein Referenzieren des Artikels innerhalb dieser Abschnitte jeweils verzichtet wird. Die Intension dieser Abschnitte ist es einen **Überblick über Ziele des Komplexitätsmanagements** auf Grundlage der vier Komplexitätsdimensionen (vgl. Bild 10 in Kapitel 2.2.1) zu geben sowie diverse Ansätze bzw. Zielrichtungen zur Erreichung dieser Ziele zu skizzieren.

Demnach müssen Komplexitätsingenieure bzw. -manager (vgl. Kapitel 1.4) auf der Suche nach einem Komplexitätsoptimum zum einen die Reduktion und zum anderen die Erhöhung von Komplexität beherrschen sowie in einem bestmöglichen Verhältnis i. S. v. Kosten-Nutzen-Verhältnis setzen. Hierfür beschreibt REIß auf Grundlage der vier Dimensionen von Komplexität **vier grundsätzliche, abstrakte Ziele bzw. Zielrichtungen des Komplexitätsmanagements** (vgl. Bild 112), die nachfolgenden jeweils kurz zusammenfassend erläutert werden.

Ziele des Komplexitätsmanagements				
	Optimale Extension	Optimale Varianz	Optimale Varietät	Optimale Stabilität
Elementkomplexität	Erweitern ↓ Bereinigen	Diversifizieren ↓ Homogenisieren	Variieren ↓ Determinieren	Dynamisieren ↓ Verstetigen
	Dimensionieren	Differenzieren	Spezifizieren	Stabilisieren
Relationenkomplexität	Koppeln ↓ Entkoppeln	Konfrontieren ↓ Harmonisieren	Liberalisieren ↓ Disziplinieren	Selbst-Organisation ↓ Fremd-Organisation

Bild 112: Abstrakte Ziele des Komplexitätsmanagements (in Anlehnung an [452])

1. Zielrichtung: Dimensionieren

Hierbei wird im Rahmen des Komplexitätsmanagements eine optimale Auslegung eines Systems angestrebt. Im Hinblick auf die Systemelemente stehen für diesem Zweck Maßnahmen der klassischen Wachstums- bzw. Schrumpfpolitik zur Verfügung. Dabei ist v. a. Verhältnis zwischen der

Erweiterung bzw. Expansion auf der einen Seite und der Bereinigung bzw. Kontraktion auf der anderen Seite der Komplexität zu finden (s. Bild 112). Mit Bezug zu den Relationen geht es einerseits um die Entkopplung und andererseits um die Kopplung von Elementen mittels Relationen (s. Bild 112). Dabei ist mit **Entkoppeln** u. a. auch Emanzipation, Unterbinden eines cross subsidizing bzw. einer Kannibalisierung zwischen Produkten (vgl. v. a. [612]), Schaffung geschlossener und abgeschotteter Märkte gemeint. Hinsichtlich **Koppeln** kann zwischen lose Kopplung (z. B. Umwandlung von wechselseitigen in einseitige Lieferbeziehungen, sukzessive Planung nach dem kurzfristigen Ausgleichsgesetz) und verstärkte Kopplung der Elemente (z. B. Einkreis- statt Zweikreis-Abstimmung zwischen Finanz- und Betriebsbuchhaltung, engere Zusammenarbeit zwischen Ressorts, Übergang von Wertketten auf Wertkreisläufe durch Recycling) differenziert werden.

Im Gegensatz zur Zielrichtung **Erweiterung** wird mit der **Konzentration** eine Reduktion von Komplexität realisiert. Dabei kann die Konzentration entweder auf bestimmte Produkte (z. B. Stammgeschäft, Schlüsselprodukte) oder auf eine Fokussierung auf Elemente unter Vernachlässigung der Relationen erfolgen. So wird bspw. mit Hilfe von Modularisierung und Separierung ein komplexes System in eine Ansammlung von (möglichst) isolierten Clustern bzw. Inseln (vgl. Fertigungsinseln) umgewandelt.

Die Zielrichtung der **Bereinigung** ist sehr weit zu interpretieren. Hierbei ist die primitivste Form die Dezimierung (i. S. v. Kürzung, Einsparung), wie dies bspw. bei sog. „Rasenmäher“-Sparprogrammen des Gemeinkostenmanagements, den undifferenzierten Budgetstreichaktionen oder auch der Ausblendung von gewissen Variablen im Rahmen der Modellbildung (isolierende Abstraktion) der Fall ist. Dagegen erweist sich die fokussierte Selektion als effizienter. Dabei orientieren sich die Kürzungs-, Downsizing- bzw. Desinvestment-Aktivitäten gezielt an zuvor definierten Prioritäten.

Zur redimensionierenden Komplexitätsreduktion dient auch die **Klassenbildung** (z. B. Ordnen, Bilden von Teilefamilien durch Cluster-Analyse) und die **Komplexbildung** (z. B. Prozessbildung, Konfigurieren). Derartige Zusammenfassungen (z. B. Verdichtung, Aggregation, generalisierende Abstraktion) verringern die Anzahl der zu betrachtenden Elemente.

2. Zielrichtung: Differenzieren

Im Rahmen des Differenzierens wird ein optimales Maß an Verschiedenartigkeit innerhalb eines Systems angestrebt. Dabei soll die **Diversifizierung** die Ausprägungs- und Ideenvielfalt fördern, z. B. durch interdisziplinäre

Teams in der PE oder durch Variantenvielfalt. Im Gegensatz dazu soll durch die **Homogenisierung** eine (starke) Angleichung erfolgen, wie bspw. durch Aufbau einer homogenen Unternehmenskultur zur Angleichung von Grundwerten und Denkweisen.

An die Stelle vieler Einzelfallregelungen, die z. T. in **Konfrontation** zueinanderstehen können, treten im Zuge der Standardisierung (i. S. v. Programmierung, Normierung) einheitliche Werte (vgl. Norm- und Gleichteile, Sammelbewertungen, Pauschalwerte, wie z. B. km-Pauschale) und Regelungen (z. B. AGB, Entgelttarife, Kontenrahmen). Demnach sind Produkte (z. B. Qualitätsnormen, EDV-Standards), Ressourcen (z. B. einheitliche Berufsbilder) sowie die Prozesse der Transformation von Ressourcen in Produkte (z. B. Prozessspezifikationen) mögliche Ansatzpunkte einer Normierung. Der generell mit einer **Harmonisierung** verfolgte Zweck ist die abgestimmte Gleichgerichtetheit aller Elemente und Relationen.

3. Zielrichtung: Spezifizieren

Im Rahmen des Spezifizierens wird mittels spezifizierender Vorgaben eine adäquate Konkretisierung der Systemparameter angestrebt. So erfolgt mittels einer komplexitätsreduzierenden **Determinierung** die Reduktion vorhandener Freiheitsgrade, um infolgedessen Ermessens-, Interpretations- und Handlungsspielräume zu beschneiden, Grauzonen zu reduzieren und Transparenz zu realisieren. Das Gegenteil davon durch die Zielrichtung der **Variation** erreichen. Diese unterstützt oder bewahrt die Flexibilität, wie z. B. die Lernfähigkeit eines Systems.

Im Sinne eines **Liberalisierens** kann der möglichen Unwissenheit (z. B. aufgrund von Unschärfe) durch mehr oder weniger willkürliche Annahmen begegnet werden. Auf diese Art und Weise werden unbekannte Relationen in bekannte Relationen bzw. variable Elemente in definierte (i. S. v. konstante) Elemente überführt, was i. d. R. zu einer viel zu stark verkürzten Bewältigung von Komplexität führt. Die hohe Kunst einer vom jeweiligen Kontext abhängigen, bedarfsgerechten Spezifikation ist daher schwierig. Als Beispiel dient die flexible, oft mehrstufige und rollende Grob- und Feinplanung von z. B. modularen Produktstrukturen.

Mit Hilfe des **Disziplinierens** lassen sich Unschärfen reduzieren. Hierfür ist ein Ansatz die Dezentralisierung durch Verschieben von zentralen Funktionen, wie z. B. Einkauf, Produktion und Qualitätsmanagement, in die dezentralen, geschäftsführenden Unternehmensbereiche (Divisionen). Dadurch steigt i. d. R. aber aus Unternehmensgesamtsicht das notwendige Personal und die Kosten im Vergleich zu zentralen Funktionsbereichen an.

So geht mit der steigenden Anzahl selbstständiger, autonomer Organisationseinheiten eine Kostenreduktion aufgrund einer geringeren Schnittstellenkomplexität einher. Ferner wird durch eine sequenzielle Abarbeitung von Prozessen auch deren Berechenbarkeit und Automatisierbarkeit besser ermöglicht. Andererseits führen diese Einschränkungen der Vieldeutigkeit im Hinblick auf Freiheitsgrade und Unschärfe auch zu einer Verlängerung der Prozessdurchlaufzeiten (s. im Vergleich zu Simultaneous Engineering).

4. Zielrichtung: Stabilisieren

Im Hinblick auf gezieltes Stabilisieren wird als Zielrichtung entweder das **Dynamisieren** bzw. Auftauen oder das **Verstetigen** bzw. Einfrieren von Spezifikation des Systems verfolgt. Dabei ist mit Bezugnahme zu jeweils bestehenden Komplexitätsbedarfe (vgl. auch Kapitel 2.2.3) unter einer optimalen Stabilität keineswegs Unveränderlichkeit (i. S. v. „Preisstabilität“) sondern äußerst starke Stabilität gemeint. Die zwei Extremausprägungen, der Hyperstabilisierung bzw. Hyperlabilisierung, und entsprechende Fehlentwicklungen lassen sich durch ein angemessenes und für die jeweilige Situation stimmiges Maßnahmenbündel aus Dynamisierungs- und Verstetigungsaktivitäten vermeiden. Typische Dynamisierungsmaßnahmen sind Beschleunigung, Zeitkompression, Rüstzeitenverkürzung und kleine Serien. Als Verstetigungsaktivitäten eignen sich zeitliche Preisdifferenzierung, Mindesttaktzeiten, geringe Fluktuation. So sind einerseits fortwährend Maßnahmen gegen Formen der flexibilitätsbehindernden Habitualisierung (i. S. v. Macht der Gewohnheit), Verkrustung (i. S. v. abnehmender Reaktionsfähigkeit) und Zementierung (i. S. v. Wahrung beharrender Eigenarten) zu ergreifen. Andererseits sind durch geeignete Maßnahmen Orientierungslosigkeit (z. B. i. S. v. Aktionismus, Zick-Zack-Kurse) sowie Verluste durch eine zu starke Verkürzung der Produktlebenszyklen (i. S. einer „Beschleunigungsfalle“, s. [495]) zu verhindern. Eine optimale Veränderlichkeit wird oft durch permanente, jedoch kleine bzw. schrittweise Veränderung (i. S. v. engl. „continuous improvement“) realisiert.

Für ein optimales Stabilisieren ist analog zum Austarieren zwischen den Dynamisierungs- und Verstetigungsbestrebungen eine bestmögliche Kombination von Selbst- und Fremdorganisation zu finden. Hierfür sind die Parameter eines Systems bzgl. der **Selbst- und Fremdorganisation** geeignet zu wählen (vgl. „geplante Evolution“, „spontane Ordnung“, „kontrolliertes Chaos“). Dabei solle sich der Komplexitätsingenieur bzw. -manager (vgl. Kapitel 1.4) i. d. R. auf das Beeinflussen der Leitplanken bzw. des Rahmens beschränken und alles Weitere der Selbstorganisation des Systems überlassen. Diese prinzipielle Vorgehensweise zur Definition weitgehend

autonomer Systeme bzw. Subsysteme liegt z. B. bei der Schaffung von Holdingsstrukturen, Profit-Centern, Fertigungsinseln, teilautonomen Teams (z. B. mit der Scrum-Philosophie) zugrunde. Dabei wird das Innenleben dieser autonomen Systeme nur geringfügig oder nicht beeinflusst, so dass innerhalb dieser Systemgrenzen eine Etablierung evolutionär gewachsener Strukturen stattfindet. Sofern ein Intervenieren in die Selbstorganisation dieser Teilsysteme stattfindet, um auch innerhalb dieser die Veränderlichkeit zu reduzieren, kommt es oft nicht zu der intendierten Stabilisierung des zuvor selbstorganisierten Teilsystems, sondern stattdessen zu einer Destabilisierung des Gesamtsystems. Dies liegt v. a. an den Widerstandskräften in den Teilsystemen, die häufig zu einer Erhöhung des „Chaos“ führen. Ein derartiges, sog. „bürokratisches Paradoxon“ hat zur Folge, dass sich die jeweiligen Wirkungsrichtungen der beiden Pfeile in Bild 112 umdrehen.

Kombination der vier Zielrichtungen

Die vier zuvor beschriebenen, grundsätzlichen Zielrichtungen im Rahmen des Komplexitätsmanagements sind in ihrer Kombination, die für ein bestimmtes komplexes System für die jeweilige Situation bzw. Zielsetzung zweckmäßig wechselseitig und holistisch angepasst ist, zu verwenden (i. S. v. simultane Komplexitätsoptimierung). Dadurch können bedarfsgerecht sowohl synergetische Verstärkungseffekte als auch Kompensationen realisieren werden. Als Beispiel dient die Erzeugung einer marktgerechten Produktvielfalt bei einem Automobilhersteller, die als Optimum zwischen den beiden Extremen einer kundenindividuellen, maßgeschneiderten Sonderanfertigung und einer vollständig standardisierten Einheitsausführung (vgl. Fords T-Modell) zu finden ist. Hierfür bietet sich als optimale Lösung und damit als ein Kompromiss eine aus Marktanalysen abgeleitete, zielgruppenspezifische und kombinatorische Nutzung der Produktvariantenvielfalt (z. B. Ausstattungsvarianten) an. Dabei ist die Größe des Variantenspektrums und somit des Variantenoptimums abhängig von der Teilevielfalt und damit von der modularen Produktstruktur.

Schlussfolgerungen für das Komplexitätsmanagements

Die zuvor beschriebenen, prinzipiellen Ziele des bzw. Umgangsmöglichkeiten mit Komplexität untermauern die sehr große Bedeutung des Managements von Komplexität für Unternehmen. Damit ist – ähnlich wie die Unternehmenskultur – **Komplexität ein (formal-abstrakter) kritischer Erfolgsfaktor**. Da das Komplexitätsmanagement sowohl Abstimmungs- als auch Anpassungsphänomene umfasst und diese häufig holistisch versucht zu optimieren, kann das komplexitätsbewusste Denken als zentraler Schlüssel

zur gleichzeitigen Optimierung von Flexibilitäts- und Integrationspotentialen einer Organisation – insb. in der PE – angesehen werden.

Dadurch wird die große Bedeutung für den Aufbau und die kontinuierliche Weiterentwicklung von **Komplexitätskompetenz** deutlich (vgl. Kapitel 1.4). Dies erfordert entsprechende Trainings, und zwar sowohl für Komplexitätsingenieure und -manager als auch für alle anderen Mitarbeiter eines Unternehmens. Das Managen von und der Umgang mit Komplexität muss nicht nur gewollt, sondern auch richtig gut gekonnt sein. Eine fehlende Komplexitätskompetenz und/oder ein alleiniges subjektives „Gespür“ bzw. „Gefühl“ für Komplexität ist gefährlich, da ein unprofessioneller Umgang mit Komplexität zur leichtfertigen Vernichtung wertvoller Komplexitätspotentiale, zur Aufgabe vor scheinbar übermächtiger Komplexität oder zur Erzeugung von vermeidbarer und nicht nutzenbringender Komplexität führen kann.

A.11 Klassifikationen und Arten von Modellen

In der PE gibt es eine große Vielzahl verschiedener Modellarten. Dies liegt v. a. daran, dass es vielfältige abzubildende Systeme (i. S. v. Originale), viele Subjekte (v. a. Modellnutzer und -ersteller), zahlreiche Zeitspannen (z. B. zur Modellerstellung und -nutzung) sowie vielfältige Modellzwecke gibt. Zudem lässt sich die Modellvielfalt bspw. mit den disziplinspezifischen Konkretisierungen des jeweiligen Modellzwecks zur Abstraktion eines Originals begründen [31]. Außerdem wird die Eignung eines Modells für einen gewissen Verwendungszweck durch die jeweiligen Besonderheiten der zu lösenden Problem- oder Aufgabenstellung essenziell beeinflusst [60]. Demzufolge kann es kein universales Modell für jede beliebige Problemstellung eines komplexen Systems geben, sondern stattdessen vielfältige und jeweils für die Problemstellung geeignete und auf diese angepassten Modelle (vgl. [313]). Daher wird in diesem Anhang mittels einer kurzen Vorstellung von Modellklassifikationen und -arten ein Überblick über die gegenwärtige und zunehmende Modellvielfalt gegeben, um dadurch **das für diese Arbeit notwendige Modellverständnis** für Gemeinsamkeiten und Unterschiede verschiedener Modelle in der Entwicklung zu schärfen (für Details s. u. a. [313], [280]).

Ähnlich wie bei der Klassifikation technischer Systeme (vgl. Anhang A.6) gibt es auch eine **Vielzahl verschiedener Ansätze bzgl. der Klassifikation von Modellen**, und zwar sowohl bzgl. der Klassifikation ohne Bezug auf eine konkrete Fachdisziplin als auch für fachspezifische Klassifikationen von Modellen, wie bspw. den Betriebs- oder Ingenieurwissenschaften (vgl. u. a. [280]). Da die zahlreichen Klassifikationsansätze u. a. je nach Zweck, Fachdisziplin, Abstraktionsniveau deutlich voneinander abweichen, schlussfolgern JOCKISCH & ROSENDAHL, dass kein Klassifikationsansatz für alle Bereiche disziplinenübergreifend geeignet ist. Nachfolgend werden für einen groben Überblick über verschiedene Modellarten zuerst zwei disziplinneutrale Klassifikationen, die keinen direkten Bezug zu einem Fachbereich hat, skizziert. Diese und weitere bereichsneutrale Klassifikationen (vgl. [280], [321], [313]) stammen i. d. R. aus der Wissenschaftstheorie, die sich als eine Art Metadisziplin versteht und sich mit der Verwendung von Modellen in unterschiedlichen Wissenschaftsdisziplinen beschäftigt [280]. Danach liegt der Fokus auf ingenieurwissenschaftliche Klassifikationen, die vor allem im Hinblick auf relevante Modelle in der PE beleuchtet werden.

Disziplinneutrale Klassifikation nach STACHOWIAK

In der **disziplinneutralen Klassifikation** nach STACHOWIAK werden Modelle in vier verschiedene Klassen eingegliedert: Graphische, Technische, Semantische und Semantisch-szientifische Modelle [526] (s. Tabelle 14). Die Klassifizierungskriterien für die einzelnen Modellklassen werden von STACHOWIAK zwar nicht explizit erläutert, aber zwei zentrale Dimensionen sind grob erkennbar: Einerseits dient der Zweck bzw. Herkunftsbereich der Modelle und andererseits die Art und Weise der formalen Darstellung (z. B. natürliche Sprache, graphische Elemente, mathematische Formeln) als Differenzierungskriterium [280]. Dabei weist STACHOWIAK selbst darauf hin, dass seine Klassifikation weder das Eindeutigkeits- noch das Ausschließlichkeitskriterium erfüllt und somit Überschneidungs- und Kombinationsmöglichkeiten zulässt [526].

Tabelle 14: Disziplinneutrale Klassifikation von Modellen (nach [526] und [280])

Bezeichnung	Beschreibung
Graphische Modelle	Graphische Modelle sind zweidimensionale veranschaulichende Repräsentationen von Originalen. Je nachdem wie (un-)mittelbar Modelle das Original abstrahieren, werden innerhalb dieser Klasse zwischen Bildmodelle und Darstellungsmodelle differenziert: - Bildmodelle sind unmittelbare Abbildungen, wie Bilder, sowie teil- und vollschematische Abbildungen. - Darstellungsmodelle sind Graphen oder Diagramme.
Technische Modelle	Technische Modelle sind meist dreidimensionale, raum-zeitliche, materiell-energetische Repräsentationen von Originalen. Diese umfassen physikalisch, technische Modelle und bio-, psycho- und soziotechnische Modelle.
Semantische Modelle	Semantische Modelle sind sog. „Zeichenmodelle“. Diese sind nur dann (verständlich) kommunizierbar, falls Vereinbarungen darüber getroffen wurden, wofür die verwendeten Zeichen und Symbole (vgl. Notation) stehen. Diese Modelle erfassen die Realität(-ausschnitte) in der besonderen Weise der Sprache im weitesten Sinne.
Semantisch-szientifische Modelle	Innerhalb der semantisch-szientifischen Modelle wird zwischen den formal-wissenschaftlichen (Mathematik, Logik), empirisch-theoretischen (Naturwissenschaften, Sozialwissenschaften), technologischen (Ingenieurwissenschaftlichen) und praxeologischen Modellen (v. a. Erkenntnistheorie) differenziert.

Disziplinneutrale Klassifikationen nach TRIER ET AL.

Diesem Abschnitt liegt das Kapitel „Systemtheorie und Modell“ von TRIER ET AL. [559] zugrunde, so dass nachfolgend auf ein Referenzieren der Autoren verzichtet wird. Ziel des Abschnitts ist durch eine weitere, disziplinneutrale Klassifikation von Modellen eine noch bessere **Schärfung des Verständnisses für die verschiedenen Modellarten** zu erreichen. In der folgenden Übersicht, die aus den drei Bildern Bild 113, Bild 114 und Bild 115 besteht, wird die Klassifikation von Modellen nach jeweils einem der drei Klassifizierungskriterien Verwendungszweck, allgemeinen Eigenschaften und Darstellungsart vorgenommen. Dabei wurden die Erläuterungen und Beispiele v. a. aus Sicht der PE ergänzt.

Modellklassen	Erläuterungen
Beschreibungsmodelle	Beschreibungsmodelle haben einerseits einen auf das Original bezogenen rein dokumentarischen Charakter und andererseits bieten diese häufig eine Unterstützungsfunktion bei der Systemabgrenzung an. Sie werden daher auch als Erfassungs- oder Ermittlungsmodelle bezeichnet. Beschreibungsmodelle dienen aber auch der Kommunikation, besonders dann, wenn das beschriebene Original ideellen Charakter hat und somit schwer fassbar ist. Als Beispiele seien Organigramme zur Beschreibung von Organisationsstrukturen und Prozessmodelle zur Darstellung des PEPs genannt.
Simulationsmodelle	Simulations-/Prognosemodelle dienen der Vorhersage zukünftiger Zustände eines Originals, bei Variierung verschiedener unabhängiger Variablen. Sie finden insb. dann Verwendung, wenn Manipulationen am Original unmöglich oder mit hohem Aufwand oder Risiko verbunden sind. Typische Beispiele sind computerbasierte Modelle zur Simulation von neuen technischen Systemen oder solche, die der Prognose von Klimaveränderungen auf der Erde dienen. In der Systemanalyse spielt die Simulation mit Modellen eine wichtige Rolle, um Aussagen über den untersuchten Ausschnitt des Unternehmens und dessen Verhaltensweisen ableiten zu können.
Erklärungsmodelle	Erklärungsmodelle werden dann eingesetzt, wenn nicht alle relevanten Eigenschaften und deren Ausprägungen eines Originals im Rahmen einer Systemerhebung ermittelbar sind. Sie enthalten Hypothesen, die sowohl der Erklärung der Zustände eines Originals in der Vergangenheit und Gegenwart als auch der Prognose seiner zukünftigen Zustände dienen.

Bild 113: Disziplinneutrale Klassifikationen nach Verwendungszweck (in Anlehnung an [559])

Modell- klassen	Erläuterungen
Reale/ Ideelle Modelle	Hinsichtlich der Existenz ihrer Bestandteile wird zwischen realen und ideellen Modelle differenziert. Reale Modelle bestehen aus beobachtbaren materiellen Elementen der objektiven Realität (z. B. von technischen Systemen, wie Rohre). Ideelle Modelle bestehen aus Elementen, die das Ergebnis menschlicher, kognitiver Leistungen sind und nur innerhalb der subjektiven Realität existieren. Sie sind immateriell und bestehen aus Zeichen (z. B. Gedankenmodelle).
Natürliche/ künstliche Modelle	Nach ihrer Art der Entstehung werden natürliche und künstliche Modelle voneinander unterschieden. Natürliche Modelle sind ohne den Einfluss des Menschen entstanden. Künstliche Modelle sind durch den Einfluss des Menschen entstanden (z. B. Entwicklungsmodelle, wie Produktmodelle).
Geschlos- sene/ offene Modelle	Geschlossene Modelle haben keine Beziehungen zwischen Modell und Umwelt und sind somit vollständig von deren Umwelt isoliert ist (s. geschlossene Systeme, wie System natürlich. Zahlen). Offene Modelle haben Beziehungen zur Umwelt (z. B. Entwicklungsmodelle, wie Produkt- zu Prozessmodellen).
Statische/ dyna- mische Modelle	Bezüglich der Zeitabhängigkeit findet eine Unterscheidung zwischen statischen und dynamischen Modellen statt. Statische Modelle verändern die Elemente und Relationen selbst sowie deren Ausprägungen in Abhängigkeit von der Zeit nicht (z. B. viele grundlegende, mechanische Modelle). Dynamische Modelle verändern die Elemente und Relationen selbst sowie deren Ausprägungen in Abhängigkeit von der Zeit. Die Struktur und das Verhalten des Modells sind also flexibel (z. B. Produkt- und Simulationsmodelle).
Determi- nistische und stochas- tische Modelle	Hinsichtlich der der Bestimmbarkeit der Ausprägungen von Eigenschaften wird zwischen deterministischen und stochastischen Modellen unterschieden. Bei deterministischen Modellen können die Ausprägungen ihrer Eigenschaften mit Sicherheit angegeben werden (z. B. mechanische Modelle). Bei v. a. einfachen, technischen Systemen (z. B. Maschinen) kann über einen kurzen Betrachtungszeitraum eine verhältnismäßig genaue Vorhersage bezüglich des gegenseitigen Einwirkens der Elemente, die das Verhalten des Modells bestimmen, getroffen werden. Bei stochastischen Modellen liegen nur Informationen bzgl. der relativen Häufigkeit der möglichen Ausprägungen vor (z. B. Ausfallhäufigkeit einer Maschine bei langfristiger Betrachtung).

Bild 114: Disziplinneutrale Klassifikationen nach allgemeinen Eigenschaften (in Anlehnung an [559])

Modell- klassen	Erläuterungen
Ikonische Modelle	Ikonische Modelle sind bildliche Modelle, die die relevanten Eigenschaften des Originals durch bildhafte Eigenschaften repräsentieren. So werden bspw. bei grafischen Benutzeroberflächen von Computern Dateiverzeichnisse oft durch Ordner dargestellt. Kennzeichnend für ikonische Modelle sind ihre Sinnbildlichkeit, wodurch sie ihre Bedeutung unmittelbar selbst repräsentieren (z.B. Icons im Strukturbaum gängiger 3D-CAD-Systeme).
Analoge Modelle	Analogen Modellen stellen die abgebildeten Eigenschaften des Originals durch andere Eigenschaften des Modells dar. Es findet also eine Kodierung der Eigenschaften des Originals statt. So ist das Morse-Alphabet ein analoges Modell des Alphabets, da es durch die Abbildung der Buchstaben des Alphabets auf Kombinationen von Punkten und Strichen entstanden ist. Ebenso kann ein mechanischer Schwingkreis durch einen elektrischen dargestellt werden. Aufgrund der Kodierung benötigen analoge Modelle einer Übersetzungstabelle, welche eine Zuordnung von Original- und Modellattributen ermöglicht (z. B. eine Codetabelle).
Symbo- lische Modelle	Symbolische Modelle repräsentieren die Eigenschaften des Originals durch Symbole. Typische Beispiele sind mathematische Formeln zur Darstellung von Systemen. So kann in einem betriebswirtschaftlichen Modell der lineare Zusammenhang zwischen dem Preis eines Gutes und der Nachfrage durch die Formel $p=1,5x$ dargestellt werden, wobei p für den Preis und x für die Menge des Gutes steht.
Sprachliche Modelle	Sprachlichen Modellen repräsentieren die relevanten Eigenschaften eines Originals durch sprachliche Konstrukte. Während verbale Modelle auf einer natürlichen Sprache basieren, werden formale Modelle in einer im Vorfeld festgelegten (formalen) Sprache dargestellt. Die schriftliche Darstellung historischer Zusammenhänge in einem Lehrbuch ist ein Beispiel für ein verbales Modell, wogegen der Quellcode eines Computerprogramms als ein formales Modell angesehen werden kann.
Gedank- liche Modelle	Gedankliche Modelle, auch mentale Modelle genannt, repräsentieren die Eigenschaften von Originalen durch imaginäre bildliche, räumliche oder sprachliche Konstrukte. Ist bspw. ein Mensch aufgefordert, allgemeine Eigenschaften eines Rohr zu beschreiben, dann verwendet er dafür sein mentales Modell eines Rohrs. Aber auch bei der zwischenmenschlichen Kommunikation spielen mentale Modelle eine wesentliche Rolle. Wird in einem Gespräch eine Aussage wie „Modelle sind Systeme“ verwendet, dann ist es für eine sinnreiche Kommunikation erforderlich, dass die Gesprächspartner eine – zumindest in groben Zügen – gleiche Vorstellung, also ein ähnliches mentales Modell, von den in der Aussage enthaltenen Begriffen haben.
Gegen- ständliche Modelle	Gegenständliche Modelle sind dreidimensionale Objekte der materiellen Welt. In ihnen werden die Eigenschaften eines Originals auf eine greifbare Weise repräsentiert. Typische Beispiele dafür sind Modellbauten von Gebäuden und Prototypen von Maschinen. Die Unterteilung bezüglich des Zwecks erlaubt es Modelle in Beschreibungs-, Simulations- und Erklärungsmodelle einzuteilen. Im Blickpunkt des Betrachters liegt hier demnach ein Teilaspekt des pragmatischen Merkmals.

Bild 115: Disziplinneutrale Klassifikationen nach Darstellungsart (in Anlehnung an [559])

Schlussfolgerungen aus den beiden disziplinneutrale Klassifikation

Die in den Ingenieurwissenschaften sowie insb. in der PE weit verbreiteten Modelle finden sich prinzipiell sowohl in allen der vier in Tabelle 14 skizzierten Klassen nach nach STACHOWIAK (s. [526], [280] als auch in allen Modellklassen, die jeweils in den drei Bildern (d. h. Bild 113, Bild 114 und Bild 115) nach TRIER ET AL. (s. [559]) beschrieben sind, wieder. Da **disziplin-neutrale Klassifikationen allgemein gehalten** werden (z. T. i. S. v. auf einem hohen Abstraktionsniveau), können diese i. S. eines disziplinenübergreifenden Rahmens für die Verortung und für Vergleiche von disziplinspezifischen Modellen verwendet werden [313].

Disziplin- bzw. domänenspezifische Modellarten

Im Gegensatz zu disziplinneutralen Klassifikationen umfassen fach- bzw. disziplinspezifische Klassifikationen lediglich eine begrenzte Anzahl an spezifischen Modellen der jeweiligen Disziplin und sind auch nur innerhalb der entsprechenden Disziplin gültig (vgl. [313]). Aus den unterschiedlichen Zielsetzungen, Rahmenbedingungen und Besonderheiten der verschiedenen Fachdisziplinen innerhalb in der Ingenieurwissenschaften ergeben sich auch eine Vielzahl an möglichen Klassifizierungskriterien von Modellen und somit auch mannigfaltige Modellklassifikationen (vgl. u. a. [280], [313]), die z. T. im Widerspruch zueinanderstehen und deren (Modell-)Klassen sich häufig untereinander z. T. überschneiden [280], [313]. Zudem ist insb. die Entwicklung durch die funktions- und disziplinenübergreifende Zusammenarbeit geprägt, so dass auch demzufolge eine **Modellklassifikation für die Entwicklung nur begrenzt sinnvoll** erscheint. So werden zwar in der PE v. a. disziplinspezifische Modelle für die Modellierung einzelner Domänen eingesetzt, aber oft nur in der jeweiligen Fachdisziplin ausreichend gut verstanden. So wird – verkürzt ausgedrückt – bspw. unter einem Produktmodell je nach Zeitpunkt im PEP ein Funktions-, Prinzip- oder ein Gestaltmodell (vgl. Bild 119) und je nach Fachdisziplin z. B. ein CAD-Modell oder Simulationsmodell (vgl. Bild 16 in Kapitel 3.1.1) verstanden. In dieser Arbeit werden auf Grundlage der sechs Komplexitätsarten (vgl. auch die beiden Kapitel 1.1 und 1.4) auch die entsprechenden Modellarten voneinander differenziert: Anforderungs-, Produkt-, Prozess-, Wissens-, Organisations- und Kostenmodellen.

Die im **Fokus der Arbeit stehenden Modellarten** sind typische, in der PE verwendete Modelle. Daher werden diese nur kurz wie folgt skizziert:

- **Anforderungsmodelle** (z. B. Lasten- und Pflichtenhefte) bzw. Spezifikationen sind der Ausgangspunkt jeder Entwicklung,

- **Produktmodelle** (z. B. Funktions- und CAD-Modelle) sind diverse immaterielle und/oder materielle Ergebnisse jeder Entwicklung,
- **Prozessmodellen** (z. B. Vorgehensmodelle, Mikrozyklen) dienen der Strukturierung der Entwicklung in Phasen und Teilschritte,
- **Wissens- und Informationsmodelle** (z. B. Ontologien, Wissensgraphen) unterstützen zum einen die Entwicklung von Produkten und zum anderen werden die zunächst durch Daten/Informationen beschriebene Produkt materiell realisiert (vgl. [151]),
- **Organisationsmodelle** (z. B. traditionelle vs. agile Entwicklung) dienen der Organisation der interdisziplinären Entwicklungsarbeit,
- **Kostenmodelle** (z. B. für direkte Kosten, Komplexitätskosten) sind in der Entwicklung einerseits zu berücksichtigen und andererseits auch weitgehend festzulegen (vgl. u. a. Bild 108 und Anhang A.8)

In Anlehnung den dieser Arbeit zugrunde liegenden sechs Domänen bzw. Arten der Komplexität sind in Tabelle 15 auf **verschiedene Klassifikationen und/oder Beschreibungen** von Anforderungs-, Produkt-, Prozess-, Wissens-, Organisations- und Kostenmodellen in der einschlägigen Literatur der PE verwiesen. Diese tabellarische Übersicht erhebt keinerlei Anspruch auf Vollständigkeit und dient als Ausgangspunkt für eine vertiefende Auseinandersetzung mit den vielfältigen Modellarten in der PE.

Tabelle 15: Überblick verschiedener Modellklassifikationen und -arten in der Entwicklung

Modellart	Autor(en)	Literatur
Anforderungsmodelle	Breiting, A. & Knosala, R. (v. a. aus Sicht anforderungsorientierter Bewertung für techn. Systeme)	[77]
	Hubka, V. (v. a. aus Sicht anforderungsorientierter Entwicklung von technischen Systemen)	[268]
	Nagarajah, A. (v. a. aus Sicht der methodischen Klärung der Aufgabenstellung, insb. in der Entwicklung)	[398]
	Hruschka, P. (v. a. aus Sicht des Requirements Engineerings zur Produkt- und Prozessverbesserung)	[265]
	Kohl, M. (v. a. aus Sicht der Anforderungsableitung aus dem Produktlebenszyklus, insb. für den PEP)	[S31]

Modellart	Autor(en)	Literatur
Produktmodelle	Buur, J. & Andreasen, M. (v. a. aus Sicht der Produktmodellierung, insb. für die PE)	[102]
	Kohn, A.; Lutter-Günther, M.; Hagg, M. & Maurer, M. (v. a. aus Sicht der Produktmodellverbesserung, insb. im Umgang mit Information in der Entwicklung)	[314]
	Eisenbart, B.; Blessing, L. & Gericke, K. (v. a. aus Sicht der Funktionsmodellierung)	[158]
	Erden, M.; Komoto, H.; van Beek, T.; D'Amelio, V.; Echavarria, E. & Tomiyama, T. (v. a. aus Sicht der Funktionsmodellierung)	[165]
	Birta, L. G. & Arbez, G. (v. a. aus Sicht von Simulationsmodellen, insb. für dynamisches Systemverhalten)	[60]
Prozessmodelle	Smith, R. & Morrow, J. (v. a. aus Sicht der Prozessmodellierung, insb. in der PE)	[513]
	Gericke, K. & Blessing, L. (v. a. aus Sicht der Prozessmodelle, insb. in der PE)	[210]
	Krehmer, H. (v. a. aus Sicht der Prozessmodelle, insb. in der PE)	[323]
	von Rosing, M.; Kemp, N.; Hove, M. & Ross, J. W. (v. a. aus Sicht der Prozessstrukturierung)	[459]
	Kobler, M. (v. a. aus Sicht der Qualitätssicherung bei der Prozessmodellierung)	[305]
Wissens- und Informationsmodelle	Roth, D. (v. a. aus Sicht der Wissensanalyse und -bewertung, insb. in der PE)	[463]
	Lehner, F. (v. a. aus Sicht des Wissensmanagements, insb. Methoden und Softwaresysteme)	[345]
	Probst, G.; Raub, S. & Romhardt, K. (v. a. aus Sicht des Wissensmanagements, insb. Wissensprozesse)	[439]
	North, K. (v. a. aus Sicht des Wissensmanagements, insb. Wissensorganisation)	[409]
	Brachman, R. & Levesque, H. (v. a. aus Sicht der Wissensrepräsentation)	[74]

Modellart	Autor(en)	Literatur
Organisationsmodelle	Ebers, M. & Arbeitskreis Organisation der Schmalenbachgesellschaft für Betriebswirtschaft e. V. (v. a. aus Sicht der BWL, insb. Organisationsmodelle)	[142]
	Pepels, W. (v. a. aus Sicht der BWL, insb. Organisationsgestaltung)	[428]
	Schreyögg, G. & Geiger, D. (v. a. aus Sicht der BWL, insb. Organisationsgestaltung)	[486]
	Vahs, D. (v. a. aus Sicht der BWL, insb. Organisationsmanagement)	[567]
	Bullinger, H.-J.; Warnecke, H. J. & Westkämper, E. (v. a. aus Sicht der BWL, insb. Organisationsformen)	[95]
	Bleicher, K. (v. a. aus Sicht der Organisationstheorie)	[65]
Kostenmodelle	Ehrlenspiel, K.; Kiewert, A.; Lindemann, U. & Mörtl, M. (v. a. aus Sicht der PE)	[152]
	Mörtl, M. (v. a. aus Sicht der PE)	[394]
	Mörtl, M. & Schmied, C. (v. a. aus Sicht der PE, insb. Design for Cost)	[395]
	Blanchard, B. & Blyler, J. (v. a. aus Sicht der Prozesskosten und des Systems Engineerings)	[63]
	Friedl, G.; Hofmann, C. & Pedell, B. (v. a. aus Sicht der BWL, insb. Kostenrechnung)	[194]

A.12 Grundlagen zum Modellverständnis und zur Modellierung

Terminologische Grundlagen zu Domäne, Abstraktion und Modell

Mit Hilfe der Sprache findet sowohl die Kommunikation zwischen IT-Entwicklern als auch die Implementierung von IT-Systemen statt [292]. Die Verwendung einer geeigneter (Fach-)Sprache ist für Modellierungsmethoden und -sprachen von herausragender Bedeutung [185]. Infolgedessen ist es für diese Arbeit wichtig, wesentlich Fachbegriffe zu definieren und zueinander in Beziehung zu setzen. Daher werden in diesem Abschnitt einige **terminologische Grundlagen** erläutert.

Zur kurzen Erläuterung der wesentlichen Grundlagen zu Modellen sowie zur Modellierung stellt die **Unternehmensmodellierung**, als ein wesentlicher Aufgabenbereich der WI, ein geeignetes Instrument dar. Im Rahmen dieser finden Konzepte, Methoden und Werkzeuge zur Modellierung sowohl des Unternehmens und dessen Gesamtsituation im Allgemeinen als auch zur Identifizierung und Konkretisierung von Verbesserungspotentialen im Speziellen Anwendung [467]. Dabei ist die holistische Betrachtung der Organisation von Bedeutung, was zur Folge hat, dass vor allem auch die Relationen zwischen (strategischen) Zielen, Arbeitsabläufen, Unternehmensstrukturen, Produkten und Dienstleistungen berücksichtigt werden [467].

Zur Erreichung dieser Unternehmensziele, die als Ausgangspunkt für die Ableitung der Ziele in der Entwicklung dienen, kann die Wirtschaftsinformatik einen wesentlichen Beitrag leisten, da die Entwicklung, Einführung, Nutzung und Wartung **betrieblicher Informationssysteme** – wie z. B. das in dieser Arbeit vorgestellte IT-System zum Komplexitätsmanagement in der Entwicklung – in deren Fokus steht (s. für Details [181], [51]). Dies stellt eine komplexe Aufgabe dar, da dies i. d. R impliziert viele verschiedene Ansätze (z. B. Anforderungsmanagement, Produktmanagement) zu berücksichtigen [290]. Dabei sind insb. Arbeiten, die sich mit der Konstruktion von Artefakten (z. B. Modelle, Modellierungssprachen, Systemarchitekturen) befassen, äußerst nützlich [178]. Zur Unterstützung einer Vielzahl von Zielsetzungen werden seit einiger Zeit Ansätze zur konzeptuellen Modellierung eingesetzt, wodurch komplexe Systeme und korrespondierende Diskurswelten aus sich ergänzenden, miteinander integrierten Perspektiven mittels konzeptueller Modelle repräsentiert werden können. Da es sich bei Informationssystemen um sprachliche Artefakte handelt, sind bei deren Konstruktion und Anwendung geeignete Sprachen erforderlich [187]. Im Rahmen der Entwicklung und Nutzung von Informationssystemen sind

i. d. R. Menschen beteiligt, die verschiedene Fachsprachen verwenden, weshalb die Notwendigkeit besteht, die Kommunikation zwischen den verschiedenen Personengruppen durch die Bereitstellung geeigneter Sprachen zu verbessern [187]. Daher sind (Modellierungs-)Sprachen das zentrale Forschungsthema der Unternehmensmodellierung [185]. Zunächst wurden hauptsächlich Allzweckmodellierungssprachen (engl. general purpose modelling languages (GPML)) wie bspw. Unified Modeling Language (UML) oder Entity-Relationship-Modelle (ERM) erforscht [178]. In den letzten Jahren stehen v. a. domänenspezifische Modellierungssprachen (engl. domain specific modelling languages (DSML)) und die dazu erforderlichen Methoden im Zentrum der Unternehmensmodellierungsforschung [184].

Der Begriff DSML impliziert die eingehende Beschreibung einer entsprechenden **Domäne**, wie z. B. technische Systeme, oder Klassen von Softwaresystemen, und wird in der Alltagssprache in einer Vielzahl von Kontexten verwendet [185]. Die folgende Beschreibung des Begriffs Domäne ist daher weniger eine Definition als vielmehr eine Charakterisierung [185]: Bei einer Domäne handelt es sich um ein Themengebiet oder einen Interessensbereich, wobei diese nicht nur auf vorhandene Objekte (z. B. i. S. v. Systemen) oder Phänomene beschränkt ist, sondern auch potenzielle Objekte oder Phänomene beinhalten kann. Die Charakterisierung der Domäne kann sowohl durch die Objekte, die es inkludiert, als auch durch Merkmale dieser Objekte sowie durch Funktionen, die durch die Objekte bereitgestellt werden, erfolgen. Dabei können unter Objekten physikalische, menschliche und immaterielle Objekte subsumiert werden [185].

Deshalb erfordert die Analyse einer Domäne die Betrachtung bestehender sprachlichen Repräsentationen dieser Domäne; diese Tatsache wird durch den Begriff **Domäne des Diskurses** (oder Diskursuniversum) ausgedrückt [185]: Demzufolge wird die Domäne des Diskurses durch die Sprache, die in dieser bestimmten Domäne verwendet wird, determiniert. Diese wird durch spezifische Konzepte charakterisiert, die relevante Themengebiete für die Durchführung sinnvoller Aktionen in dieser Domäne widerspiegeln. Diese Konzepte sind i. d. R. nicht genau definiert und somit kann deren Interpretation innerhalb der Domäne variieren [185]. Analog zur Domäne des Diskurses spricht [292] von einem untersuchten System (engl. System under Study (SUS)). Demnach ist ein untersuchtes bzw. zu untersuchendes System ein abgegrenzter Teil der Welt, der eine Menge von Elementen und Interaktionen berücksichtigt [292].

Die Modellierung einer bestimmten Domäne mittels einer Modellierungssprache erfordert **Abstraktion** [185] (tiefergehende Erläuterungen zu

Abstraktionskonzepten sind u. a. [185] zu entnehmen). Zur Charakterisierung bzw. Modellierung einer Domäne ist die Identifikation von Merkmalen bzw. Funktionen notwendig, die bei Interesse berücksichtigt und bei geringer Relevanz ausgeblendet werden [185]. Eine „fortgeschrittene“ Abstraktion nach [187] konzentriert sich nicht nur auf das Weglassen von bzw. den Verzicht auf Aspekte(n) der zu modellierenden Domäne, sondern auch auf deren Vielfalt und Änderung bzw. Abwechslung. Um entscheiden zu können, ob es sinnvoll ist, von einem gewissen Aspekt zu abstrahieren, sind die beiden nachfolgenden Kriterien besonders relevant: Zum einen die Abhängigkeit des Systems vom zu modellierenden Aspekt und zum anderen die Kenntnisse über mögliche Realisierungen des Aspektes (vgl. für ausführliche [187]). Zusammenstellungen von Konzepten zur Abstraktion sind u. a. in [208], [187] zu finden; weiterführende Ausführungen zur Modellqualität können [187] entnommen werden.

Die **Abstraktion dient v. a. der Verringerung von Komplexität** bzw. Kompliziertheit (s. Kapitel 2.2.2) durch Weglassen des je nach Zielsetzung bzw. Zwecks Unwesentlichen [151]. Die wesentliche Intension der Abstraktion ist daher ein komplexes System zielorientiert bzw. zweckmäßig zu modellieren bzw. aufzubereiten und dadurch der weiteren Analyse, Visualisierung und Bewertung zugänglich zu machen. Daher ist es essenziell, dass Modelle in Bezug zur Problemstellung sowie der Situation bzw. Rahmenbedingungen eine ausreichende Aussagefähigkeit besitzen und damit zweckmäßig und problemrelevant sind (s. [237]). Nach EHRENSPIEL & MEERKAMM kann sich die Abstraktion sowohl auf ganze Anforderungsgruppen (z. B. Kosten- und Qualitätsanforderungen), auf einzelne Teilfunktionen (z. B. zuerst zentrale Funktionen) oder auf Beschaffenheitsmerkmale (z. B. Teile, Maße, Werkstoffe) beziehen, was sich auch in den sog. Abstraktionsebenen der Konstruktionsmethodik (vgl. Bild 102 und Bild 103) ausdrückt [151].

Die heterogene Verwendung des **Begriffs Modell** in verschiedenen Disziplinen ist ein Grund für die unterschiedlichen Begriffsdefinitionen in der Literatur (vgl. u. a. [178], [185], [290], [338], [420], [467], [534]). Für den Begriff Modell lassen sich zahlreiche weitere Varianten von Definition unterscheiden, wobei in der Wirtschaftsinformatik nach WILDE & HESS [609] insb. zwischen formalen und konzeptionellen (semi-formalen) Modelle differenziert wird. Nach [290], [533] kann die Klassifikation von Modellen anhand der Sprache, die für ihre Erstellung verwendet wird, erfolgen: Zum einen die nicht-sprachlichen oder ikonischen Modelle, die Zeichen und Symbole verwenden und die eine scheinbare Ähnlichkeit mit den Konzepten der zu modellierenden realen Welt haben und zum anderen sprachliche Modelle,

die Grundelemente (z. B. Zeichen, Buchstaben, Ziffern) verwenden und scheinbar keine Beziehung zu dem modellierten Teil der Realität haben, außer zu dem, der in expliziter Weise definiert ist. Fast alle in der (Wirtschafts-)Informatik verwendeten Modelle sind dem letzteren Sprachtyp zuzuordnen [292].

Nach FRANK resultiert ein **Modell aus einer zweckgerichteten Abstraktion einer Domäne** [185]. SANDKUHL ET AL. verstehen unter einem Modell eine verallgemeinernde Abbildung eines Ausschnitts der realen Welt, wobei nur die relevanten Eigenschaften der Realität für die Modellbildung berücksichtigt werden [467]. Dagegen versteht FRANK unter einer (fortgeschrittenen) Abstraktion nicht nur das Weglassen von Aspekten („Projektion“) sondern auch das Einbeziehen des Umgangs mit Vielfalt und Veränderung in der Diskurswelt [185]. Dabei ist ein Modell explizit nicht auf die Annahme begrenzt, dass ein Original existiert [185]. Da die spezifischen Eigenschaften mentaler Modelle nicht Gegenstand dieser Arbeit sind, wird nachfolgend diese allgemeine Charakterisierung eines Modells durch das Modell als Artefakt verfeinert [185]. Demnach ist ein Modell als Artefakt eine Abbildung, die das Resultat einer gezielten und bewussten Konstruktion ist [185]. Zudem ist das Modell wahrnehmbar und kommunizierbar [185] (vgl. auch Bild 120). Die Abstraktionen, auf denen es basiert, sind abhängig von dem Zweck der Konstruktion, von mentalen Aktivitäten der an der Konstruktion involvierten Akteure sowie zusätzlich von externen Reizen und Zwängen [185].

Dagegen ist nach KURTEV ET AL. ein **Modell eine Repräsentation eines gegebenen (zu untersuchenden) Systems**, ein gerichteter Multigraph, der aus einer Menge von Knoten, einer Menge von Kanten und einer Abbildungsfunktion zwischen den Knoten und Kanten besteht, wobei Knoten mit mehr als einer Kante verbunden sein können [338]. Dieser Begriffsdefinition wird auch in der Arbeit von [292] gefolgt. Ähnlich dazu beschreibt [178] ein Modell als eine Repräsentation eines gewissen Ausschnitts der betrieblichen Realität. Modelle werden hier als das Ergebnis eines Modellierungsprozesses verstanden, in dem Modellierungsmethoden und -sprachen eingesetzt werden [178].

Der Fokus auf sprachliche Konstruktionen führt unweigerlich zum **Begriff des konzeptuellen Modells** [185]. Ein konzeptuelles Modell ist nach FRANK durch folgende Merkmale gekennzeichnet: sprachliche Konstruktion, Abstraktion von besonderen Fällen, Unabhängigkeit von technologischen Veränderungen, Übereinstimmung mit der gesprochenen Sprache, integrative Abstraktion und Rekonstruktion von Konzepten [185]. Dagegen

repräsentiert ein konzeptuelles Modell nach KARAGIANNIS & VISIC, welches auch als Domänen-Modell bekannt ist, Konzepte (Entitäten) und Relationen zwischen diesen; zudem sind Konzeptmodelle unabhängig vom Entwurf und der Implementierung [292] (vgl. [392]). Der Zweck eines konzeptuellen Modells ist es, die Bedeutung der Begriffe und Konzepte, die von Domänenexperten (z. B. Produkt- und Prozessingenieure, vgl. Bild 3 in Kapitel 1.4) verwendet werden, auszudrücken, die Aufgabenstellung zu diskutieren sowie die richtige Beziehung zwischen verschiedenen Konzepten zu finden [292]. Mit Hilfe konzeptueller Modelle wird versucht, die Bedeutung der verschiedenen, meist mehrdeutigen Begriffe zu klären und Probleme aufgrund abweichender Auslegungen der Begriffe und Konzepte zu vermeiden [292].

In Bild 116 sind die zuvor **skizzierten Fachbegriffe zueinander in Beziehung gesetzt** (vgl. Verständnis in [185]). Obwohl diese Abbildung darauf hindeutet, dass mentale Modelle einfach existent sind, kann davon ausgegangen werden, dass diese durch Interaktion und Kommunikation und demzufolge durch die Domäne des Diskurses als auch durch Konzeptmodelle beeinflusst werden [185]. Gleichzeitig werden Konzeptmodelle nicht durch entsprechende Domänen von Diskursen festgelegt, sondern können auch zum Wandel zugehöriger natürlicher Sprachen beitragen [185]. Diese (sprachphilosophische) Erwägung ist insb. für die Konstruktion und Nutzung von Informationssystemen von Bedeutung. Es zeigt sich, dass die Sprache, die wir verwenden (Domäne des Diskurses), als Grundlage für Informationssysteme, die wir entwerfen (durch entsprechende Konzeptmodelle), dient und, dass die Informationssysteme, die wir verwenden, die Sprache, die wir sprechen, verändern kann [185].

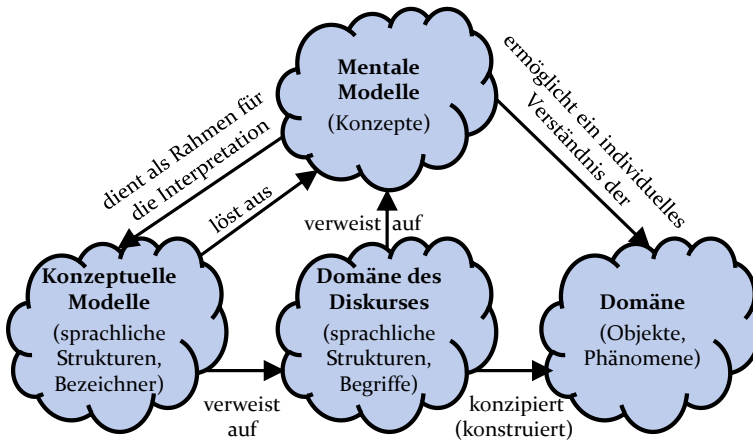


Bild 116: Semiotisches Rechteck, angepasst vom semiotischen Dreieck (in Anlehnung an [185])

Hauptmerkmale eines allgemeinen Modells

Deshalb kann nach SCHÜTTE ein **Modell als das Ergebnis eines Modellierungsprozesses** eines Modellierers, der für den Nutzer des Modells eine Repräsentation eines Originals bzw. eines Realitätsausschnitts zu einer bestimmten Zeit als relevant mit Hilfe einer geeigneten Sprache i. w. S. deklariert, aufgefasst werden [500]. In Bild 117 wird das Modell als Konstruktionsleistung ausgehend von einem beliebigen Original (z. B. ein komplexes, technisches System) über interne, mentale Modelle bis zu externen, expliziten Modellen dargestellt. In dieser Abbildung wird die Zeit, die Auskunft über den Erstellungszeitpunkt und die Gültigkeit des Modells gibt, nicht berücksichtigt [500].

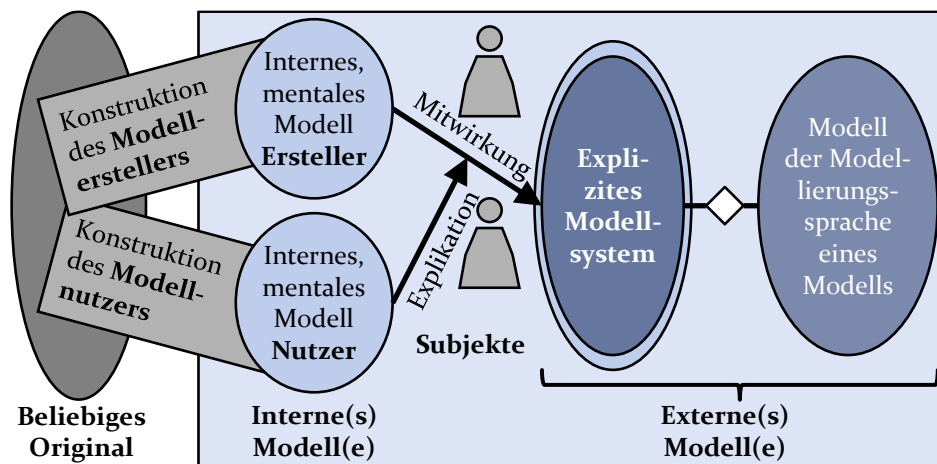


Bild 117: Modelle als Konstruktionsleistung in Anlehnung an [500])

Die Zweckorientierung geht auf die Charakterisierung des Modellbegriffs von STACHOWIAK zurück [524]. Er vertritt trotz der großen Vielzahl verschiedener Modellarten die Auffassung, dass alle Modelle auf den drei folgenden **charakteristischen Hauptmerkmalen eines allgemeinen Modells** beruhen, welche in STACHOWIAK ausführlich diskutiert werden [524]:

- Infolge des **Abbildungsmerkmal** sind Modelle Darstellungen bzw. Repräsentationen eines natürlichen oder künstliches Originals, das selbst ein Modell sein kann;
- Gemäß dem **Verkürzungsmerkmal** erfassen Modelle i. d. R. nicht alle Aspekte bzw. Attribute des Originals, sondern nur diejenigen, welche den jeweiligen Modellerstellern und/oder Modellnutzern als relevant erscheinen (s. auch Bild 117);
- Nach dem **pragmatischen Merkmal** sind Modelle ihren Originalen nicht allgemein eindeutig zugeordnet, da sie ihre Substitutions-

bzw. Ersetzungsfunktion erstens für bestimmte Subjekte, zweitens innerhalb bestimmter Zeitperioden und drittens mit Beschränkung auf bestimmte fiktive oder reale Operationen erfüllen.

Gültigkeitsprüfung von Modellen

Die Grundlage dieses Abschnitts ist insb. die Publikation „Grundlagen der Systemanalyse“ von GRONAU [224] mit Modifikationen von TRIER ET AL. [559] und demzufolge wird nachfolgend auf ein Referenzieren der zwei Publikationen verzichtet wird. Intension des Abschnitts ist die Beschreibung von **vier möglichen Methoden zur Gültigkeitsprüfung von Modellen**. Dabei sei angemerkt, dass Gültigkeitsprüfungen im Rahmen der Modellierung vorzugsweise kontinuierlich und mindestens nach den Modellierungsschritten durchzuführen sind, um größere Iterationen und Fehler zu vermeiden (s. z. B. Bild 122). Allerdings ist der Nachweis einer (uneingeschränkten) Gültigkeit bzw. Korrektheit von Modellen aufgrund der Subjektivität im Rahmen der Modellerstellung nicht möglich (vgl. vorhergehender Abschnitt).

Die nachfolgend aufgezählten Methoden stehen grundsätzlich für eine Gültigkeitsprüfung von Modellen zur Verfügung. Dabei ist im Hinblick auf die **Reihenfolge der Anwendung dieser Methoden** zu beachten, dass vor der Validierung sowohl die beiden Methoden Verifikation und Kalibrierung vorzunehmen und abzuschließend sind, da mit der Validierung die verifizierten und kalibrierten Modelle bewertet werden. Dagegen ist die Anwendung der Sensitivitätsanalyse schon während der Verifikation und Kalibrierung hilfreich und daher zu empfehlen.

Mittels der **Verifikation** werden zum einen die benutzten Daten überprüft und zum anderen deren korrekte Implementierung bzw. Verwendung im Modell nachgewiesen. Zu Beginn der Verifikation wird u. a. die Richtigkeit der Ausgangsdaten überprüft. Danach erfolgt die Überprüfung der schrittweisen erstellten Abbildungen u. a. auf Korrektheit und Vollständigkeit bzgl. auf Darstellung und Implementierung der einzelnen Relationen.

Mittels der **Kalibrierung** wird das Gesamtverhalten des Modells möglichst an die wahrgenommene Realität angeglichen. Dies erfolgt durch schrittweise Prüfung und Angleichung des Verhaltens auf Basis von Vergleichen des Outputs und von Parameteränderungen. Dabei sind Parameter die Größen, die während der Simulation des Verhaltens gleich bzw. konstant bleiben und die Stärke der Wechselwirkungen zwischen einzelnen Eigenschaften ausdrücken. Vorrangig zu ändern sind die Parameter, welche bei der Systemmodellierung nicht hinreichend genau modelliert werden

konnten. Das Ergebnis der Kalibrierung ist eine subjektiv ausreichende Ähnlichkeit zwischen Modell und Original (i. S. v. empirische Gültigkeit).

Mittels der **Sensitivitätsanalyse** wird die Sensibilität des Outputs in Abhängigkeit der Modifikationen von bestimmten Parametern ermittelt. Bereits während der Verifikation und Kalibrierung können Sensitivitätsanalyse hilfreich sein, um mögliche Strukturfehler im Modell zu identifizieren. Dabei können auch diejenigen Modellparameter identifiziert werden, welche i. S. v. Verbesserungen des Modellverhaltens zu modifizieren sind (d. h. also die für das Verhalten unwesentlichen und wesentlichen Einflussgrößen). Mittels der Sensitivitätsanalyse lassen sich auch Daten ermitteln, die für die Applikation der Simulationsergebnisse auf das reale System bzw. Original eine gewisse Nützlichkeit haben (z. B. obere und untere Grenzwerte von kritischen Parametern). Das Resultat der Sensitivitätsanalyse ist i. d. R. die Visualisierung der quantitativen Abhängigkeiten des Outputs von den Modifikationen einzelner oder mehrerer Parameter des Modells.

Mittels der **Validierung** wird nach der Verifizierung und Kalibrierung das verifizierte und kalibrierte Modell bewertet. Mit der Validierung wird die Modellgüte für einen zuvor festgelegten Modellzweck bestimmt. Dabei ist die Güte ein qualitatives Maß für die Zielangemessenheit oder pragmatische Gültigkeit und somit wird auf Basis der Güte beurteilt, ob die mittels des Modells gewonnene Ergebnisse und Erkenntnisse für den Zweck akzeptabel sind. Diese Beurteilung kann entweder durch den Vergleich mit alternativen Modellen oder durch den Beleg, dass die Aufgabenstellung durch das Modell hinreichend abgebildet wird, vorgenommen werden. Daher ist auch bei der Validierung die Subjektivität der Beurteilung das zentrale Problem. Mit Hilfe einer ausführlichen Dokumentation kann zumindest eine intersubjektive Nachvollziehbarkeit sichergestellt werden.

Unterscheidung zwischen deskriptiven und präskriptiven Modellen

Ein Modell nach STACHOWIAK ist ein Abbild eines Originals, in dem nicht alle Aspekte bzw. Attribute des Originals dargestellt sind, sondern lediglich diejenigen, die in Bezug auf einen bestimmten Zweck relevant sind [524]. Da ein **Modell auf einen bestimmten Anwendungs- bzw. Verwendungszweck hin ausgerichtet**, kann zum einen dasselbe Original durch eine Vielzahl verschiedener Modelle dargestellt werden [524] und zum anderen ein Modell des gleichen Originals verschiedene Aspekte beinhalten oder vernachlässigen [89]. Weitere Details sind in [524], [429] zu finden.

Dabei wird grundsätzlich zwischen **deskriptiven und präskriptiven Modellen** unterschieden. Während mit einem deskriptiven Modell bspw. ein

tatsächlich ablaufender Prozess dargestellt wird, um diesen besser verstehen und u. a. die Kosten und die Dauer näherungsweise ermitteln zu können, wird mit einem präskriptiven Modell ein Standardprozess repräsentiert, um Anweisungen für die Ausführung von Prozessschritten zu machen (vgl. [89]). Im Gegensatz zu deskriptiven Modellen ist bei präskriptiven Modellen die reale Ausführbarkeit sowie die Effektivität und Effizienz z. B. bei Prozessmodellen nicht relevant [89]. Daher postulieren KASTENS & KLEINE BÜNING, dass Modelle deklarativ oder deskriptiv sein müssen, was zur Folge hat, dass das Modell immer gültige Aussagen zu den folgenden Aspekten des Originals macht [296]:

- Struktur, d. h. Zusammensetzung des Originals aus Bestandteilen,
- Eigenschaften, d. h. von Teilen des Originals,
- Beziehungen, d. h. zwischen Teilen des Originals) und
- Verhalten, d. h. des Originals bei Operationen.

Diese abbildungsorientierte Definition des Modellbegriffs kritisieren die Befürworter der konstruktionsorientierten Definition mit dem Argument, dass Modelle als Repräsentationen den Anspruch haben Ähnlichkeit zur Realität zu besitzen (vgl. u. a. [500], [13]). Allerdings lässt sich dieser **Anspruch nach Homomorphie nicht formal nachweisen**, auch wenn dieser häufig für möglich gehalten wird [500].

Unterscheidung zwischen Isomorphie und Homomorphie

Im Hinblick auf das Abbildungsmerkmal ist es sinnvoll, zwischen den zwei Begriffen der **Isomorphie (Strukturgleichheit) und der Homomorphie (Strukturähnlichkeit) zu differenzieren** [559] (vgl. Bild 118).

Nach TRIER ET AL. besteht **Isomorphie** zwischen dem System S und einem Modell M dann, wenn

- jedem Element E des Systems S ein Element im Modell M eindeutig zugeordnet ist und diese Zuordnung auch umgekehrt eindeutig ist,
 - jeder Relation R im System S eine Relation R im Modell M eindeutig zugeordnet ist und diese Zuordnung auch umgekehrt eindeutig ist,
 - einander zugeordnete Relationen R ausschließlich einander zugeordnete Elemente E enthalten;
- ➔ folglich sind bei einer isomorphen Abbildung eines Systems S alle Elemente E und alle zwischen diesen bestehenden Relationen R des Systems S (Originals) auch im Modell M zu finden und somit sind die Strukturen des Originals und des Modells identisch [559].

Dagegen besteht nach TRIER ET AL. **Homomorphie** zwischen dem System S und einem Modell M dann, wenn

- jedem Element E des Modells M zwar ein Element E vom System S eindeutig zugeordnet ist, jedoch nicht umgekehrt,
 - jeder Relation R des Modells M zwar eine Relation R des Systems S eindeutig zugeordnet ist, jedoch nicht umgekehrt,
 - die Relationen R des Modells M ausschließlich Elemente E aufweisen, denen ein Element E vom Systems S zugeordnet werden kann;
- ➔ folglich sind bei einer homomorphen Abbildung eines Systems S nicht alle Elemente E und Relationen R des Systems S (Originals) im Modell M wiederzufinden und somit sind die Strukturen des Originals und des Modells nicht identisch, da diese lediglich eine Ähnlichkeit aufweisen [559] (s. Bild 118).

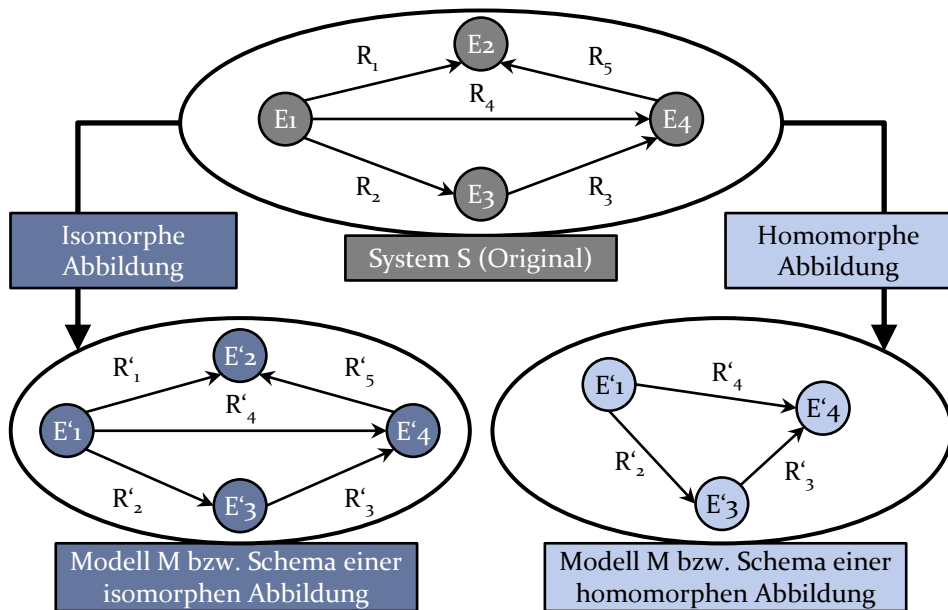


Bild 118: Isomorphe und homomorphe Abbildung (in Anlehnung an [559])

Modellraum des Konstruierens nach RUDE

Ein bekanntes **Beschreibungsmodell für technische Systeme**, das auf verschiedenen Ebenen die Produktkonkretisierung erschließt, ist der Modellraum des Konstruierens nach RUDE. Ein weiteres Beispiel hierfür ist das Pyramidenmodell der Produktkonkretisierung nach EHRENSPIEL & MEERKAMM [151] in Bild 103. Weitere Beispiele s. u. a. in [174] und [455] zu finden.

Im Modellraum **des Konstruierens nach RUDE** werden den vier übereinander angeordneten Konkretisierungsstufen bzw. Teilmodellräumen Anforderung, Funktion, Prinzip und Gestalt jeweils passende Partialmodelle eines integrierten Produktmodells zugeordnet (s. Bild 119 und [437]).

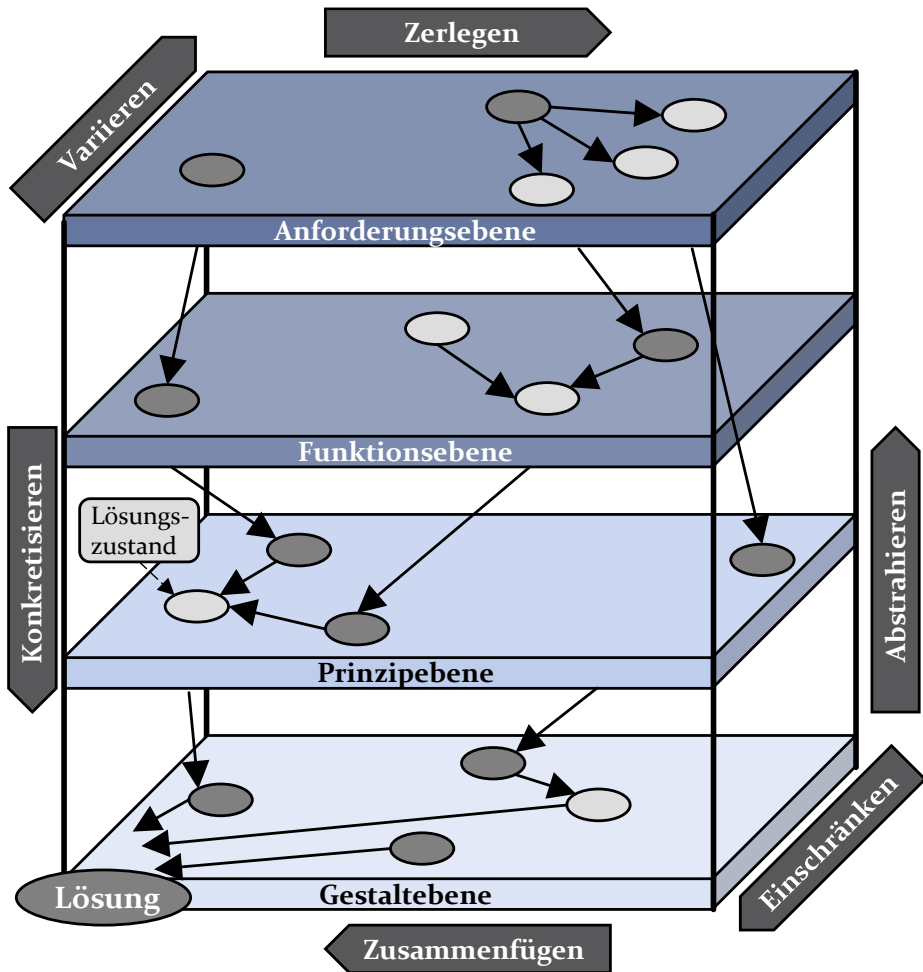


Bild 119: Modellraum des Konstruierens (nach [464] und ergänzt gemäß [437])

Die drei Dimensionen Konkretisierungsgrad, Zerlegungsgrad, Variationsgrad spannen den sog. **Modellraum des Konstruierens** nach RUDE auf und dienen der Verortung bzw. Ordnung von Entwicklungsergebnissen [464]. Der Modellraum kann auch als Navigationsmodell der Entwicklung aufgefasst werden [437]. Als Beispiel dient der Übergang von der Anforderungs- auf die Funktionsebene, indem auf Basis einer Spezifikation die Erstellung der Funktionsstruktur erfolgt (d. h. Konkretisierung) [437]. Die Festlegung eines Gesamtkonzepts mittels eines Morphologischen Kastens, der zu verschiedenen Teilfunktionen alternative Wirkprinzipien und somit alternative Lösungsmöglichkeiten enthält, ist ein Beispiel für die Navigation in Bezug auf die Dimensionen Zerlegungs- und Variationsgrad (d. h. Kombination und Einschränkung) [437].

Der **Abstraktions- bzw. Konkretisierungsgrad** der in der Entwicklung genutzten Modelle von komplexen, technischen Systemen variiert (i. S. v. Navigation vom Abstrakten zum Konkreten und umgekehrt), ist u. a. abhängig von der jeweiligen Entwicklungssituation, der Entwicklungsphase sowie den Entwicklungszielen (s. [437]). Ein bekanntes Beispiel für ein Beschreibungsmodell von Entwicklungsprozessen ist der Vorgehensplan nach der VDI-Richtlinie 2221 (s. [580], [581]), der den Prozess des steigenden Konkretisierungs- bzw. Detaillierungsgrads der entwickelten Produktmodelle skizziert.

Neben dem Abstraktions- bzw. Konkretisierungsgrad ist der **Zerlegungsgrad des Systems** die zweite wesentliche Dimension im Modellraum bzw. Navigationsmodell der Entwicklung. Die entsprechenden Tätigkeiten sind auf der einen Seite das Zerlegen und Detaillieren (d. h. Erhöhung des Zerlegungsgrades) und auf der anderen Seite das Kombinieren und Zusammenfügen (d. h. Reduzierung des Zerlegungsgrades) [437]. Die Zerlegung eines Problems in besser überschaubare und somit leichter handhabbare und v. a. lösbare Teilprobleme ist ein zentraler Schritt bei der Lösung eines (größeren) Gesamtproblems (vgl. Grundprinzip der Problemzerlegung [133]). Anschließend sind die entwickelten Teillösungen zur Gesamtlösung zu kombinieren, was auch dem grundlegenden V-Modell entspricht, nach dem im Systementwurf (linker Ast des Vs) der Zerlegungsgrad erhöht und in der Systemintegration (rechter Ast des Vs) der Zerlegungsgrad reduziert wird [437].

Die dritte Dimension ist der **Variationsgrad**, der zu einem bestimmten Zeitpunkt die betrachtete Menge an alternativen Lösungen ordnet [464]. Die hierfür erforderlichen Tätigkeiten sind einerseits das Variieren (d. h. Erhöhung des Variationsgrades) und andererseits das Einschränken bzw. Festlegen (d. h. Reduzierung des Variationsgrades) [437]. Zudem ist dieses Denken in Varianten bzw. Alternativen auch ein zentrales Grundprinzip der Problemlösung im Allgemeinen sowie der Entwicklung [238], um realistische Lösungsalternativen zu bereits vorhandenen Lösungen zu finden, wodurch auch die Chance auf innovative Lösungsoptionen steigt [437] (vgl. z. B. TRIZ-Philosophie).

A.13 Modellierung und Modellierungsansätze

In diesem Anhang sind tiefergehende **Erläuterung zur Modellierung** von Modellen mittels grundsätzlicher Modellierungsansätze zu finden. Daher gliedert sich dieser Anhang in die folgenden Abschnitte: Grundlagen zum Modellierungsvorgehen, Grundlagen zu Modellierungsansätzen, Grundsätze der Modellierung und potenzielle Fehlerquellen bei der Modellierung.

Grundlagen zum Modellierungsvorgehen

Die **Modellierung von komplexen, technischen Systemen** ist für die Analyse von Systemen mit deren Teilsystemen essenziell, da diese aufgrund ihrer Komplexität nicht vollständig erfasst und somit analysiert werden können (vgl. [559]). Zudem sind Originalsysteme bzw. komplexe Systeme selbst für die „Simulation“ von Auswirkungen und Konsequenzen in Folge von manipulativen Modifikationen der Systemelemente und/oder -relationen oft ungeeignet. Daher ist die Erstellung geeigneter Modelle zur Analyse und Bewertung von Systemen erforderlich. Typische Fragen im Rahmen der Modellierung sind u. a. welche Ziele und Zwecke werden durch das Modell (nicht) verfolgt sowie welche und wie viele Elemente und Relationen des Systems sind zu modellieren (vgl. u. a. [559]). Um eine effektive und effiziente Modellerstellung und -interpretation durch alle Beteiligten zu unterstützen, wurden mannigfache Modellierungsansätze für diverse Modelle mit vielfältigen Zwecken für viele Anwendungsbereiche entwickelt (vgl. Tabelle 15 im Anhang A.11).

Das Ergebnis einer Modellierung kann ein internes oder externes Modell eines Originals sein. Die methodische Vorgehensweise bei der Modellierung von Systemen lässt sich als heuristischer, iterativer und rückgekoppelter Prozess charakterisieren [559]. Das allgemeine Vorgehen vom Original über interne, mentale Modelle bis zu externen Modellen ist in der folgenden Darstellung schematisch skizziert (s. Bild 120). Im Rahmen des **Modellierungsvorgehens** nimmt die Person im internen, mentalen Modell die ihrer Meinung nach für einen bestimmten Zweck relevanten Aspekte des Originals (z. B. eines technischen Systems) wahr. Dabei werden durch Abstraktion die unwesentlichen Aspekte vernachlässigt und die relevanten Aspekte im externen Modell abgebildet. Dabei die folgenden drei aus dem pragmatischen Merkmal des Modellbegriffs ableitbaren Zielkriterien durch den Modellersteller zu befolgen: Relevanz der abgebildeten Originalaspekte (Problemkonformität), Richtigkeit und Genauigkeit der Abbildung sowie Modelltransparenz (intersubjektive Nachvollziehbarkeit) [559]. Nach der Modellerstellung kann das externe Modell manipuliert werden, um zielgerichtet Wissen über das Modell und damit auch über das

Original zu erlangen (i. S. v. Analogieschlüsse). Dadurch können nicht nur Erkenntnisse und Erklärungen zu Aspekten des Originals (z. B. Systemeigenschaften und -verhalten), sondern auch Prognosen zu zukünftigen Aspekten des modifizierten Originals gewonnen werden (vgl. Wahrnehmung). Abschließend können die abgeleiteten Schlussfolgerungen in Form von Maßnahmen auf das Original angewendet werden (vgl. [559]).

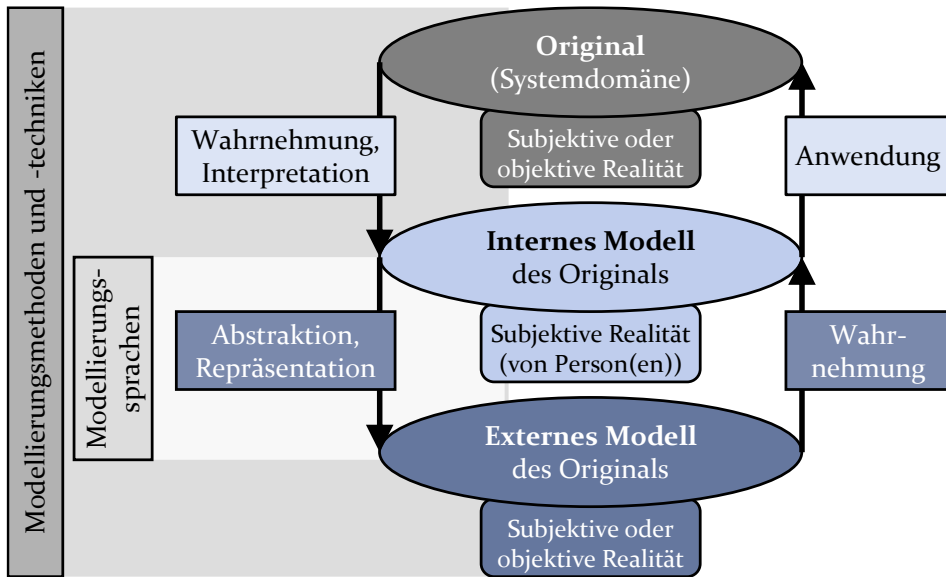


Bild 120: Vorgehen zur Repräsentation eines Originals durch ein Modell (in Anlehnung an [170], [37] und [559])

Grundlagen zu Modellierungsansätzen

Die links in Bild 120 dargestellten Modellierungsmethoden und -techniken dienen zur Unterstützung im Rahmen der Modellierung und werden nachfolgend kurz erläutert. Die zu den Modellierungsmethoden gehörenden Modellierungssprachen dienen v. a. zur Abstraktion und Repräsentation des externen Modells. Für die Modellerstellung stehen nach TRIER ET AL. die folgenden **drei Modellierungsansätze** grundsätzlich zur Verfügung: Top-Down-, Bottom-Up und hybrider Ansatz (s. [559]):

- **Top-Down-Ansatz:** Ausgehend vom Gesamtsystem mit Relationen zur Umwelt werden relevante Aspekte des Systems in schrittweiser Konkretisierung im Modell abgebildet. Dann werden die Eigenschaften der Subsysteme sukzessive auf die jeweils untergeordneten Modellebenen (vgl. Bild 14) bis zu einem intendierten Konkretisierungsgrad abgebildet. Dieser Ansatz ist v. a. bei der (Neu-)Gestaltung von Systemen vorteilhaft, da die Integration der einzelnen

Subsysteme erleichtert wird. Von Nachteil ist, dass auf einer hohen Abstraktionsebene noch nicht erkennbar ist, ob ein noch abstrakt geplanter Bestandteil eines Systems auch wirklich realisierbar ist. Daher können Entscheidungen auf höheren Abstraktionsebenen revidiert werden müssen und die Konkretisierung von dort erneut erfolgen (vgl. Kapitel 2.3.2).

- **Bottom-Up-Ansatz:** Ausgehend von der untersten Abstraktionsebene (d. h. von einzelnen Elementen und Relationen) wird mit der Modellierung begonnen. Die Realisierbarkeit einzelner Systembestandteile kann aufgrund des hohen Konkretisierungsgrads gut bewertet werden (vgl. Bild 14). Dabei ist es im Rahmen der (Neu-)Gestaltung von Systemen problematisch, dass a priori nicht beurteilt werden kann, ob das Zusammenfügen der verschiedenen Elemente (z. B. Bauteile) zum Gesamtsystem prinzipiell möglich ist. Zudem besteht die Möglichkeit zur Doppelarbeit (i. S. v. Mehrfachgestaltung), falls gleiche Elemente in unterschiedlichen Subsystemen nicht identifiziert werden. Ferner ist der Ansatz auch für die Systemanalyse von Nachteil, da aufgrund des fehlenden Überblicks über das System ein systematisches Vorgehen im Hinblick auf die Strukturierung und Abgrenzung von Systemgrenzen schwierig ist.
- **Hybrider Ansatz:** Der in der Praxis bewährte Ansatz geht zwar generell vom Top-Down-Ansatz aus, ist aber eine Kombination aus dem Top-Down- und Bottom-Up-Ansatz. Dabei wird insb. bei komplexen, kritischen Modellierungsschritten (z. B. bei ungenügenden Kenntnissen bzgl. eines Systembestandteils) stark konkretisiert, um dann die erlangten Erkenntnisse i. S. v. bottom-up auf einer höheren Abstraktionsebene zu nutzen. Zur Überprüfung der Konsistenz und Korrektheit des Modells sind sog. „System Reviews“ in regelmäßigen Abständen und an bestimmten Meilensteinen sinnvoll.

In Folge der subjektiven Realität von Modellen und deren Erstellung (vgl. Bild 120) ist das **Modellverständnis** besonders wichtig, um bspw. die Gültigkeit, Aussagekraft und weitere Besonderheiten von Modellen beurteilen zu können (s. Anhang A.12 für Details).

Grundsätze der Modellierung

Aufgrund der Bedeutung der Informationsmodellierung für die Güte, Aussagefähigkeit und Qualität von Anwendungssystemen, wie z. B. dem in dieser Arbeit vorgestellten IT-System zum Komplexitätsmanagement in der Entwicklung, sind bestimmte Grundsätze der Modellierung zu beachten, um eine ordnungsmäßige und qualitätsgerechte Modellerstellung zu

realisieren. Diese sog. **Grundsätze ordnungsmäßiger Modellierung** nach BECKER ET AL. formulieren zentrale Richtlinien, die der Sicherstellung von Qualität und Vergleichbarkeit von Modellen stehen [40]. Neben Aspekten zur syntaktischen und semantischen Korrektheit der Modelle (vgl. Anhang A.15) beziehen die Grundsätze auch repräsentative, organisatorische und ökonomische Aspekte mit ein [40]. Für eine Vertiefung der Inhalte der Grundsätze ordnungsmäßiger Modellierung empfiehlt sich [40], worauf auch dieser Abschnitt basiert, so dass innerhalb des Abschnitts auf ein Referenzieren dieser Publikation verzichtet wird. Ziel des Abschnitts ist es, einen Überblick über die sechs Grundsätze ordnungsmäßiger Modellierung nach BECKER ET AL. zu geben, da diese die substanzielle Grundlage für gute Modelle darstellen (s. Bild 121).

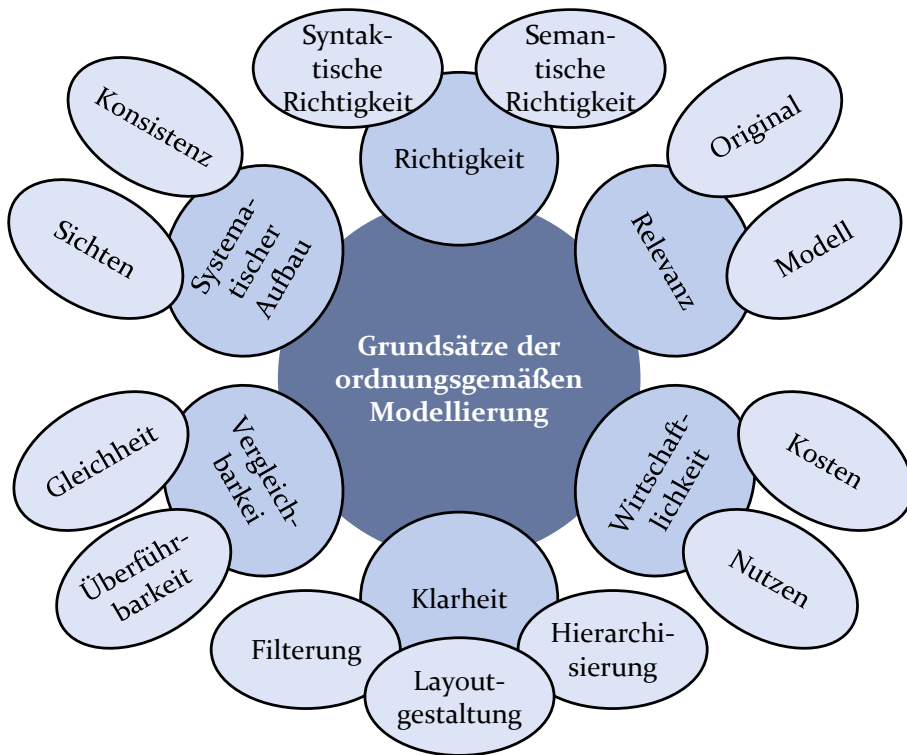


Bild 121: Grundsätze ordnungsmäßiger Modellierung (inhaltlich in Anlehnung an [40])

Mit Blick auf den Grundsatz der **Richtigkeit** wird zwischen syntaktischer und semantischer Richtigkeit differenziert. Dabei ist die syntaktische Richtigkeit gegeben, wenn das Modell alle Regeln, die mittels einer bestimmten Modellierungssprache vorgegeben sind, einhält. Die semantische Richtigkeit eines Modells ist (vereinfacht) dann erreicht, wenn durch das Modell zum einen vernünftig repräsentiert, was der Modellerstell beabsichtigt hat,

und zum anderen ausreichend zweckmäßig aus Sicht des Modellnutzers ist (vgl. Anhang A.14 für Details zur Modellierungssprache und Anhang A.15 für den Zusammenhang zu Metamodellen für diverse Details).

Gemäß dem Grundsatz **Relevanz** wird das Modell aus zwei Sichtweisen begrenzt. Auf der einen Seite soll das Modell alle Sachverhalte umfassen, die im entsprechenden Realitätsausschnitt bzw. der Domäne (vgl. Bild 116) für den Modellierungszweck dienlich sind. Auf der anderen Seite soll das Modell nichts enthalten, was nicht auch sein entsprechendes Gegenstück im Realitätsausschnitt bzw. der Domäne hat. Diese zwei Ansprüche werden als externe und interne Minimalität bezeichnet.

Der Grundsatz der **Wirtschaftlichkeit** der Modellierung sagt aus, dass ein gefordertes Modellierungsziel mit minimalem Aufwand realisiert werden soll bzw. dass mit einem vorgegebenen Aufwand ein Modell modelliert werden soll, dass den Modellierungszweck bestmöglich erfüllt. Daraus folgt, dass ein Modell so weit konkretisiert werden soll (vgl. Bild 14 in Kapitel 2.3.2), bis die zusätzlichen Kosten der Konkretisierung sich gerade noch mit dem zusätzlichen Nutzen durch die Konkretisierung die Waage halten. Im Rahmen der Modellierung ist die Nutzung von Referenzmodellen (vgl. Anhang A.15) aus Gründen der Wirtschaftlichkeit sinnvoll, da der Modellersteller bestehende Modelle (mit best und common practices) verwenden und lediglich auf die unternehmensspezifischen Rahmenbedingungen und domänenspezifischen Charakteristika adaptieren muss.

Mit dem Grundsatz der **Klarheit** ist v. a. die Verständlichkeit der Modelle gemeint, also insb. einfache Lesbarkeit, Deutlichkeit und Anschaulichkeit (z. B. mittels schlüssigen Begriffs-, Struktur- und Referenzbausteinen). Dabei dient eine Hierarchisierung von Modellen zur einfachen Navigation durch diverse Modelle über verschiedene Stufen. Mit Hilfe einer zweckgerichteten Layoutgestaltung lassen sich Modellelemente übersichtlich strukturieren und Relationen möglichst überschneidungsfrei anordnen. Außerdem kann durch Filtern (i. S. v. Ein- und Ausblenden) bestimmter Modellelemente nach gewissen Kriterien die Übersichtlichkeit und Lesbarkeit von Modellen erhöht werden.

Bzgl. des Grundsatzes **Vergleichbarkeit** sind Aspekte der Gleichheit und Überführbarkeit zu unterscheiden. Damit sich die wirkliche Gleichheit von Sachverhalten im Realitätsausschnitt bzw. der Domäne auch in der Gleichheit von Elementen und Relationen im Modell widerspiegelt, sind (die) identischen Sachverhalte (in) der Real- bzw. Vorstellungswelt auch mit bzw. innerhalb einer Modellierungssprache gleich zu modellieren (Aspekt der Gleichheit). Damit Modelle (z. T. für denselben Realitätsausschnitt),

die in verschiedenen Modellierungssprachen modelliert sind, miteinander verglichen werden können, müssen diese ineinander überführt werden können (Aspekt der Überführbarkeit). Die Überführbarkeit von Modellen wird erleichtert, falls diese auf ähnlichen Konstrukten aufbauen und durch Metamodelltransformation überführbar sind (vgl. Anhang A. 15).

Mit Blick auf den Grundsatz des **systematischen Aufbaus** sind die beiden Forderungen nach hoher Konsistenz (i. S. v. Zusammenhang, Logik) und nach sinnvollen Sichten wichtig. Da oft Sachverhalte aus einem Realitätsausschnitt bzw. einer Domäne aus unterschiedlichen Perspektiven (z. B. Prozess- und Organisationssicht) modelliert werden, sollten bspw. in einem Prozessmodell integrierte Organisationselemente so modelliert sein, wie diese auch im Organisationsmodell beschrieben sind. Dadurch kann die Konsistenz domänen- bzw. modellübergreifend (i. S. v. eines Gesamtmodells) konsistent gewährleistet werden.

Mit diesen sechs Grundsätzen ordnungsmäßiger Modellierung werden relevante Ziele i. S. v. von **Vorgaben spezifiziert**, die bei der Modellierung vom Modellersteller zu befolgen sind. Dabei lassen sich verschiedene Vereinbarungen möglicherweise mehreren Vorgaben zuordnen. So dienen die Verwendung von Begriffs- und Referenzbausteinen zum einen dem Grundsatz der Klarheit, da die verwendeten Fachbegriffe vertraut sind, und zum anderen auch dem Grundsatz der Vergleichbarkeit, da identische Sachverhalte in gleicher Art und Weise abgebildet sind.

Potenzielle Fehlerquellen bei der Modellierung

Die Modellierung von bspw. komplexen Systemen erfolgt durch Personen (i. S. v. Subjekten), womit eine gewisse Subjektivität jeder Modellerstellung zugrunde liegt. In Folge dieser subjektiven Modellierung und/oder auch Ungenauigkeiten und/oder Unsicherheiten von diversen technischen Aktivitäten (z. B. Laboranalysen, Messungen) wird die Gültigkeiten von Modellen eingeschränkt. Mit Hilfe des in Bild 122 dargestellten Modellierungsvorgehens gemäß GRONAU werden zu den diversen Schritten **potenziellen Fehlerquellen der Modellierung** dargestellt und verortet [224].

Die begrenzten sensorischen und kognitiven Fähigkeiten von Menschen schränken **Wahrnehmung der Realität** im Rahmen der Modellierung besonders stark ein [559]. Diese lassen sich in ihrer Auswirkung mit technischen Hilfsmitteln abschwächen, wie bspw. mittels geeigneter Visualisierungen zur Verbesserung des Verständnisses von z. B. komplexen, technischen Systemen. Weitere Erläuterungen zu den potenziellen Fehlerquellen sind in [224] und [559] zu finden.

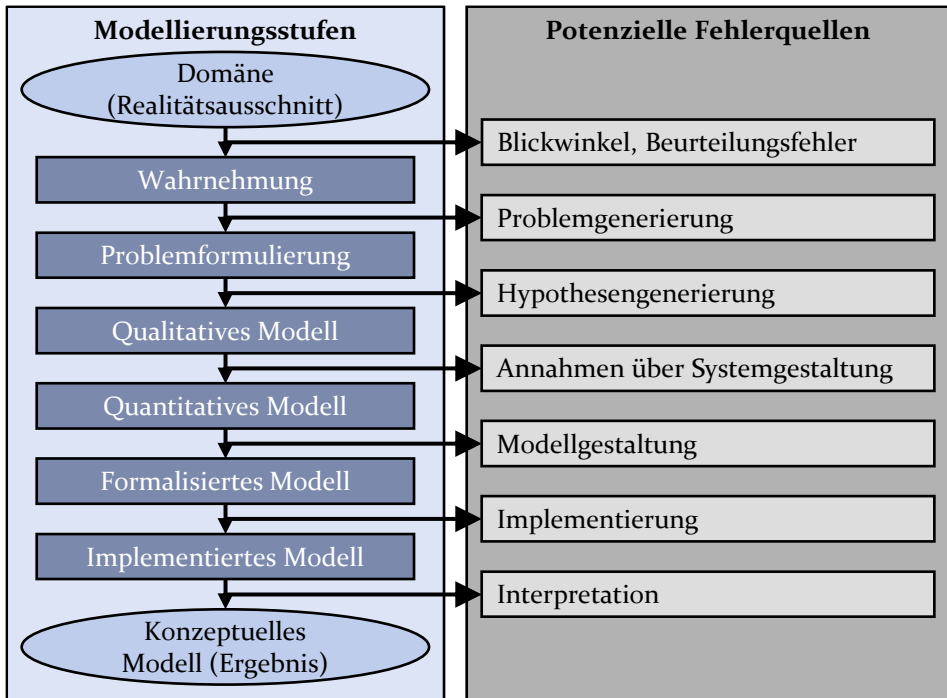


Bild 122: Potenzielle Fehlerquellen der Modellierung (nach [224])

A.14 Modellierungsmethoden und -sprachen

In diesem Anhang werden zwei grundlegende Begriffsverständnisse von Modellierungsmethode und -sprache sowie damit zusammenhängender Begriffe kurz skizziert. Dies ist zum einen grundlegende Begriffsverständnis nach KARAGIANNIES und zum anderen nach FRANK, wobei beide Wissenschaftler „ihr“ Begriffsverständnis zusammen mit „ihren“ jeweiligen Kollegen präzisiert und ergänzt haben. Ein von vielen weiteren Begriffsverständnissen von Modellierungsmethode und -sprache ist von GOLDKUHL ET AL. in [218] beschrieben. Die gedankliche **Gegenüberstellung unterschiedlicher Begriffsverständnisse**, die sich teils überlappen und teils widersprechen, schärft das Verständnis dieser relevanten Begriffe.

Die **Differenzierung zwischen Modellierungsmethode und -sprache** erfolgt in der Literatur nicht grundsätzlich. So verwenden z. B. die Autoren SCHMELZER & SESSELMANN die Begriffe Modellierungsmethode und -sprache und Notation synonym [473]. Auf die gleichbedeutende Verwendung dieser Begriffe weisen auch KARAGIANNIES & VISIC hin, die aber selbst zwischen diesen Begriffen wie nachfolgend erläutert klar differenzieren [292].

Grundlegendes Begriffsverständnis nach Karagiannis

Modellierungsmethoden bestehen nach KARAGIANNIS & KÜHN aus zwei Bestandteilen (vgl. Bild 123): Zum einen aus einer Modellierungstechnik, die wiederum in eine Modellierungssprache und ein Modellierungsverfahren unterteilt wird, und zum anderen aus Mechanismen und Algorithmen, die für Modellierungssprachen und in Modellierungsverfahren Verwendung finden [291] (vgl. auch [292], [293], [294]). Ein Modellierungsverfahren beschreibt wie die Konstrukte einer Modellierungssprache zu kombinieren sind, um ein gültiges Modell zu erstellen [290].

Zu einer **Modellierungssprache** gehören diejenigen Elemente, die zur Beschreibung eines Modells erforderlich sind. Konstrukte einer Modellierungssprache sind i. d. R. Entitäten, Beziehungen und Attribute [290]. Eine Modellierungssprache selbst wird durch ihre Syntax, Semantik und Notation beschrieben [291].

Dabei beschreibt die **Syntax** die Elemente und Regeln zur Erstellung von Modellen und wird durch eine Grammatik spezifiziert. In Bezug auf der Beschreibung der Syntax von Modellierungssprachen können zwei Ansätze voneinander unterschieden werden: graphische Grammatiken oder Metamodelle [291], [294].

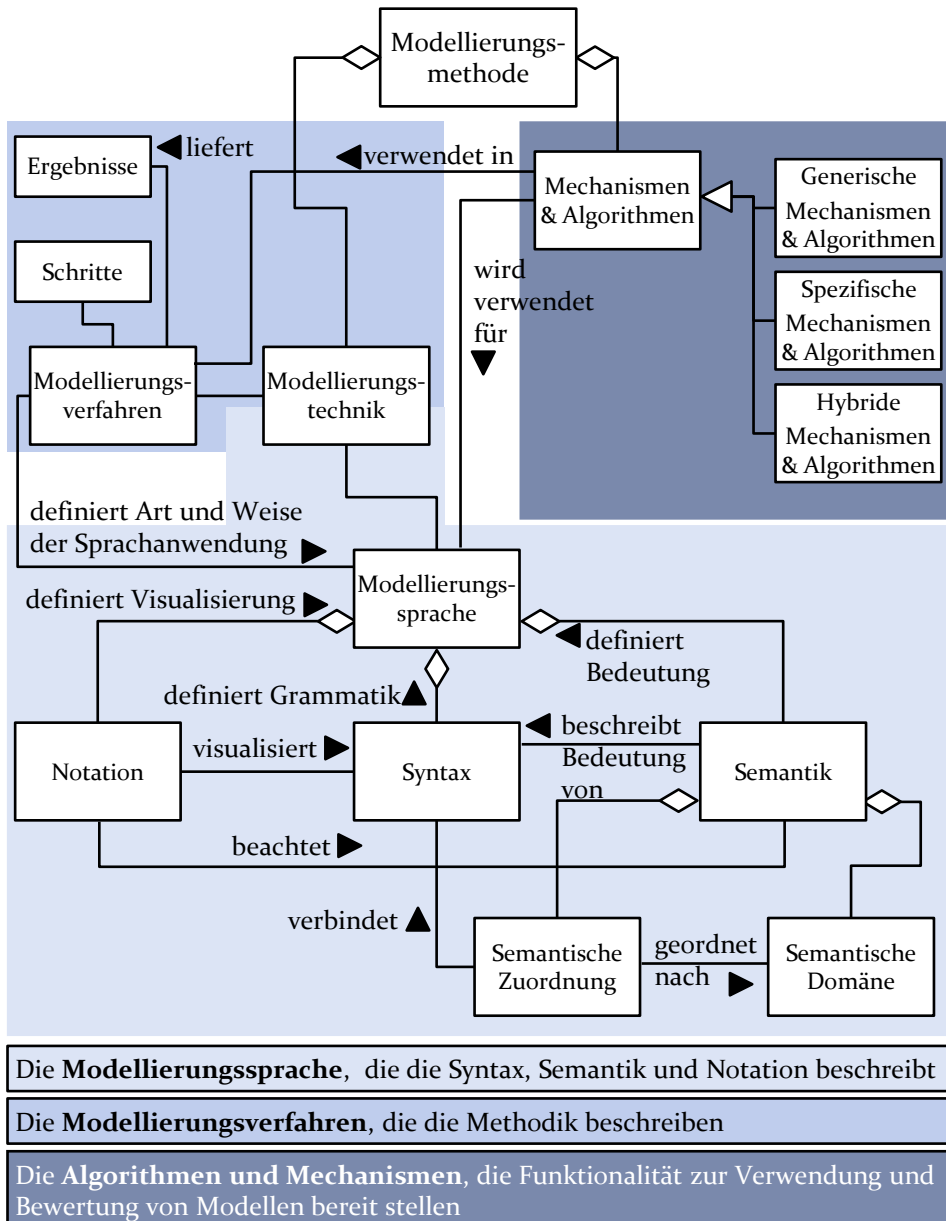


Bild 123: Bestandteile von Modellierungsmethoden und deren Zusammenhänge nach KARAGIANNIS & KÜHN (ergänzte und modifizierte Darstellung gemäß [291])

Die **Semantik** beschreibt die Bedeutung bzw. den Sinn einer Modellierungssprache und besteht aus einer semantischen Domäne und der semantischen Zuordnung. Dabei beschreibt die semantische Domäne die Bedeutung unter der Verwendung von mathematischen Zusammenhängen und

Ontologien. Eine Ontologie ist ein konzeptuelles Modell (spezifiziert in einer formalen Sprache), das Schlussfolgerungen erlaubt, verschiedene Abstraktionsebenen abdeckt und bei dem oft eine grafische Darstellung fehlt [187]. Die semantische Zuordnung verbindet die syntaktischen Konstrukte mit ihrer Bedeutung, die in der semantischen Domäne definiert wurden [291]. Weitere Ausführungen zu abstrakter Syntax und (statischer und dynamischer) Semantik für intensionale und extensionale Objekte sind [208] zu entnehmen.

Die **Notation** beschreibt schließlich die Visualisierung der Modellierungssprache. Dafür können nach [291] sowohl statische als auch dynamische Ansätze eingesetzt werden.

Die erforderlichen Schritte zur Anwendung einer Modellierungssprache werden mittels eines **Modellierungsverfahrens** beschrieben. Modellierungsverfahren zählen nach [609] zu den konstruktivistischen Verfahren, die entweder induktiv (von Beobachtungen ausgehend) oder deduktiv (z. B. von Theorien abgeleitet) vereinfachte Abbildungen der Realität erstellen. In einem Modellierungsverfahren werden generische, spezifische und hybride Mechanismen und Algorithmen eingesetzt, um Ergebnisse (z. B. Modelle) zu erstellen [291]. Die Mechanismen und Algorithmen bieten zudem Funktionalitäten zum Gebrauch und zur Bewertung von Modellen, die von Modellierungssprachen beschrieben werden [292].

Die Erläuterung von **Algorithmen und Mechanismen**, die Funktionalität zur Verwendung und Bewertung von Modellen, die durch eine Modellierungssprache beschrieben sind, bereitstellen, sind [292] zu finden. Weitere Erläuterungen zum Verständnis der einzelnen Begrifflichkeiten und v. a. deren Beziehungen sind insb. [291], [292] und [293] zu entnehmen.

Grundlegendes Begriffsverständnis nach Frank

Nach FRANK ist eine **Modellierungssprache** eine Sprache, die zur Erstellung einer Klasse von (konzeptuellen) Modellen dient [187] (vgl. Bild 125). Eine konzeptuelle Modellierungssprache wird durch ihre Syntax und Semantik definiert [187]. Dabei definiert die abstrakte Syntax Regeln für die Konstruktion syntaktisch korrekter Modelle. Die konkrete Syntax definiert die verwendeten Symbole, um die abstrakte Syntax zu repräsentieren. Da diese Symbole i. d. R. graphisch sind, wird diese oft als graphische Notation bezeichnet. Die Semantik einer Modellierungssprache, die auf einer formalen Spezifikation basieren kann, definiert die (formalen) Interpretation der

Modellierungskonzepte. Oft wird die Semantik einer Modellierungssprache jedoch nicht vollständig formalisiert [187].

Für die Erstellung eines konzeptuellen Modells ist – ähnlich wie bei KARAGIANNIS & KÜHN [291] – eine Modellierungssprache zwingend notwendig, aber nicht ausreichend [187]. Aufgrund der Komplexität der Modellierungsaufgaben ist i. d. R. ein professionelles Vorgehen bzw. eine geeignete Methode erforderlich. Eine Methode dient im Allgemeinen der Lösung einer Klasse von Problemen [187]. Sie besteht aus einer Terminologie, bewährten Annahmen über die erfolgreiche Aktion und einem entsprechenden Vorgehensmodell, das durch den Ablauf der Problemlösungsschritte führt [187]. Die Disziplin der systematischen Gestaltung von Modellierungsmethoden wird als Method Engineering bezeichnet [187]. Eine Modellierungsmethode ist eine bestimmte Art einer Methode. Diese dient der Lösung einer Klasse von Problemen durch die Entwicklung und Verwendung von Modellen. Eine **Modellierungsmethode** besteht aus mindestens einer Modellierungssprache und mindestens aus einem entsprechenden Modellierungsverfahren, das bei der Konstruktion und Analyse von Modellen unterstützt. Darüber hinaus enthält sie Annahmen über erfolgreiche Handlungsmuster. Eine Modellierungsmethode, die auf die Unterstützung der Gestaltung eines Metamodells oder die Konstruktion einer Modellierungssprache abzielt, wird als eine Metamodellierungsmethode bezeichnet: Eine Metamodellierungsmethode ist eine Methode, die aus einer Metamodellierungssprache und ein entsprechendes Vorgehensmodell besteht [187] (s. Anhang A.15).

Eine **domänenspezifische Modellierungsmethode** bedient sich mindestens einer DSML und mindestens eines domänenspezifischen Vorgehensmodells [187]. Eine Modellierungssprache wird als domänenspezifisch bezeichnet, falls die ihr zugrunde liegenden Konzepte weitestgehend aus der Fachsprache einer bestimmten Anwendungsdomäne stammen [39]. Um den spezifischen Sprachgebrauch der Vertreter der zu modellierenden Anwendungsdomäne verstehen und nachvollziehen zu können, ist eine persönliche Interaktion mit diesen und ein Hineinversetzen in deren Arbeitsalltag erforderlich. Dadurch lassen sich sprachliche Anomalien (z. B. Homonyme, Synonyme) identifizieren und auflösen [39].

A.15 Referenzmodelle und Metamodelle

In diesem Anhang wird zuerst das Referenzmodell erläutert. Darauf aufbauend erfolgt die **Erläuterung und Verortung von Metamodellen** bzw. Metametamodellen sowie eine jeweils zusammenfassende Beschreibung von drei Begriffsverständnissen von Metamodell aus Perspektive der Wirtschaftsinformatik nach drei bekannten Sichtweisen.

Referenzmodell

DSML unterstützen die Entwicklung von konzeptuellen Modellen durch domänenspezifische Konzepte für die Wiederverwendung. Eine ähnliche Zielsetzung wird mit der Entwicklung von Referenzmodellen verfolgt [185], die alle Kriterien einer Methode erfüllen [609]. So soll durch die **Entwicklung von Referenzmodellen** insb. die Konstruktion von anderen leistungsfähigeren Modellen (z. B. für bestimmte Domänen) unterstützt werden. Im Vergleich zu einem Referenzmodell bietet eine DSML erhöhte Flexibilität, erfordert aber immer noch Spezifikationen [185]. Um die Vorteile der Referenzmodelle (z. B. Zweckmäßigkeit, Wiederverwendungskomfort) und der DSML (z. B. Flexibilität) zu nutzen, können beide Ansätze miteinander kombiniert werden [185].

Die Referenzmodellierungsforschung ist ein Teilbereich der konzeptuellen Modellierungsforschung der Wirtschaftsinformatik und verfolgt das Ziel der Abbildung einer (zukünftig) geplanten oder optimierten Realität [609]. Mit einem Referenzmodell wird beabsichtigt, eine Domäne, die eine Klasse von möglichen Anwendungsfällen (z. B. alle Unternehmen einer bestimmten Branche) umfasst, abzubilden [183], [185]. Dabei besitzen **Referenzmodelle** einen deskriptiven als auch einen präskriptiven Anspruch: Zum einen sollen sie tatsächliche Handlungsmuster und entsprechende Konzepte berücksichtigen und zum anderen sollen sie als Vorlage für besonders effektive Informationssysteme dienen [183], [185]. Außerdem spezifizieren Referenzmodelle diverse Konzepte nicht isoliert (voneinander), sondern in einem Kontext, der die Semantik erheblich [183]. Als Beispiele für Referenzmodelle lassen sich Daten-, Objekt- und Geschäftsprozessmodelle sowie deren Integration mit Hilfe von Unternehmensmodellen anführen [183].

Da Forschungsergebnisse erst durch deren Implementierung in der Praxis ihren Nutzen entfalten können, kann die Fokussierung der Referenzmodellierungsforschung auf die Gestaltung möglicher Welten als wissenschaftstheoretisch überzeichnet angesehen werden [182]. Dabei ist aber zu bedenken, dass neben der technologischen Implementierung i. d. R. auch

wesentliche organisatorische und weitere Veränderungen (z. B. Handlungsmuster und -abläufe) notwendig sind. Dazu werden **mittels Referenzmodellen (zukünftige) Informationssysteme entworfen**, die aufgrund eines hohen Integrationsgrads und eines hohen Maßes an Wiederverwendung ein besseres Kosten-Nutzen-Verhältnis aufweisen und zudem qualitativ hochwertiger als gegenwärtige Systeme sind [419], [182]. Zudem erhält ein Unternehmen durch ein Referenzmodell eine Ausgangslösung für die Gestaltung der Prozesse, die eine Orientierung bzgl. des fachlichen Inhalts und des Detaillierungsgrads der Modellierung bietet [469]. Dabei wird aus einem Referenzmodell durch die Anpassung an die jeweiligen unternehmensspezifischen Anforderungen ein unternehmensbezogenes Modell [469]. Wie aus Erfahrungsberichten entnommen werden kann (vgl. insb. [419]), ist durch die Verwendung von Referenzmodellen in Organisationsprojekten eine Reduzierung von Zeit als auch von Kosten um mehr als 30 % möglich [469].

Neben den wissenschaftlichen Herausforderungen, die es bei der Entwicklung von Referenzmodellen zu bewältigen gilt, ist der Nutzen von Referenzmodellen von der Bereitschaft (z. B. Änderung bisherigen Sprachgebrauchs), diese also einzusetzen, abhängig [182]. Der mit der Entwicklung von Referenzmodellen verbundene Aufwand sowie das damit einhergehende wirtschaftliche Risiko sind weitere Herausforderungen der **Referenzmodellierungsforschung** [183]. Dies liegt u. a. daran, dass Referenzmodelle mehreren Tausend einzelne Modelle umfassen und somit sehr umfangreich sein können [469]. Hierzu sei auch auf die Erläuterungen von [419] zu Konzepten angewandter Analyse und Designmethoden und dabei insb. auf das Referenz-Metamodell für die Analyse und das System-Design verwiesen.

Grundlagen zu Metamodell und Metametamodell

Obwohl Metamodelle im Softwareengineering und der (Wirtschafts-)Informatik weit verbreitet sind, gibt es immer noch Debatten über die Konzeption von Metamodellen [187]. Der **Begriff des Metamodells** wird in der Literatur sehr heterogen definiert [533] und ist selbst selten im Fokus sprachphilosophischer Arbeiten [533]. Metamodelle v. a. in den methodologischen Teilgebieten der (Wirtschafts-)Informatik weit verbreitet [534]. Sofern Modelle und die Erstellung von Modellen selbst zum Gegenstand der Modellierung werden, wird der Begriff Metamodell verwendet.

Begriffsverständnis von Metamodell nach STRAHRINGER

Im Kontext der konzeptuellen Modellierung werden Modelle mit bestimmten Modellierungssprachen (z. B. DSML) erstellt. Aus diesem Grund kann nach [534] die Sprachstufentheorie auf die Modellbildung übertragen werden und infolgedessen lässt sich ein **Metamodell vereinfacht als ein Modell eines Modells** beschreiben. Dementsprechend handelt es sich bei der Abbildung der (Modellierungs-)Sprache, die zur Formulierung eines Modells verwendet wird, in einem Beschreibungsmodell um ein Metamodell; oder anders ausgedrückt: ein Metamodell liegt vor, wenn die Sprache, mit der ein Modell formuliert wird, in einem Beschreibungsmodell abgebildet wird (vgl. [534]). Wird die zur Erstellung des Metamodells verwendete Metasprache erneut in einem Modell abgebildet, liegt ein Metametamodell vor [534]. Demnach handelt es sich bei den in der Spracharchitektur jeweils übergeordneten Modellen um sprachliche Beschreibungsmodelle, die die Sprache, in der die jeweils entsprechenden untergeordneten Modelle formuliert sind, abbilden nach [534] (s. Bild 124). Somit kann dieselbe Modellierungssprache – je nach Ebene des zugehörigen Modells – in unterschiedlichen Rollen (z. B. als Objektsprache, Metasprache, Metametaspache) auftreten [534]. Auf Grundlage des sprachbasierten Metamodellbegriffs können Metamodelle grundsätzlich dadurch differenziert werden, in welcher Sprache sie selbst formuliert sind, sowie welche Eigenschaften (i. d. R. die abstrakte Syntax) der zu modellierenden Sprache sie abbilden [535].

Analog zur Modellierungssprache lässt sich ebenfalls der **Prozess der Modellbildung** auf Metamodellebene modellhaft abbilden [534]. Nach dem Metaisierungsprinzip kann auf der darüber liegenden Ebene der Prozess der Prozessmodellierung modelliert werden (prozessbasierter Metamodellbegriff, s. Bild 124). Dabei beschreibt ein Metaisierungsprinzip denjenigen Aspekt eines Modells, der in der darüber liegenden Modellierungsebene abgebildet wird. Durch die Verwendung eines Metaisierungsprinzips lässt sich der sprachbasierte Metamodellbegriff verallgemeinern [534].

Die beiden betrachteten Metaisierungsprinzipien – sprachbasierter und prozessbasierter Metamodellbegriff – können prinzipiell in einer **Modellhierarchie** kombiniert werden (vgl. Bild 124). Dabei ist zur Vermeidung konzeptioneller Widersprüche im Rahmen der Bildung der Ebenen die Offenlegung des zugrunde liegenden Metaisierungsprinzips dienlich [534]. Ein Beispiel für einen widersprüchlichen Metamodellbegriff in Folge der gleichzeitigen Verwendung verschiedener Metaisierungsprinzipien ist [534] zu entnehmen. Weitere Problembereiche bzw. -kreise im Kontext von

Meta-Begriffen werden in [535] erläutert und mit Hilfe von Beispielen aus der (Wirtschafts-)Informatik veranschaulicht.

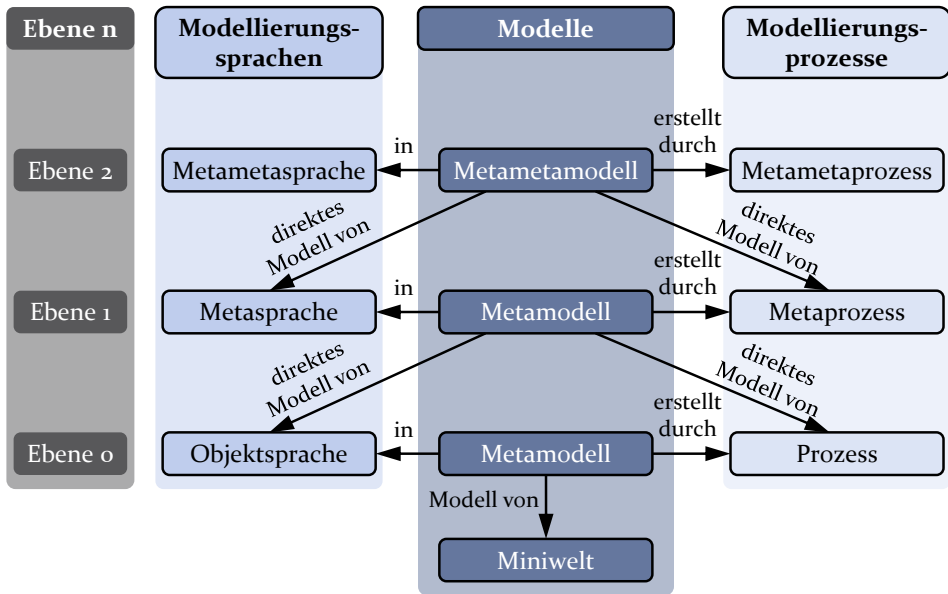


Bild 124: Sprachbasierter und prozessbasierter Metamodellbegriff (in Anlehnung an [534])

Begriffsverständnis von Metamodell nach Karagiannis

Metamodelle werden am häufigsten als „Modelle von (ähnlichen) Modellen“ definiert [290]. Um die zu veranschaulichen, wird i. d. R. eine graphische Darstellung – wie Bild 124 – verwendet, welche von [290] und ebenso von [291] zitiert wird. Dabei befindet sich auf der unteren Ebene 0 die zu modellierende Realität (d. h. die reale Welt) mit all ihren Objekten und Fakten. Dies wird mittels einer Modellierungssprache realisiert. Zur Abstrahierung von der Realität kann zum Beispiel die Technik der Entity-Relationship-Modellierung (ERM) verwendet werden. Die dafür verfügbaren Grundelemente der ERM (d. h. Entitäten, schwache Entitäten, Beziehungen mit verschiedenen Kardinalitäten) werden durch das Metamodell auf Ebene 2 mit Hilfe einer Metamodellierungssprache beschrieben. Diese Grundelemente der Modellierung werden durch eine andere Metaebene (Ebene 3) definiert, die als Metametaebene oder Meta²-Ebene bezeichnet wird und ein Meta²-Modell enthält und eine Metamodellierungssprache gebraucht. Somit wird hier der **Begriff Metamodell als Mittel zur Sprachdefinition** verwendet [290].

Demzufolge ist ein Metamodell ein Modell in der Art, dass sein Referenzmodell ein Metametamodell ist [338]. Ein **Metamodell ist i. w. S ein**

Modell einer Modellierungssprache und dieses muss die wesentlichen Merkmale und Eigenschaften der zu modellierenden Sprache erfassen [292]. Deshalb sollte ein Metamodell in der Lage sein, die Syntax, Semantik und Notation einer (Modellierungs-)Sprache zu beschreiben [292].

Ein **Metametamodell ist ein Modell, welches sein eigenes Referenzmodell ist** (d. h. es entspricht sich selbst) [338]. Es muss nach [293] alle üblichen Begriffe (u. a. Metamodell, Klassen, Beziehungen, Attribute, Modelltypen) zur Methodendefinition sowie zur -anwendung enthalten (vgl. [331]). Die Metamodellierung ist nach [292] der Schlüssel, der es ermöglicht, alle Modellierungssprachen in einer einheitlichen Art und Weise spezifizieren zu können (d. h. z. B. alle Metamodelle durch eine einzige Metametamodell zu beschreiben). Somit lassen sich die einzelnen Metamodelle durch Instanziierung des entsprechenden (übergeordneten) Metametamodells erzeugen (Top-down-Vorgehen) und umgekehrt lassen sich die einzelnen, miteinander in Beziehung stehenden Metamodelle durch Klassifizierung zum entsprechenden Metametamodell zusammenfassen bzw. vereinen [291].

Die traditionelle Sicht auf Metamodellierung umfasst nur eine von zwei wichtigen Dimensionen [290]. Diese berücksichtigt nicht explizit, dass es nicht nur eine linguistische (z. B. ein „Student“ ist eine Instanz einer gewissen „Entität“), sondern auch eine ontologische (z. B. da „Student“ auch eine „Person“ ist) **Instanziierung von Begriffen** existiert. In der linguistischen Perspektive der Metamodellierung befinden sich (der) „Student“ und (die) „Person“ auf derselben Ebene, während sich die „Person“ vom ontologischen Standpunkt aus auf einer übergeordneten Metaebene befindet. Demnach soll eine anspruchsvolle Metamodellierung beide Dimensionen gleichermaßen beachten [290].

Die Fähigkeit, Modellierungssprachen in einer einheitlichen Art und Weise zu beschreiben, zählt zu den **Vorteilen der Metamodellierung** [292]. Dadurch können Sprachen einheitlich gemanagt und bearbeitet werden und infolgedessen kann das Problem der Sprachenvielfalt besser bewältigt werden. So lässt sich bspw. die Zuordnung zwischen einer beliebigen Anzahl von bereitgestellten Sprachen konstruieren, da diese in derselben Metamodellierungssprache beschrieben sind. Mit Hilfe von Metamodellen können viele verschiedene Abstraktionen modelliert und (miteinander) kombiniert werden, um neue (Modellierungs-) Sprachen zu entwerfen, die speziell für eine bestimmte Anwendungsdomäne angepasst werden können. Als Ergebnis wird die Produktivität stark verbessert [292].

Begriffsverständnis von Metamodell nach Frank

Auch FRANK vertritt die Auffassung sich um eine differenzierte Terminologie (z. B. in Bezug auf die Unterscheidung von Sprachebenen einer Spracharchitektur) zu bemühen [186]. Ähnlich zu STRAHRINGER ist nach FRANK ein **Metamodell ein konzeptuelles Modell für eine Klasse ebendieser Modelle** [187]. Folglich legt ein Metamodell fest, wie zulässige Modelle einer bestimmten Klasse zu spezifizieren sind und definiert somit die abstrakte Syntax und Semantik einer Modellierungssprache [187]. Dabei kann die Syntax in eine abstrakte sowie eine konkrete Syntax (grafische Notation) aufgeteilt werden (vgl. auch Anhang A.14).

Das **semantische Netz** (vgl. Bild 125) zeigt die wesentlichen Begriffe mit den dazugehörigen bzw. entsprechenden Abstraktionsebenen und setzt diese zueinander in Beziehung [186]. Dabei entsprechen die zur Kennzeichnung der Ebenen verwendeten Ziffern den gebräuchlichen Konventionen (z. B. beginnend bei Ebene 0 bzw. M₀ für die Repräsentation der Instanzen). Die Spezifizierung eines Modells (Ebene 1 bzw. M₁) erfolgt durch eine Modellierungssprache, die wiederum teilweise durch ein Metamodell (Ebene 2 bzw. M₂) konkretisiert wird. Gleichzeitig ist ein Modell eine Instanz eines Metamodells, das wiederum eine Instanz eines Metametamodells (Ebene 3 bzw. M₃) ist [186]. Neben der Spezifizierung der Modellierungssprache wird durch ein Metamodell nur die abstrakte Syntax und Semantik spezifiziert (vgl. Bild 125). Metamodelle sind für diesen Zweck aus den folgenden beiden Gründen besonders geeignet [185]: Zum einen basieren sie auf demselben Paradigma wie die Modelle selbst und infolgedessen sollten potenzielle Sprachentwickler mit diesen vertrauter sein als mit Grammatiken. Zum anderen bieten sie eine bessere Grundlage für die Gestaltung von entsprechenden Modellierungswerkzeugen als Grammatiken, da sie auf direktem Wege zu Objektmodellen transformiert werden können.

Das **semantische Begriffsnetz** in Bild 125 ist vereinfacht dargestellt, da nicht nur die Modellierungssprache der zweiten Ebene (M₂), sondern auch alle anderen (Meta-)Modellierungssprachen eine entsprechende (spezifizierte) Semantik und Syntax enthalten. So spezifiziert ein Metametamodell eine abstrakte Syntax und Semantik einer Metamodellierungssprache und einen (erweiterbaren) Satz von Metamodellen, die der Spezifikation der DSML dienen und die wiederum durch eine Metamodellierungssprache festgelegt werden [187].

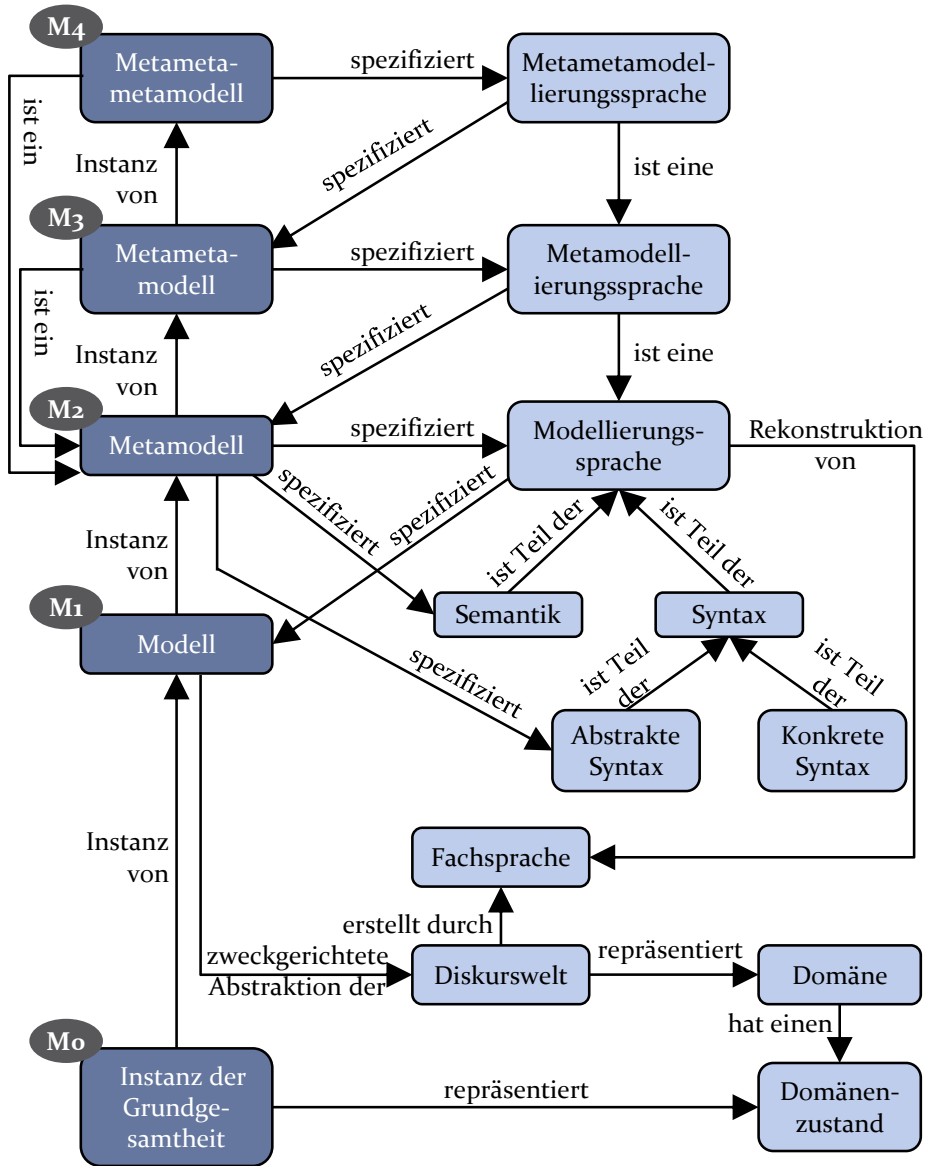


Bild 125: Semantisches Begriffsnetz entsprechend der Abstraktionsebenen (in Anlehnung an [186])

A.16 Überblick über domänenspezifische Modelle

In der wissenschaftlichen Literatur und auch in der industriellen Praxis gibt es viele **domänenspezifische Modelle und Modellierungsansätze**, die sich grundsätzlich zum Komplexitätsmanagement in der PE eignen. Dabei beziehen sich die meisten dieser Modellierungsansätze bzw. grundlegenden Modelle nicht nur auf eine, sondern auf mehrere Domänen. Demzufolge werden diese Ansätze – in Ergänzung zu den Ausführungen in Kapitel 3.1 – in diesem Anhang entsprechend, der in dieser Arbeit im Fokus stehenden Domänen kategorisiert. Somit werden die Ansätze, worunter i. w. S. auch Theorien, Methoden und Vorgehensweisen gezählt werden (vgl. [557]), bzgl. ihres jeweiligen Schwerpunkts in eine der sechs Domänen verortet. In den nachfolgenden sechs Abschnitten werden die Domänen in dieser Abfolge beleuchtet: Anforderungen, Produkt, Prozess, Wissen, Organisation und Kosten.

Anforderungsorientierte Modellierungsansätze

Im Folgenden werden einige, **ausgewählte Ansätze zur Modellierung von insb. Anforderungen** bzw. – allgemeiner gefasst – Spezifikationen überblicksartig vorgestellt. Für eine Übersicht diverser Spezifikationen (z. B. Anforderungen, DfX-Kriterien), deren Klassifikationsmöglichkeiten und einer Methodik zur Ableitung von Anforderungen aus dem Produktlebenszyklus sei auf [S31] verwiesen.

Konzept zur Anforderungsmodellierung (s. [527], s. auch [528]): Das Konzept zur Anforderungsmodellierung von STECHERT besteht aus acht Partialmodellen auf Grundlage von SysML, welches bei der ganzheitlichen Erfassung, Verarbeitung und Bereitstellung von Anforderungen unterstützt. Dabei soll mittels IT-Unterstützung eine zielgerichtete Analyse des Anforderungsmodells ermöglicht werden (z. B. systematische Identifikation von Zielkonflikten). Im Fokus steht ein Vorgehenskonzept für die Entwicklung von Baukastensystemen in der interdisziplinären, verteilten PE.

Produktionsseitiges Anforderungsmanagement (s. [384]): Auch dieser Ansatz nach MEIS dient zur zielgerichteten, kontextbezogenen Kommunikation zwischen allen Stakeholdern in den frühen Phasen der PE. Fokus des Konzepts ist insb. ein effizientes Management produktionsseitiger Anforderungen. Dabei wurde das Anforderungsmodell in Form einer Entity-Relationship-Datenbank mit dem Datenbankformat MS Access aufgebaut. Dadurch soll die Spezifikationserstellung unterstützt und die unternehmensfunktionsorientierte Kommunikation insb. zwischen PE und Produktionsplanung verbessert und somit v. a. Simultaneous Engineering

unterstützt werden. Dabei werden insb. auch strategische Aspekte berücksichtigt, die über eine produktionsgerechte Konstruktion hinausgehen.

Konzept zur Integration von Produktanforderungen (s. [283]): Ziel des Konzepts von JÖRG. ist die Integration von Produktanforderungen i. S. einer anforderungsorientierten PE. Die hierfür postulierte Methodik umfasst neben der Systematik zur Anforderungsmodellierung folgende Schwerpunkte: Integration der Anforderungsmodells in den PEP durch die Verknüpfung mit CAD-Modellen, sukzessive Einschränkung des Lösungsraums durch mathematische Constraints, Präzisierung von natürlichsprachlichen Anforderungen durch linguistische Methoden. Dabei ist insb. die Verbindung des Anforderungsmodells mit dem CAD-Modell für diese Arbeit von Interesse.

Rechnerunterstützte Verarbeitung von Anforderungen (s. [299]): Ziel der Arbeit von KICKERMANN ist die Verbesserung der Verfügbarkeit von Anforderungen in der PE durch Sicherstellung einer Konsistenz zwischen den Anforderungen und den entsprechenden Produkteigenschaften. Dazu wird eine computerunterstützte Formalisierung von Anforderungen für deren zielgerichtete Verarbeitung vorgestellt. Dabei wird der Bedeutungsumfang einer Anforderung bzgl. der drei Teilaspekte Semantik, Syntax und Pragmatik unterteilt, was auch für diese Arbeit relevant ist (vgl. Anhänge A.10, A.11 und A.12). Nach KICKERMANN bezieht sich die Semantik auf die Bedeutung einer Anforderung (z. B. worauf bezieht sich diese, welche Maße gibt sie vor). Die Syntax einer Anforderung gibt ihre formale (Aufbau-)Struktur vor und die Pragmatik einer Anforderung bezieht sich auf deren Anwendung.

Property based requirements (PBR) nach MICOUIN (s. [387]): Mit PBR (dt. eigenschaftsbasierte Anforderungen) wird das Ziel der constraintbasierten bzw. mathematischen Formalisierung von Anforderungen verfolgt. PBR stellt insb. eine Alternative zu textbasierten Anforderungen dar. Daher sind PBR ein Art constraints, die auf eine Eigenschaft/Property „P“ eines Systems angewandt wird. Ein constraint bedingt, dass die Werte von „P“ innerhalb des Wertebereichs „D“ liegen, der eine echte Teilmenge der für „P“ möglichen Werte ist. Daher wird ein PBR formal mittels mathematischer (v. a. boolscher) Ausdrücke aus der Mengenlehre definiert (z. B. $PBR:[when\ C] \rightarrow val(O.P) \in D \subset Im(P)$). Dieser Ansatz zur Formalisierung von Anforderungen wird auch in dieser Arbeit verwendet, da dadurch das Anforderungsmanagement computerunterstützt werden kann.

Requirements Engineering (RE) z. B. nach POHL (s. [436]): Der Fokus von RE als Teil des Software Engineering liegt zwar v. a. auf der

Softwareentwicklung, lässt sich aber u. a. nach POHL auch auf die Anforderungsmodellierung für technische Systeme übertragen. Im RE-Prozess wird das zu entwickelnde softwareintensive bzw. technische System bzgl. der vier Facetten des Systemkontexts untersucht: Die Gegenstandsfacette umfasst die durch das System abgebildeten, abstrakten oder immateriellen Gegenstände und Ereignisse (z. B. Gesetze, Standards); die Nutzungsfacette enthält die Aspekte der Systemverwendung; die IT-Systemfacette umfasst das dem System übergeordnete IT-System; die Entwicklungsfacette umfasst Aspekte des PEPs sowie Entwicklungswerkzeuge. Dabei unterstützt der RE-Prozess mit folgenden drei Kernaktivitäten: Dokumentation der Anforderungen zusammen mit weiteren wichtigen Informationen (z. B. Entscheidungen, Begründungen); Gewinnung der Anforderungen aus diversen Quellen mit einem ausreichend tiefen Verständnis dieser; Übereinstimmung der Anforderungen in Folge deren konfliktfreier Formulierung in Abstimmung mit den Stakeholdern. Da i. S. v. einem holistischen Komplexitätsmanagement alle drei genannten Kernaktivitäten sowie v. a. die Gegenstands- und Entwicklungsfacette wichtig sind, stellt auch das RE einige hilfreiche Aspekte für diese Arbeit bereit.

Anforderungsdiagramm in SysML (s. [14], s. auch [156], [132], [193]): Auch wenn in vielen Unternehmen noch immer die Dokumentation der Anforderungen bzw. Spezifikationen (insb. Lasten- und Pflichtenhefte) i. d. R. in Form von Word- und Excel-Dokumenten erfolgt, wird in der Softwareentwicklung häufig die Modellierungssprache UML zur Modellierung von z. B. Use-Cases verwendet. Besonders für die Systementwicklung bietet sich auch SysML an (s. Kapitel 3.2.1). Die mittels Modellierungssprachen erstellten Anforderungsmodelle haben vielfältige Vorteile, wie bspw. Reduzierung des Aufwands bei der Erstellung und insb. Änderung von Anforderungsdokumenten bzw. -modellen aufgrund der strukturierten, bildhaften Darstellung (z. B. von Relationen zwischen Anforderungen) im Vergleich zu textuellen Formulierungen. Zudem hat eine durch Computer interpretierbare Modellierungssprache – wie SysML – weitere Vorteile, wie z. B. deren Vernetzung und Integration in andere computerunterstützte Modelle (vgl. [S33]). Ein beispielhaftes Anforderungsdiagramm auf Basis von SysML ist in Bild 126 dargestellt. Die beiden wesentlichen Bestandteile des Anforderungsdiagramms sind Anforderungsobjekte (Rechtecke), welche mit einem Namen bezeichnet und mit Attributen genauer beschrieben werden, und Relationen, die Anforderungsobjekte übergeordneten Gruppen von Anforderungen zuordnen.

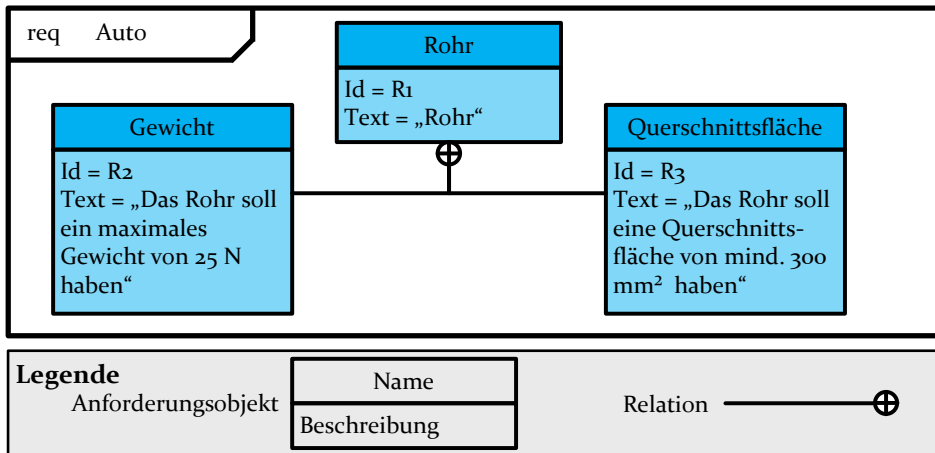


Bild 126: Ausschnitt aus einem Anforderungsdiagramm für ein Rohr

Produktorientierte Modellierungsansätze

Universal Design Theory (UDT) (s. [223], s. auch [359], [358]): Die UDT ist eine Theorie, die Erkenntnisse über die Entwicklung aus verschiedenen wissenschaftlichen Disziplinen in einer konsistenten, kohärenten und kompakten Form formuliert. Die UDT dient insb. als wissenschaftliche Grundlage zur Rationalisierung der interdisziplinären PE. Dabei vereinfacht eine auf v. a. folgenden Axiomen aufbauende, einheitliche „Modellierungssprache“ die Zusammenarbeit zwischen Entwicklern aus verschiedenen Domänen: Zerlegung von komplexen Problemen in Teilprobleme; Existenz von Wissen in den einzelnen Domänen für konkrete Problemlösungen; domänenspezifisches Wissen kann in anderen Domänen zur Lösungsfindung beitragen. Diese drei Axiome spielen auch in dieser Arbeit eine wesentliche Rolle.

Characteristics-Properties Modelling (CPM) (s. [590], s. auch [593], [594], [592]): Gemäß dem CPM nach WEBER werden durch die Festlegung geeigneter Merkmale (engl. characteristics) durch Entwickler bestimmte Eigenschaften (engl. properties) des Produktsystems realisiert. Durch das CPM wird die gezielte Änderung der die Eigenschaften beeinflussender Merkmale unterstützt. Dabei sind die Relationen zwischen den Elementen im Rahmen der Analyse und Synthese von Rand- und Modellierungsbedingungen abhängig. Dieser Modellierungsansatz ist aufgrund der expliziten Berücksichtigung diverser Elemente und auch Bedingungen für diese Arbeit grundlegend. Auch weitere Wissenschaftler bauen auf den CPM-Ansatz auf, wie z. B. KREHMER für sein Vorgehensmodell in der eigenschaftsbasierte PE [323]. Zudem baut auch die Konstruktionsmethodik für

Hybride Intelligente Konstruktionselemente (HIKE) von CROSTACK auf dem CPM-Ansatz auf [115]. Seine Arbeit zu Grundlagen einer Konstruktionsmethodik für HIKE ist im Rahmen der DFG-Forschergruppe 981 HIKE entstanden (vgl. [116]). Im Fokus seiner Arbeit steht die Definition und Modellierung des Referenzprozesses (vgl. auch Anhang A.15) mit Methoden für das Konstruieren von HIKE-Systemen. Dabei wird mit Hilfe von Gestaltungsrichtlinien und einer Informations-/Wissensbasis die Modellierung der HIKE und ihrer zugehörigen Konstruktionsprozesse unterstützt [115]. Eine Erweiterung des CPM zusammen mit einem Anwendungsbeispiel u. a. vom Autor dieser Arbeit ist in [P2] zu finden.

Contact and Cannel-Ansatz (C&C-Ansatz) (s. [10] und für Erweiterungen u. a. [7], [9], [372], [630]): Der Fokus des C&C-Ansatzes steht v. a. die Modellierung, Analyse und Visualisierung der Relationen zwischen Systemfunktionen (i. S. v. Kundenwunsch) und -gestalt (i. S. v. Ergebnis der PE). Dazu werden komplexe Systeme mit folgenden Basiselemente, aus welchen jedes beliebige System abgebildet werden kann, modelliert: Wirkflächen (WFs), Wirkflächenpaare (WFPs), Leitstützstrukturen (LSSs) und Connectoren (C). Im C&C-A wird zwischen Betrachtungsraum mit relevanten Elementen für die Funktionsbetrachtung (z. B. C, WF) und Designraum mit von Entwicklern festlegbaren Elementen (z. B. LSS, WF) differenziert. Während in der Synthese alle gewünschten Systemfunktionen definiert werden, werden bei der Analyse in den sog. Wirknetzen die Wechselwirkungen während der Funktionserfüllung untersucht. Dabei sind nur die stringente Trennung zwischen Synthese und Analyse und das Systemdenken für diese Arbeit hilfreich, sondern auch das detaillierte Konzept der von Wirkflächen und -paaren sowie Leitstützstrukturen und Connectoren.

Prozessorientierte Modellierungsansätze

Handlungsorientierte Ansätze

Property-Driven Development/Design (PDD) (s. [590], s. auch [593], [594], [592]) Ein Teil des bereits in Kapitel 3.1.1 erwähnten CPM/PDD-Ansatzes stellt der PDD-Teil dar, welcher sich im Gegensatz zum CPM mit der Produktsicht auf die Prozesssicht fokussiert (insb. die Modellierung iterativer Entwicklungsprozesse). Dabei wird im PEP kontinuierlich zwischen Analyse und Synthese i. S. eines Regelkreises gewechselt. Die Kernzielsetzung ist in einem erneuten Syntheseschritt neue/bekannte Merkmale anhand einer Relation (z. B. FE-Simulation anstatt Überschlagrechnung) genauer bestimmen zu können. Anschließend werden in der Analyse neue/bekannte Eigenschaften genauer bestimmt. Dieses Vorgehen wird so lange durchgeführt, bis alle relevanten Merkmale bekannt sowie die

wichtigen Ist-Eigenschaften mit einer hinreichenden Genauigkeit realisiert sind und den Soll-Eigenschaften entsprechen.

Function-Behaviour-Structure-(FBS)-Framework (s. [213], s. auch [212] und die FBS-Ontologie in [214]): Das FBS-Framework geht auf die Arbeit von GERO zurück. Mit Function wird der Zweck des Objekts und somit die Anforderungen an das Objekt beschrieben. Structure umfasst alle Objektbestandteilen und dessen Relationen. Behaviour beschreibt aus Synthesicht das Soll-Verhalten (Ableitung aus Function) und aus Analysesicht das Ist-Verhalten (Ableitung aus Structure). Im PEP ordnet der Entwickler aufgrund seiner Erfahrung die Funktion dem Verhalten zu und leitet Verhalten aus der Struktur des Objekts ab. Das FBS-Framework unterstützt die Darstellung des Objekts auf Aggregationsebenen sowie mittels mehrerer Modellvarianten zur Darstellung verschiedener Ziele. Das Verknüpfen von Anforderungen auf die jeweils dafür erforderlichen Elemente, das Soll-Verhalten (Synthese) und das Ist-Verhalten (Analyse) sind nur einige Aspekte des FBS-Frameworks die auch in dieser Arbeit (modifiziert) verwendet werden.

Axiomatic Design (s. [541], s. a. [539], [540]): Beim Axiomatic Design, welches v. a. auf den Arbeiten von SUH beruht, wird durch ein Mapping (i. S. v. Festlegen) von Elementen durch ein iteratives Vorgehen zwischen den Kunden-, Funktions-, Physik- und Prozessdomäne die Anforderungen schrittweise konkretisiert und letztlich erfüllt. Dabei wird das Ziel einer mathematischen Darstellung des Entwurfs in einer Design Matrix unter Beachtung folgender Axiome verfolgt. Das Independence Axiom postuliert, dass alle funktionalen Anforderungen unabhängig voneinander formuliert und erfüllbar sein müssen. Das Information Axiom besagt, dass der Informationsgehalt der Entwicklung konsequent reduziert werden muss, ohne dabei Informationen wegfällen zu lassen. Das Komplexitätsmanagement erfolgt dabei durch das Herunterbrechen der Entwicklung auf einzelne einfachere Systeme erreicht (s. [595] für Beispiele aus der Automobilindustrie). Ein ideales System i. S. v. Axiomatic Design besitzt keine Relationen zu anderen Systemen (vgl. Modularisierung). Insbesondere die mathematische Formalisierung des zu entwickelnden Systems als auch die genannten Axiome sind wichtige Aspekte des in dieser Arbeit vorgestellten Konzepts zum Komplexitätsmanagement in der PE.

Phasen- und zustandsorientierte Ansätze

VDI-Richtlinie 2221 (s. [580] und [581]): Die VDI-Richtlinie 2221 Blatt 1 stellt ein allgemeines Modell der PE vor, welches sich in Begleit- (z. B. Änderungs- und Kostenmanagement) sowie Querschnittsaktivitäten (z. B.

Prozess- und Wissensmanagement) einbettet. Das Modell der PE selbst wird in Ziele (v. a. Anforderungen), Aktivitäten (z. B. Klären bzw. Präzisieren der Problem-/Aufgabenstellung), Phasen (z. B. Planung, Konzept) und Ergebnissen (z. B. Anforderungs-, Funktionsmodelle) gegliedert. Wie aus diesem allgemeinen Modell ein konkretes Vorgehen für ein reales Projekt werden kann, wird in der VDI 2221 Blatt 2 dargelegt. Darin wird aufbauend auf der Modellierung des PEPs und Kontextfaktoren in der PE die Gestaltung spezifischer PEP aufgezeigt (z. B. für diverse Branchen).

VDI-Richtlinie 2206 (V-Modell) (s. [575], s. a. [174], für eine Weiterentwicklung hin zu einem SE-Ansatz [157]): Diese Richtlinie stellt eine Entwicklungsmethodik für mechatronische Systeme bereit. Ziele des V-Modells sind die Unterstützung notwendiger Iterationen sowie die Absicherung von Eigenschaften (i. S. v. einem ständigen Abgleich der Anforderungen mit den entwickelten Eigenschaften). Ausgehend von den Anforderungen gliedert sich das V-Modell in einen domänenübergreifenden Systementwurf, in einige domänenspezifische Entwürfe (z. B. für Mechanik, Elektrotechnik) und in eine domänenübergreifende Systemintegration. Auch dieser Arbeit liegt die Auffassung zugrunde, dass domänenspezifische Modelle für konkrete Aufgaben- und Problemstellungen zu verwenden sind und diese auf einer höheren Abstraktionsebene in domänenübergreifende Modelle zu integrieren sind.

FORFLOW-Prozessmodell. (s. [380], s. auch [324], [325]): Das im Rahmen des Bayerischen Forschungsverbunds „FORFLOW für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung“ entwickelte Prozessmodell hat das Ziel alle im PEP Beteiligten zu unterstützen und eine kontextspezifische Navigation durch den PEP zu ermöglichen. Dafür stellt das Prozessmodell drei Ebenen zur Verfügung, wobei die Schritte der ersten zwei Ebenen wie vorgegeben zu durchlaufen sind. Die Abfolge der Schritte der dritten Ebene ist beliebig. Zudem wird zwischen zwingend auszuführenden, optionalen und nur bei Neuentwicklung durchzuführenden Schritte differenziert. Dadurch wird eine spezifische Prozesskonfiguration ermöglicht sowie bei der Integration von DfX, geeigneten Methoden, CAx-Werkzeugen sowie wie beim Iterations- und Produktreifegradmanagement unterstützt. Der hierarchische Aufbau sowie die Unterstützung bei der Konfiguration von Prozessen sind für diese Arbeit relevant.

Mikrozyklen in der PE

Problemlösungsprozess (in der VDI-Richtlinie 2221) (s. [580]): Gemäß der VDI-Richtlinie 2221 bestehen die Aktivitäten eines typischen

Problemlösungsprozesses vom Problem zur Lösung aus drei iterativen Schritten mit je zwei Teilschritten: Zielsuche mit Situationsanalyse und Zielformulierung, Lösungssuche mit Synthese und Analyse von Lösungen sowie der Lösungsauswahl mit Bewertung und Entscheidung. Diese drei grundlegenden Schritte dienen auch dem in dieser Arbeit vorgestellten Konzept als wesentliche Grundlage, da erstens das Ziel der matrixbasierten Modellierung klar sein muss, zweitens die Lösungssuche bestehend aus Synthese und Analyse zu unterstützen ist und drittens Hilfestellung bei der Bewertung zur Lösungsauswahl gegeben werden muss.

Test-Operate-Test-Exit-Schema (TOTE-Schema) (vgl. [151]): Das TOTE-Schema hat als eines der grundlegenden Mikrozyklen in der PE das Ziel bei der Problemlösung zu unterstützen. Dabei helfen die vier zyklusförmig angeordneten Schritte: Der Operate-Schritt stellt eine reale oder abstrakte Handlung oder das Aufstellen einer Hypothese dar (d. h. synthetischer Schritt). Im Test-Schritt wird die Kontrolle des vorigen Schrittes bzgl. Zieleigenschaftserreichung ausgeführt (d. h. analytischer Schritt). Sofern der Zielwert erreicht ist, kommt der Exit-Schritt, andern-falls wieder der Operate-Schritt. Prinzipiell bauen alle Problemlösungs-prozesse auf dem TOTE-Schema auf und bieten i. d. R. auch mehr unterstützende Handlungsschritte für die PE an. Dieser Regelkreis wird dem Konzept zum Komplexitätsmanagement in der PE von komplexen Systemen zugrunde gelegt, da die Auswirkungen von Modifikationen von Elementen und/oder Relationen auch immer zu testen und evaluieren sind, um darauf aufbauend schrittweise verbessert werden zu können.

Plan-Do-Check-Act-Zyklus (PDCA-Zyklus) (vgl. [355]): Der PDCA-Zyklus unterstützt beim Problemlösen mit dem Fokus auf einen prinzipiellen Prozess für kontinuierliche Verbesserungen. Dazu beschreibt der PDCA-Zyklus die Problemlösung als Kreislauf aus den Schritten Plan (Problem-analyse und Maßnahmenbeschreibung), Do (Maßnahmenrealisierung in gewissen Umfang), Check (Überprüfung der Maßnahmenwirksamkeit) sowie Act (Maßnahmenrealisierung in großen Umfang). Aufgrund der ähnlichen Argumente wie zum TOTE-Schema ist auch der PDCA-Zyklus für das in dieser Arbeit vorgestellte Konzept zum holistischen Komplexitätsmanagement in der PE relevant.

Weitere, prozessorientierte Modellierungsansätze

Denk- und Handlungsrahmen für eine prozessorientierte PE (s. [44]): In der sog. Konzeption eines Denk- und Handlungsrahmens für eine prozessorientierte PE unter Nutzung der Systemtechnik nach BENEKE werden viele Lösungsansätze zu einem Konzept für ein Prozessmodell integriert.

Ausgehend von Anforderungen und Entwicklungsrandbedingungen sind u. a. folgende Teilkonzepte enthalten: Prozessnavigation und -schnittstellen sowie Methoden- und Werkzeugintegration. Ein wesentlicher Beitrag dieser Dissertation ist es, dass darin jeweils einige strukturelle Wechselwirkungen zwischen den Domänen Produkt, Prozess und Organisation hinsichtlich deren Auswirkungen auf den PE thematisiert werden. Daher sind daran auch für diese Arbeit interessante, konzeptuelle Aspekte für eine prozessorientierte PE zu finden.

Prozessorientierter Entwurf intelligenter technischer Systeme (s. [599]): Der Ansatz zum prozessorientierten Entwurf intelligenter technischer Systeme von WEISSHAUPT verfolgt das Ziel mittels einer interdisziplinären Methode die Entwickler beim Übergang aus dem Lösungsraum in den Prozessraum zu unterstützen. Während im Lösungsraum bereits definierte technische Lösungen für die Realisierungen funktionaler Anforderungen verfügbar sind, besteht im Prozessraum (noch) die Möglichkeit zur lösungsneutralen Spezifikation von Anforderungen eines Prozesses. Der Arbeit von WEISSHAUPT liegt ein ähnliches Verständnis wie in dieser Arbeit zugrunde, nämlich das Verständnis bzw. Ziel erst möglichst spät im PEP von einer lösungsneutralen Denkweise in das Denken in konkreten technischen Lösungen zu „wechseln“. Zudem wird in dieser Arbeit auch einer objektorientierten Modellierung (z. B. formale Darstellung des Entwurfs gewählter Lösungsprinzipien) als gemeinsame Sprache aller im PEP Beteiligten gefolgt.

Wissensorientierte Modellierungsansätze

Nachfolgend wird ein Überblick über **ausgewählte Ansätze der Wissensmodellierung in der PE** gegeben. Zur Modellierung von Wissen, Informationen und Daten gibt eine Vielzahl an Ansätzen in der Literatur. Dies liegt insb. an den vielfältigen charakteristischen Eigenschaften von Wissens- und Informationselementen und deren Relationen (s. eigene Arbeiten [P6], [P8], [P9]) und mit Fokus auf Rolle und Aufgaben des Wissensingenieurs (s. eigene Arbeiten [P7], [P24], [P28]) sowie auch die mitentwickelten Methode zur Einbindung von heuristischem Wissen in die erweiterte matrixbasierte Produktbeschreibung in [P35] und Die Arbeit zur Analyse und Bewertung von Wissen in der Produktentwicklung von ROTH [463]. Aufgrund der Eigenschaften von Wissen und Informationen eignen sich v. a. auch die drei im folgenden Kapitel 3.2 ausführlicher erläuterten domänenübergreifenden Modellierungsansätze: Systems Engineering, Ontologie (vgl. auch Bild 130 und Bild 131) und Matrizen. Da u. a. nach [579] und [313] CommonKADS und MOKA die relevantesten Methodiken bei der Unterstützung der

Wissensmodellierung in der PE (v. a. für die wissensbasierte Konstruktion sowie auch des Knowledge Engineerings) sind, werden zuerst diese beiden Ansätze kurz vorgestellt. Anschließend erfolgt noch ein Überblick über allgemeine Ansätze zur Wissensmodellierung.

Common-Knowledge Acquisition and Documentation Structuring (Common-KADS) (s. [483], s. auch [396] und die unter Mitwirkung des Autors dieser Arbeit entstandene VDI-Richtlinie 5610 [579]): Common-KADS ist eine Methodik mit dem Ziel die Entwicklung von z. B. Expertensysteme, wissensbasierte Systeme und KBE-Systeme zu unterstützen. CommonKADS setzt sich aus sechs Modellen (Organisations-, Aufgaben-, Agenten-, Wissens-, Kommunikations- und Designmodell) zusammen, welche in die Kontext-, Konzept- und Artefakt-Ebene unterteilt werden. Das Wissensmodell ist dabei das zentrale Modell und definiert die erforderlichen Wissensarten sowie Anforderungen an das zu entwickelnde System. Dieses bildet auch gemeinsam mit dem Kommunikationsmodell, welches das User Interface (UI) definiert, die Eingabe für das Systemdesign und die Implementation. Das Wissensmodell ist in die drei Ebenen Domänen-, Folgerungs- und Aufgabenwissen unterteilt. Die Common-KADS Methodik ist für diese Arbeit hilfreich, da einerseits auch in dieser Arbeit ein IT-System zum Komplexitätsmanagement erarbeitet wird und andererseits dafür auch Wissen und Informationen zu diversen Modellen zu akquirieren und geeignet zu strukturieren ist. Dabei ist jeweils auch die Common-KADS Methodik hilfreich.

Methodology and tools Oriented to Knowledge-based Applications (MOKA) (s. [417], s. auch [448], [80]): Die MOKA-Methodik verfolgt das Ziel mittels einer strukturierten Vorgehensweise die Akquisition und Repräsentation von Produkt- und Prozesswissen zu unterstützen. Dafür stellt MOKA die aufeinander aufbauenden Stufen Identifizieren, Legitimieren, Erfassen, Formalisieren und Aktivieren bereit. MOKA unterstützt auch bei der Verknüpfung des unverarbeiteten Wissens mit einer KBE-Anwendung. Zudem hilft MOKA Wissen sowohl zu zerlegen als auch zu erweitern, um dieses effizient und effektiv in ein Wissensnetz (z. B. eine Ontologie) zu integrieren. Dazu dient ein Rahmenwerk, welches zur Erfassung und Repräsentation von Wissen auf der formalen und informalen Ebene dient. Bei der Transformation von Wissen von der informalen auf die formale Ebene unterstützt die auf UML basierende MOKA Modelling Language. Die MOKA-Methodik enthält ähnlich wie bei Common-KADS hilfreiche Aspekte für diese Arbeit. Dazu zählen u. a. das Formalisieren und Zerlegen von Wissensselemente (inkl. Informaler Elemente) und deren Integration in ein Wissensnetz (s. Bild 131 in Anhang A.17).

Wissensmodellierung im Schritt-für-Schritt-Vorgehen nach MILTON (s. [388] und für Details [162], [163], [61], [313]): Die Wissensmodellierungsansätze im Schritt-für-Schritt-Vorgehen zur Wissensakquisition dienen insb. zur Erstellung und Nutzung von Wissensmodellen. In der Fachliteratur wird bzgl. der Wissensakquisition zwischen den drei Grundformen der indirekten, direkten und automatischen Wissensakquisition differenziert (s. u. a. [337]). Die Wissensmodellierungsansätze unterstützen das gewonnene Wissen zu explizieren, zu validieren und in den Kontext einzuordnen sowie auch das Identifizieren von Wissensträgern oder -quellen. Zu den Wissensmodellierungsansätzen nach MILTON zählen neben Matrizen (s. Kapitel 3.2.3) u. a. auch Baumdiagramme und Wissenskarten. Baumdiagramme dienen zur Abbildung der hierarchischen Anordnung von Wissen, wobei jeder Knoten des Baums ein Wissensselement der Wissensbasis darstellt und jede Kante eine Relation zwischen diesen Elementen. Im Gegensatz dazu müssen bei Wissenskarten die Wissensselemente als Knoten und ihre Relationen als Kanten nicht hierarchisch angeordnet sein. Das Ziel einer Wissenskarte ist Transparenz durch eine Visualisierung der Wissensstruktur zu realisieren sowie das Auffinden von Wissen zu unterstützen. Weitere Ansätze der Wissensmodellierung sind in der Diplomarbeit des Autors [P6] und in betreuten studentischen Arbeiten wie u. a. in [S15], [S16] und [S60] zu finden.

Organisationsorientierte Modellierungsansätze

BLEICHER hat in seiner umfangreichen Schriftenreihe bereits 1972 auch die Organisation als System aufgefasst und ausgehend von der Systemtheorie (vgl. Kapitel 2.1.1) die Organisation beleuchtet. Im Folgenden stehen die **Ansätze der Organisationsmodellierung** im Fokus. Dazu werden in Anlehnung an SCHREYÖGG & GEIGER zuerst klassische Organisationsmodelle der Arbeitsteilung durch organisatorische Differenzierung und danach Abstimmungsverfahren im Rahmen der organisatorischen Integration skizziert [486]. Abschließend werden moderne Organisationsmodelle beschrieben. Da die PE als Organisation grundsätzlich in allen Organisationsmodellen zu finden ist und zudem i. d. R. mit zahlreichen anderen Organisationseinheiten eine Vielzahl an Relationen hat, sind alle Organisationsmodelle für diese Arbeit von Bedeutung. Eine Verortung von Organisationsmodellen bzgl. der beiden Dimensionen Marktunsicherheit (allgemeiner aufgefasst externe Komplexität bzw. VUCA-Welt) und Produktkomplexität (allgemeiner aufgefasst auch interne Komplexität) ist folgendes Bild 127 dargestellt.

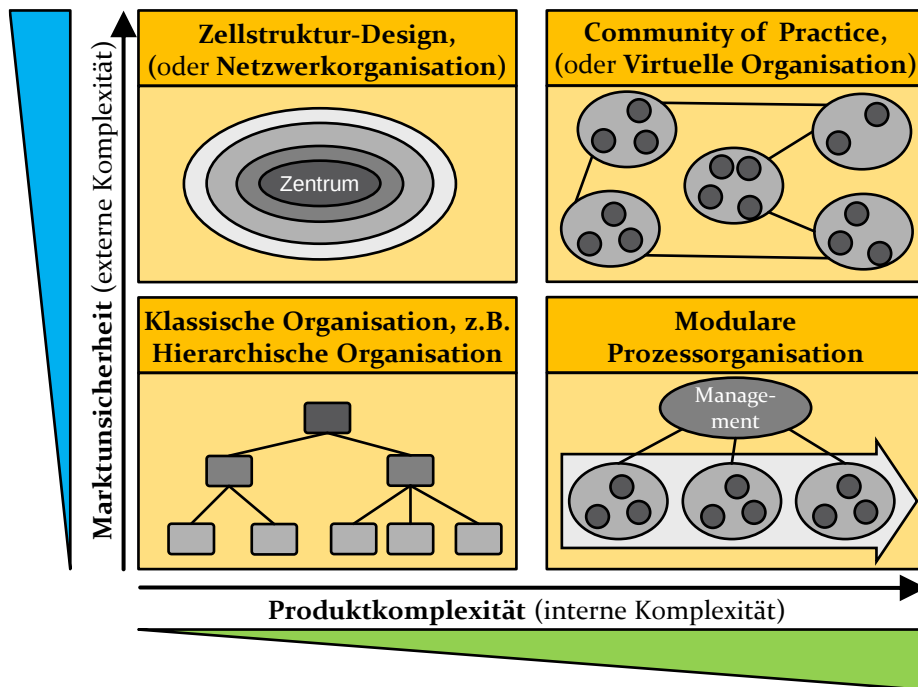


Bild 127: Verortung von Organisationsmodellen (in grober Anlehnung an [438])

Organisatorische Differenzierung

Organisation nach Verrichtung (s. [486], s. auch [428]): Eine verbreitete Form organisatorischer Arbeitsteilung ist die Spezialisierung nach Verrichtungen bzw. Funktionen, d. h. Zusammenfassung von Stellen mit gleichartigen Verrichtungen zu Funktionsbereichen (z. B. Entwicklung, Produktion). Das Ziel funktionaler Organisationsmodelle ist Kostenreduzierung durch effiziente Ressourcenausnutzung (z. B. durch Synergie- und Lerneffekten). Dabei ist jeder Funktionsbereich von anderen Funktionsbereichen abhängig (z. B. Entwickler benötigen Feedback aus dem Vertrieb zur optimalen Erfüllung der Kundenbedürfnisse).

Organisation nach Objekten (s. [486], s. auch [428]): Die Organisation nach Objekten geht mit der Spezialisierung nach Objekten i. S. v. z. B. Kunden, Märkte, Produkte einher. In diesen objektorientierten Geschäftsbereichen (oft als Divisionen bezeichnet) werden verschiedene Verrichtungen aus unterschiedlichen Funktionsbereichen gebündelt. Mit divisionalen Organisationsmodellen werden v. a. schnellere Entscheidungen sowie flexibleres Handeln unterstützt.

Organisatorische Integration

Abstimmung durch Hierarchie (s. [486]): Als klassisches Integrations- und Koordinationswerkzeug dienen diverse hierarchische Organisationsmodelle, wie das Einlinien- und Mehrliniensystem, wobei das Einliniensystem durch Integration Stabstellen zum Stabliniensystem erweitert werden kann. Während beim Einliniensystem eine Stelle nur einer einzigen weisungsbefugten Instanz zugeordnet ist, haben beim Mehrliniensystem mindestens zwei spezialisierte Instanzen Weisungsbefugnisse.

Abstimmung durch Programme (s. [486]): Eine zusätzliche oder alternative Integrationsform sind unternehmensweite und funktionsübergreifende Programme, wie z. B. Konditionalprogramme und Zweckprogrammen. Diese unterscheiden sich v. a. bzgl. ihrer Strukturiertheit und Veränderlichkeit mit entsprechenden Vor- und Nachteilen.

Organisatorische Selbstabstimmung (s. [486]): Eine neuere Integrationsformen stellt die horizontale Abstimmung als eine Form der Selbstabstimmung dar. Dazu können einerseits formale Regelungen (d. h. explizit modellierte, verbindliche Relationen) und andererseits informale bzw. informelle Relationen (d. h. als Folge sozialer und personaler Netzwerke) dienen. Eine zweite, weit verbreitete Form ist die Matrixorganisation nach der Organisationselemente mind. zwei ebenbürtigen, aber unterschiedlich ausgerichteten Autoritätsdimensionen unterstehen (z. B. verrichtungs- und objektorientiert Dimension). Dabei ist das Lösen von Aufgaben und Problemen aus mehreren Perspektiven von Vorteil. Eine dritte, immer wichtiger werdende Form stellt die Projektorganisation dar. Im Gegensatz zu klassischen Organisationsmodellen dient die vorübergehende Projektorganisation für ein einmaliges Vorhaben mit zeitlich definiertem Beginn und Ende sowie mit der Instanz eines Projektleiters. Als Sekundärorganisation überlagert die Projektdimension die bestehende Primärorganisation (z. B. funktionale, divisionale Dimension). Dabei kann je nach gewährten Leitungskompetenzen des Projektleiters (d. h. Matrix-Instanz) die projektorientierte Dimension durch eine nachrangige, gleichberechtigte oder vorrangige Relation in die Primärorganisation integriert werden (s. [120] für Details zur Projektorganisation in der PE).

Moderne Organisationsmodelle

Abstimmung durch Netzwerke (s. [486]): Mit intra-organisatorischen Netzwerken werden Organisationsmodelle mit weitestgehend teilautonomen Teams beschrieben. Dabei setzen sich die Teams (i. S. v. Kompetenzgruppen) entsprechend der für eine Aufgabe erforderlichen Mitglieder und

deren Kompetenzen zusammen (vgl. auch Center-Strukturen). Fehlt bspw. einem Entwickler im Entwicklungsprojekt Radsatzgetriebe Wissen z. B. zu leichtbaugerechter Konstruktion, kann sich dieser als Querschnittsmitglied der Kompetenzgruppe F&E Leichtbau dort Know-how von Experten beschaffen. Die oft vorzufindende Kompetenzgruppe Cost-Center stellt z. B. die Einhaltung des Kostenbudgets aller Projektgruppen sicher (s. [390]).

Zellstruktur-Design (s. [431]): Das Zellstruktur-Design ist auch ein netzwerkartiges Organisationsmodell mit teilautonomen Teams, die die kleinste Organisationseinheit sind und daher als Zellen bezeichnet werden. Dabei ist eine Zelle ein Art Mini-Unternehmen im Unternehmen und somit ein funktional integriertes Team (vgl. unternehmensinternes Start-up). Alle erforderlichen Funktionen werden Rollen zugeordnet und jedes Teammitglied hat i. d. R. mehrere Rollen inne. Der in konzentrischen Kreisen – hierarchisch gesehen von innen nach außen – ist charakteristisch für diesen Ansatz. So sind die Unternehmensleitung sowie zentrale Unternehmensfunktionen als Zellen im Zentrum und Zellen, die holistische Wertschöpfung für Kunden generieren in den äußeren Kreisringen zu verorten. Dabei können Zellen in der Peripherie (z. B. Produktlinien) sowohl dauerhaft als auch temporär (z. B. Entwicklungsprojekte) sein.

Modulare Prozessorganisation (s. [306], s. auch [220]): Die modulare Prozessorganisation von KÖBLER stellt ein wandlungsfähiges Organisationsmodell für die PE dar. Darin stehen zur Ableitung der vier möglichen Innovationsarten, d. h. markt- und technologieinduzierte Produktinnovationen, technologieinduzierte Prozessinnovationen sowie Prozessoptimierungen, vier entsprechende Klassen generischer Kernprozesse zur Verfügung. Kernprozesse leisten die zentralen Beiträge zu den Kernkompetenzen einer Organisation und bestehen aus modularen bzw. adaptiven Teilprozessen, die beim Durchgehen der Analyse-, Strategie-, Entwicklungs-, Planungs- und operativen Ebene aus einer Matrix nach bestimmten Kriterien zusammengesetzt werden (vgl. Vorgehen im morphologischen Kasten). Dabei hat v. a. das Management der Kernprozesse die Unternehmensführung inne.

Community of Practice (CoP) (s. [601], s. auch [600]): Eine CoP ist eine i. d. R. eine domänenbezogene, hierarchieunabhängige, teils informelle Wissensgemeinschaft von Personen zum Wissens- und Informationsaustausch (s. v. a.). Aufgrund der selbstorganisierenden Dynamik von CoPs mit sich ändernden Teilnehmer sind diese nur eingeschränkt steuer- und auch modellierbar [484]. Die drei Grundkonzepte zur Modellierung einer

CoP sind nach WENGER ET AL. die Domäne (i. S. v. Themenbereich), die Community (i. S. v. Teilnehmenden mit ihren Beziehungen) sowie die Practice (i. S. v. von geteiltem Wissen, Erfahrungen).

Kostenorientierte Modellierungsansätze

Abschließend wird bzgl. der Domäne Kosten ein kurzer Überblick über für diese Arbeit relevanten **Ansätze zur Modellierung von Kosten** gegeben. Für eine umfassende Übersicht zum Kostenmanagement in der integrierten PE im Allgemeinen sowie auch von Kostenmodellen, wie bspw. zur Kostenrechnung und Kostenfrüherkennung wird auf das Standardwerk „Kostengünstig Entwickeln und Konstruieren“ von EHRENSPIEL ET AL. [152] verwiesen (s. auch [394]). Eine Übersicht von qualitativen und quantitativen Ansätzen zum (Produkt-)Kostenschätzen mit ihren jeweiligen Vorteilen und Grenzen ist in [404] zu finden. Aufbauend auf und ergänzend zu [404] ist in MÖRTL & SCHMIED aus Perspektive des Design to Cost (d. h. Methodik, die Kosten als Entwicklungsparameter einbezieht) eine sehr systematische und umfassende Übersicht von Kostenmodellen, -methoden und Softwarewerkzeugen sowohl zur Kostenschätzung als auch zu entwicklungsbegleitenden Kalkulationen in [395] zu finden. Darin befinden sich auch Ansätze zur Verknüpfung von Kostenmodellen mit matrixbasierten Modellen (s. Kapitel 3.2.3). In dieser Arbeit sind neben dem Target Costing v. a. die Ansätze zum Kostenschätzen und zur entwicklungsbegleitenden Kalkulation relevant (s. Bild 128), so dass eine Auswahl nachfolgend kurz vorgestellt wird.

Kostenschätzung (s. [152]): Das Schätzen von (Herstell-)Kosten ist zwar ungenauer als Berechnungen aber schneller und oft ausreichend genau. Dabei sind Erfahrungen aus ähnlichen Situationen hilfreich. Die in Bild 128 angeführten vier Maßnahmen erhöhen die Genauigkeit der Kostenschätzung. Im vorgestellten Konzept in Kapitel 5 können diese an der Kostenschätzung Beteiligten mit den Elementen relevanter Domänen in Beziehung gesetzt und dadurch die Bewertung unterstützt werden.

Kurzkalkulation (s. [152]): Diese Ansätze dienen zur Ermittlung von Kosten für gewisse Kostenträger mit Hilfe der wesentlichen und bereits bekannten Einflussparameter. So sind bspw. fertigungstechnische Parameter (z. B. Schnittgeschwindigkeit, Vorschub beim Drehen eines Zylinders) in der PE noch nicht bekannt. Zur Ermittlung relevanter Einflussparameter dient das in dieser Arbeit erarbeitete Konzept und daher ergeben sich vielfältige Verknüpfungsmöglichkeiten zu diesen Ansätzen.

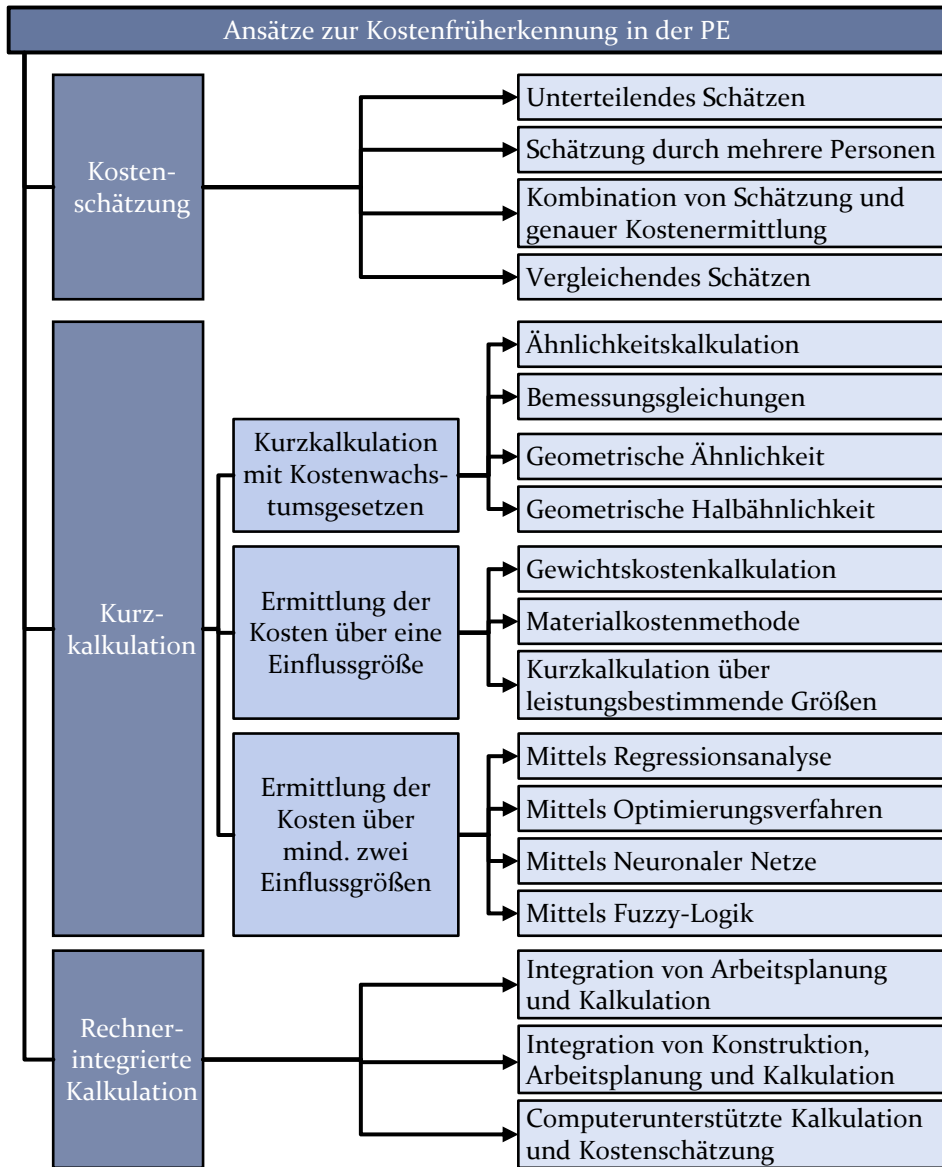


Bild 128: Ansätze zur Kostenfrüherkennung in der PE (in Anlehnung an Gliederung in [152])

Rechnerintegrierte Kalkulation (s. [152]): Hierbei wird die Zielsetzung verfolgt, dass die in CAD-Modellen festgelegten Daten an ein Kalkulationsprogramm zur Kostenrechnung übergeben werden können. Dafür bestehen bereits einige Ansätze und Software-Tools für die computerunterstützte Kalkulation und Kostenschätzung. Für diese Arbeit sind v. a. die

Ideen zur CAD-Kostenkoppelung, die featurebasierte Wiederhol- und Ähnlichkeitssuche von Interesse.

Target Costing ([502], s. a. [503], [152], [222]): Der Ansatz Target Costing (dt. Zielkostenmanagement) dient zur marktorientierten Beeinflussung von Kosten im PEP. Dabei berechnen sich die Target Costs aus der Differenz des erzielbaren Absatzpreis (v. a. abhängig vom Kundennutzen) und des angestrebten Zielgewinns. Das Target Costing verfolgt durch Analysen von Märkten und Kunden bereits zu Beginn des PEPs bestimmte Produkteigenschaften kosten- und nutzungsgerecht aus Sicht der Kunden zu bewerten. Durch den Vergleich der Target Costs mit den Drifting Costs (d. h. Produktkosten bei der Beibehaltung bestehender Technologien und Prozesse) wird das Ziel für kostenreduzierende und/oder nutzensteigernde Maßnahmen ermittelt. Aufgrund der i. d. R. höheren Drifting Costs sind die Zielkosten auf einzelne Funktionen bis zu Komponenten herunterzubrechen, um zu ermitteln, wie viel bestimmte Produktelemente aus Kundensicht kosten dürfen.

Integrated Value Engineering (IVE) (s. [367]): Mit dem IVE-Ansatz wird das Ziel der Optimierung des Produktwerts (i. S. v. Wert gleich Nutzen durch Kosten) verfolgt. Dazu setzt IVE zum einen bereits in frühen PEP-Phasen an, um durch die höheren Einflussmöglichkeiten besser den Produktwert optimieren zu können, und zum anderen Gestaltungsmöglichkeiten ein, die sowohl zur Erhöhung des Produktnutzen als auch zur Reduzierung der Produktkosten dienen. Neben dem detaillierten Leitfaden zur Anwendung des IVE-Ansatzes sind v. a. auch das Mapping der Target Costs (Zielkosten) mit den Drifting Costs (Standardkosten) mit Hilfe einer MDM bestehend aus den drei Domänen Anforderungen, Funktionen und Komponenten für diese Arbeit hilfreich. Jedoch führt die grundsätzlich fehlende Softwareunterstützung zu einem hohen Aufwand sowie die nicht vorhandene IT-Schnittstelle zu einer fehleranfälligen Akquise erforderlicher Elemente und Relationen aus den Domänen.

Prognosemodell für indirekte Änderungskosten (s. [205]): Ziel des Prognosemodells ist die Quantifizierung indirekter Änderungskosten im Voraus. Dazu dienen die entwickelten partiellen Prognosemodelle und -funktionen des kausalanalytischen Prognosemodells, welche auf einem formalisierten und theoriebasierten Erklärungsmodell beruhen. Aus dieser eher theoretischen Arbeit sind insb. die Ansätze zur Parametrisierung der partieller Prognosefunktionen für diese Arbeit von Interesse.

Parametrisches Kostenmodell zur Vorhersage der Produktentstehungskosten (s. [255]): Mit dem parametrischen Kostenmodell, das auf gewissen multivariaten parametrischen Kostenermittlungsverfahren beruht, wird das Ziel einer frühzeitigen Prognose der Produktentstehungskosten verfolgt. Für die vorliegende Arbeit ist v. a. der zweite Schritt des Modellierungsprozesses von Kostenmodellen mit folgenden vier Schritten von Bedeutung: Strukturierung und Abgrenzung der relevanten Kosten, Erstellung des hypothetischen Kostenmodells aus abhängigen und nicht abhängigen Modellparametern mittels der Auswahlmatrix, Überführung der Hypothese mittels bestimmter statistischer Verfahren in ein Erklärungsmodell sowie Prognose zukünftiger Entwicklungen mit dem Prognosemodell auf Basis des Erklärungsmodells.

Leitfaden für ein transdisziplinäres Zielkostenmanagement (s. [631]): Zielsetzung des Leitfadens ist die Unterstützung bei der Entwicklung komplexer mechatronischer Systeme in der Konzeptfindung im Hinblick auf Kostengesichtspunkte. Dabei wird zur Modellierung der Produktfunktionen eine DSM verwendet, die mittels DMMs mit den einzelnen Prozessen und Ressourcen als auch Komponenten verbunden wird. Da unterschiedliche Produktkonzepte und -strukturen auch unterschiedliche Prozesse bzw. Ressourcen für die Produktrealisierung benötigen, verursachen diese auch entsprechend unterschiedliche Kosten. Durch die Verwendung geeigneter Kostenschätzmethoden können kostenorientierte Konzeptoptimierungen unterstützt werden. Für diese Arbeit sind v. a. die Integration der Berechnung der Prozessschätzkosten aus der Schätzung des Ressourcenverzehrs, die Umrechnung der Prozessschätzkosten in Komponentenschätzkosten sowie die darauf aufbauende Mischschätzung von Funktions-, Komponenten- und Prozesskosten hilfreich.

Weitere kostenorientierte Modellierungsansätze sind zu indirekten Änderungskosten in den Publikationen von Schmied et al. zu finden (vgl. u. a. [475], [476], [477]). Zudem ist eine methodische Unterstützung zur kostenbasierten Auswahl modularer Produktstrukturen in [454] zu finden. Außerdem sind auch in der VDI-Richtlinie 2225 „Konstruktionsmethodik. Technisch-wirtschaftliches Konstruieren“ verschiedene Ansätze zur Kostenermittlung in der PE zu finden (s. [571], [573], [574], [572]).

A.17 Grundlagen zu ontologiebasierten Modellen

Für ein besseres **Verständnis von Ontologien** und zur Vertiefung einiger dafür erforderlicher Grundlagen dienen die folgenden Abschnitte. Darin wird jeweils auch auf entsprechende Fachliteratur aufmerksam gemacht.

Ontologien zur Abbildung von Modellen

Ontologien können zur Abbildung von Modellen eingesetzt werden. Mit der Definition, Erstellung und Nutzung von Ontologie beschäftigt sich der **Forschungsbereich Ontology Engineering** (vgl. [250]). Im Fokus des Ontology Engineering steht das Erarbeiten einer sog. T-Box (z. T. auch als „Ontologie“ oder „Schemaebene“ benannt), mittels deren Terminologie und Konzepten der Realitätsausschnitt beschrieben wird [250]. Nach u. a. [17] und [250] weist eine gute Ontologie v.a. folgende Eigenschaften auf:

- Semantische Integrität: Ontologien stellen als semantische Modelle viele Definitionsmöglichkeiten mittels mathematisch-logischer Konstrukte zur Verfügung (z. B. keine logischen Widersprüche);
- Ausgewogener Detaillierungsgrad: Ontologien repräsentieren die relevanten Relationen des Realitätsausschnitts i. S. v. einer ausgewogenen Balance zwischen Allgemeingültigkeit und Spezifität;
- Zeitliche Stabilität: Ontologien modellieren die reale Welt in einer Art und Weise, die über eine gewisse Zeit stabil und gültig ist;
- Mehrheitsfähigkeit (d. h. allgemeines, gemeinsames Verständnis): Ontologien sind so modelliert, dass viele Modellersteller und -nutzer sich in der Ontologie wiederfinden und gemeinsam verstehen;
- Wiederverwendbarkeit: Ontologien sind möglichst generisch und allgemeingültig definiert und als Ausgangspunkt für spezifischere Ableitungen dienen.

In der Literatur (s. [17], [219], [250]) lassen sich für die Erstellung von Ontologien eine Vielzahl verschiedener Methoden und Vorgehensweisen finden. **Methodiken für das Ontology Engineering** sind bspw. die Methodik nach USCHOLD & KING [566], die Methodik nach GRÜNINGER & FOX [230], die Methodik nach NOY & MCGUINNESS [410], METHONTOLOGY nach FERNANDEZ-LOPEZ ET AL. [177], On-To-Knowledge nach STAAB ET AL. [522] sowie Unified Process for Ontology (UPON) nach DE NICOLA ET AL. [405].

Wissensmetaprozess für die Entwicklung einer Ontologie

Als ein Beispiel für die **Entwicklung einer Ontologie** für eine Wissensmanagementanwendung (WM-Anwendung) dient bspw. der sog. Wissensmetaprozess nach STAAB ET AL. [520], der nach [522], [523] alle erforderlichen Aspekte enthält. Dabei orientiert sich das Vorgehen des Wissensmetaprozesses an CommonKADS, einer etablierten Methodik zur Einführung wissensbasierter Systeme [483]. Zwar enthält CommonKADS eine Vielzahl von verschiedenen Aspekten von Wissensmanagementsystemen, umfasst aber keine geeignete Methode zur Entwicklung von Ontologien. Dafür dient nach STAAB ET AL. [520] der Wissensmetaprozess, der im Bild 129 dargestellt und ausführlich in [520] erläutert wird.

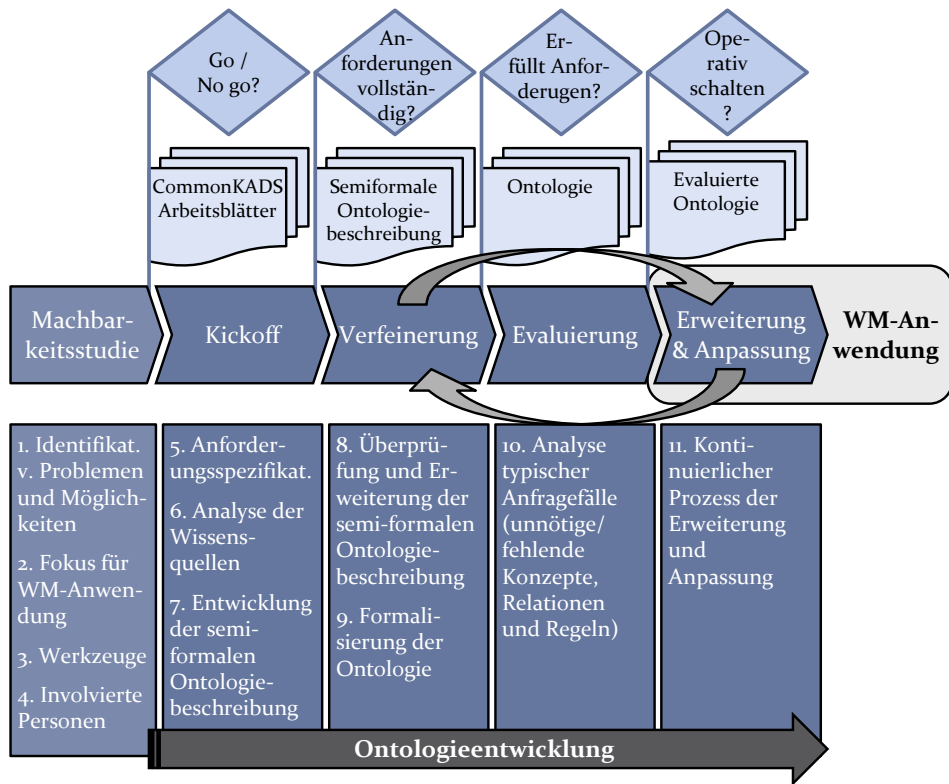


Bild 129: Wissensmetaprozess zur Entwicklung einer Ontologie (nach [522], [520])

Ontologie als eine Methode der Wissensrepräsentation

Für die Wissensrepräsentation gibt es eine Vielzahl verschiedener Ansätze. Beispiele für **Wissensrepräsentationsformen** sind neben Ontologien auch kontrollierte Vokabularen, Taxonomien, Thesauri, Semantische Netze und Metamodelle, wobei letztere ausführlich in Anhang A.15 erklärt sind. Diese werden in diesem Abschnitt vorgestellt und v. a. (hierarchisch) zueinander in Beziehung gesetzt. Dazu eignet sich besonders die Begriffshierarchie der Wissensrepräsentationsformen nach THEL [549] in Bild 130, welche im Folgenden Bild 131 mit einem durchgehenden Beispiel für die einzelnen Formen der Wissensrepräsentation veranschaulicht werden. Sofern eine festgelegte Grammatik befolgt wird, können alle in Bild 130 skizzierten Strukturierungssysteme als Grundlage für eine Ontologie verwendet werden [549].

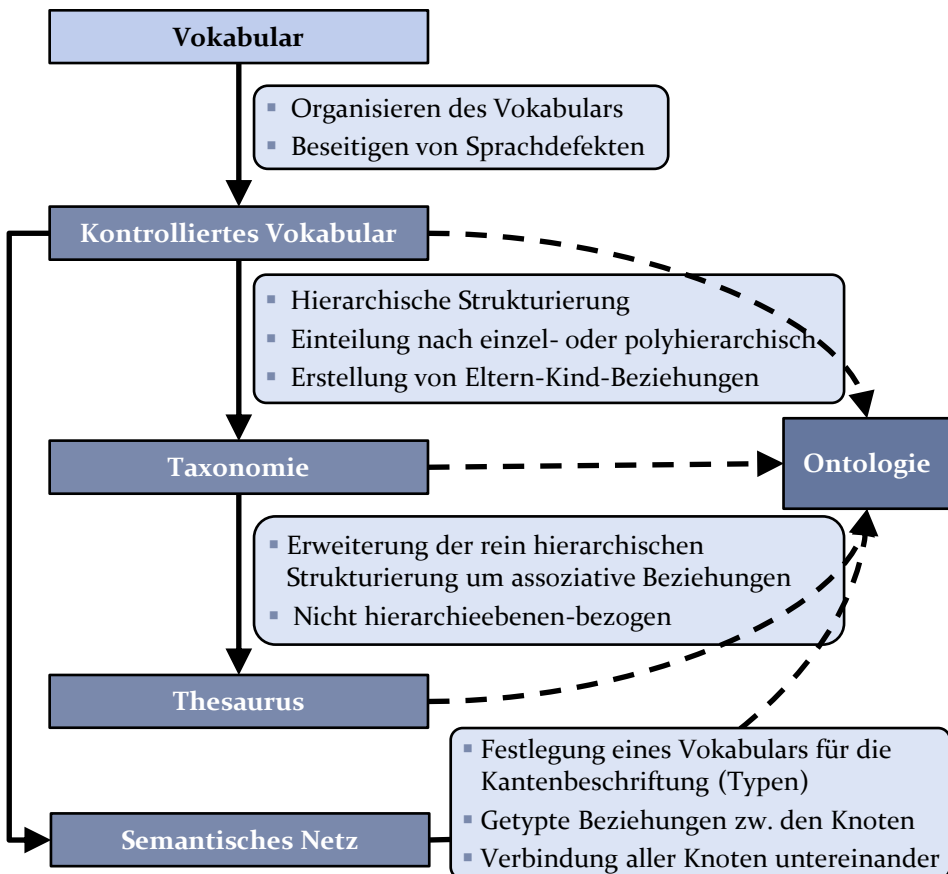


Bild 130: Begriffshierarchie von Wissensrepräsentationsformen und deren Relationen (in Anlehnung an [549])

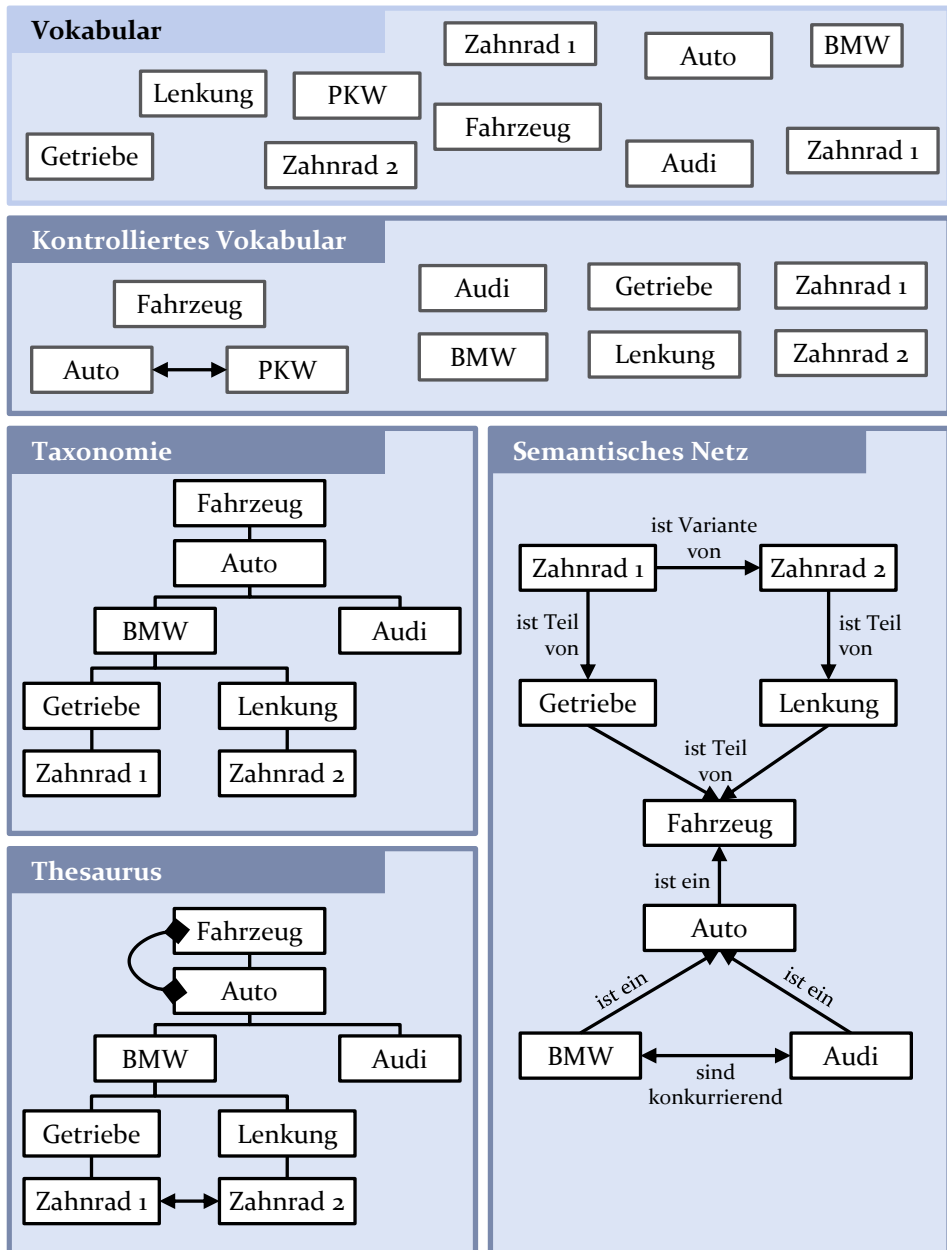


Bild 131: Beispiele für Wissensrepräsentationsformen (in Anlehnung an [549])

Diese fünf in Bild 130 Wissensrepräsentationsformen (d. h. Kontrollierte Vokabularien, Taxonomie, Thesaurus, Semantisches Netz und Ontologien) werden nachfolgend nach THEL zusammenfassend erläutert, so dass auf ein Referenzieren des Autors bzw. der Literaturquelle [549] z. T. verzichtet wird. Dadurch sollen insb. **Gemeinsamkeiten und Unterschiede dieser Wissensrepräsentationsformen** deutlich werden.

Kontrollierte Vokabularien stellen Sammlungen von Bezeichnungen dar (z. B. Glossare, Indizes, Kataloge, Verzeichnisse). Dabei haben die einzelnen Elemente dieser Vokabularien keine Beziehungen miteinander (vgl. Bild 131). [549]

Taxonomien sind insb. vertikale, hierarchische Strukturierungen von Begriffen. Mit Hilfe der hierarchischen Organisation ist die Modellierung von Klassen mit entsprechenden Unterklassen möglich (z. B. hat die Klasse Auto die beiden Unterklassen BMW und Audi, vgl. Bild 131). Dabei bestehen zwischen diesen hierarchisch angeordneten Klassen Eltern-Kind-Beziehungen. Im Gegensatz zur einzelhierarchischen können bei der komplexeren, polyhierarchischen Taxonomie einzelne Begriffe auch mehrere Eltern haben. Für derartige Modellierungszwecke wird v. a. ein Strukturbaum eingesetzt, der in Top-down-Richtung spezialisierter sowie in umgekehrter Richtung generalisierter wird. [549]

Thesaurus stellt ein kontrolliertes Vokabular dar, dessen Begriffe durch Relationen miteinander verbunden sind. Neben hierarchischen Relationen (vgl. Taxonomie) hat ein Thesaurus auch assoziative Relationen zwischen verschiedenen Begriffen. Dadurch wird sowohl eine hierarchische als auch eine horizontale (i. S. v. gleichgestellte) Begriffsstrukturierung möglich (vgl. Bild 131). Typische Anwendungsbeispiele von Thesauri sind Suchmaschinen, welche das Suchwort auf Grundlage eines Thesaurus durchforsten (z. B. nach Synonymen sowie sinnverwandten Begriffen), und Schlagwortlisten, die bei der Indexierung von Dokumenten hilfreich sind. [549]

Semantische Netze sind netzwerkartige Strukturen, die Relationen zwischen den Elementen und deren Bedeutungen (vgl. Semantik) modellieren. Somit dient ein semantisches Netz bestehend aus Kanten und Knoten zur Begriffsstrukturierung bzgl. einer Anwendungsdomäne. Dabei stellen die Knoten die Begriffe (Konzepte) und die Kanten deren Relationen dar, welche durch semantische Verbindungen charakterisiert werden (z. B. „ist ein“ und „ist Variante von“, s. Bild 131). Hierfür ist es zweckdienlich eine Menge an standardisierten, vordefinierten Vokabularien festzulegen, so dass das semantische Netz übersichtlich bleibt. Daraus lassen sich folgende Voraussetzungen zur Bildung eines semantischen Netzes ableiten [549]:

- Namenseindeutigkeit: Zwei Knoten ungleicher Bezeichnung werden als verschieden anerkannt;
- Einmaligkeit: In einem semantischen Netz existieren keine identischen Kanten;
- Implizite Konjunktion: Alle Knoten eines semantischen Netzes stehen durch Kanten in Beziehung und sind somit vernetzt

Ontologien sind bereits in Kapitel 3.2.2 erläutert und definiert. Daher folgt nachfolgend nur eine Diskussion zur **Abgrenzung der Ontologie von den anderen Wissensrepräsentationsformen**. Ein (nicht) kontrolliertes Vokabular sagt nicht nur nichts über die einzelnen Beziehungen zwischen den Begriffen, sondern auch nichts über die Bedeutung dieser Begriffe aus. Die Begriffe eines Vokabulars dienen nur als Grundlage für eine Ontologie. Sofern eine Taxonomie eine umfangreiche Grammatik besitzt, kann diese auch als eine Ontologie aufgefasst werden. Als Ergebnis der hierarchischen Relation(en) hat eine Taxonomie schon automatisch eine Ontologie, die aber auf diesen hierarchischen Strukturierungsansatz zwischen den Konzepten beschränkt ist. Während der Thesaurus diese hierarchischen Relationen um assoziative Relationen ergänzt, erweitern semantische Netze diese beiden Arten von Relationen nochmals um semantische Relationen. Daher können diese Wissensrepräsentationsformen für die Wissensrepräsentationsform Ontologie als Grundlage verwendet werden. Im Gegensatz zu den in Bild 131 dargestellten Wissensrepräsentationsformen weisen Ontologien grundsätzlich eine profiliertere interne Struktur sowie einen höheren Komplexitätsgrad auf [549]. Somit können Ontologien nach LEHNER ET AL. zur Beschreibung aller Arten von Ordnungsschemata bzw. Klassifizierungen verwendet werden [345]. Für eine ausführliche Erläuterung der Gemeinsamkeiten und Unterschiede zwischen Ontologie auf der einen und kontrollierten Vokabularen, Taxonomien, Thesauri und semantischen Netzen auf der anderen Seite sei auf die Dissertation von THEL [549] sowie auf [245] und für ein anschauliches Beispiel auf [345] verwiesen.

Klassifizierung von Ontologien

Für ein besseres Verständnis von Ontologien dient deren Klassifizierung. In der Literatur lassen sich viele Klassifizierungen von Ontologien finden, was sich auch mit der hohen Anzahl an Begriffsdefinitionen von Ontologie erklären lässt. Nach HAASE können Ontologien anhand der **Dimensionen Formalität und Allgemeinheit** klassifiziert werden [235] (vgl. auch [232], [411]. Die Grundlage der nachfolgenden Erläuterungen bildet daher die Publikation [235].

Die **Dimension Formalität** kann prinzipiell in formale, semi-formale und informale Ontologie gegliedert werden. Dabei ist die informale Ontologie die einfachste Form einer Ontologie, die nur eine Menge an konzeptuellen Begriffen umfasst. Diese Begriffe sind hierarchisch strukturiert drücken ihre Definitionen mit Hilfe natürlicher Sprache aus. Dagegen enthalten formale Ontologien weitere Definitionen und Axiome, die in einer formalen Sprache modelliert sind und somit von Rechnern genutzt werden könne. Als Zwischenform lassen sich auch semi-formale Ontologie beschreiben. Semi-formale Ontologie sind teils in formaler und teils auch in informaler Sprache modelliert. Während formale Ontologien für den Austausch von Wissen zwischen Computern eingesetzt werden, werden informale Ontologien für den Wissensaustausch zwischen Personen.

Die **Dimension Allgemeinheit** kann in fünf Abstufungen gegliedert werden. Top-Level-Ontologien werden zur Definition von äußerst allgemeinen Konzepten (z. B. von Zeit, Raum) eingesetzt und sind unabhängig von einer spezifischen Domäne. Dagegen sind Kern-Ontologien zwar spezifizierter als Top-Level-Ontologien, werden aber auch zur Modellierung domänenübergreifender Konzepte verwendet (z. B. zur Modellierung diverser Arten von Web Services [411]). Die Domänen-Ontologien stehen in sehr enger Beziehung mit einer spezifischen Anwendungsdomäne (z. B. der Disziplin PE). Aufgaben-Ontologien werden zur Definition von Konzepten eingesetzt, die dazu dienen eine spezifische Aufgabe oder einen Prozess auszuführen (z. B. eine Zeit- oder Terminplanung). Zudem beschrieben Anwendungsontologien Konzepte für eine einzelne Anwendungsdomäne, die häufig in einem engen Zusammenhang mit einer ganz spezifischen Domäne oder speziellen Aufgabe stehen.

Ontologien zur Wissensrepräsentation in der PE

Es gibt es eine **Vielzahl verschiedener Wissensrepräsentationsformen** in der Entwicklung, die im folgenden überblicksartig zusammengefasst werden. Bei regelorientierten Repräsentationsformen werden Regeln eingesetzt, die sich aus einem Bedingungsteil (Prämisse) und einer Konklusion zusammensetzen (i. S. v. Wenn-Dann-Beziehungen, vgl. auch [P31]). Dagegen werden in der objektorientierten Repräsentation Objektstrukturen verwendet, um Wissen darzustellen. So können z. B. Vererbungsbeziehungen zwischen Klassen für die Repräsentation ähnlicher, sich wiederholender Objekte (z. B. Baugruppen, Teile) eingesetzt werden [P5]. Zur Wissensrepräsentation in der Entwicklung werden nach SPUR & KRAUSE sowohl syntaktische (i. S. v. Regeln der Anordnung) sowie semantische (i. S. v. Regeln der Bedeutung) Vorschriften verwendet [519]. Mit Hilfe von

logikorientierten Repräsentationsformen werden Elemente und Relationen in Aussagen (i. S. v. Sätzen) modelliert, deren Inhalt entweder richtig oder falsch sein kann. Dabei kann durch die Verbindung vieler Aussagen zum einen komplexes Wissen beschrieben werden und zum anderen (implizites) Wissen geschlussfolgert (d. h. inferiert) werden [P5]. Dagegen werden sog. Constraint-Repräsentationen dazu verwendet, um Relationen zwischen verschiedenen Größen darzustellen (z. B. Berechnung der Dehnsteifigkeit des Rohrs als Multiplikation von Elastizitätsmodul und Querschnittsfläche des Rohrs, vgl. Anhang A.4) [P5].

Eine zentrale **Zielsetzung der Wissensrepräsentation durch Ontologie ist Inferenz** (d. h. Wissensgewinnung durch logisches Schlussfolgern bzw. Reasoning) in verschiedenen Anwendungsdomänen mit hoher Komplexität. Hier sind die Methodologie DILIGENT zur Unterstützung des dezentralen Wissensmanagements in [434] und die domänenspezifische, ontologiebasierte Sprache zur Konfiguration von Kommunikationsnetzwerken mit diversen Möglichkeiten zum Schlussfolgern in [586] zwei Beispiele mit jeweils einem Fokus auf Grundlagen zu Ontologien.

Aufgrund der vielen Vorteile und Nutzenpotentiale werden **Ontologien in der PE** für vielfältige Anwendungsfälle und Aufgaben eingesetzt. Als Beispiele mit Anwendungsbezug v. a. in der PE können folgende Publikationen angeführt werden:

- Die Design Ontologie von ŠTORGA ET AL. für eine effizientere PE durch eine geeignete Beschreibung von Wissen für dessen Wiederverwendung [532], welche auf dem Vorgehensmodell genetic design model system (GDMS) von MORTENSEN (s. [393]) aufbaut;
- Eine Ontologie von AHMED & ŠTORGA für die PE in Form eines Template (i. S. v. Vorlage), die sowohl Forscher in der Wissenschaft als auch Praktiker in der Industrie auf ihren jeweiligen Kontext adaptieren können [6];
- Die Design Activity Ontology (DAO) von KUMAR zur Modellierung und Analyse v. a. von Aktivitäten und Informationsflüssen in der Entwicklung [333], wobei die Ontologie selbst mittels der DSM-Methodik modelliert und analysiert wird [334];
- Der Ansatz von SYLDATKE ET AL. zur Modellierung von Domänen inkl. deren Elementen und Relationen als Ontologie, die in einer MDM dargestellt (d. h. mit inter- und intra-Domänen Relationen i. S. v. DSM und DMM) und mit dieser auch analysiert werden kann [544]

- Das Konzept zum Einsatz von Semantic-Web-Technologien von HAHN für eine wissensorientierte Integration von verteilten Produktmodellen mittels Ontologien [241];
- Der Ansatz von WOLL ET AL. für eine ontologiebasierte Datenintegration im Rahmen des MBSE, um verschiedene Modelltypen in unterschiedlichen IT-Systemen miteinander zu verknüpfen und zu bearbeiten [618];
- Das Konzept zur Performancemessung für PEPs von HAUSMANN, das v. a. auf der formalen Repräsentation eines integrierten Produktmodells mittels generischer Ontologiemodelle beruht [250];
- Die Ontologie von GAAG zur funktionsorientierten, semantischen Suche nach bewährten Lösungen in der PE aus Anwender- und Gebrauchssicht [200];
- Der semantische Ansatz von BREITSPRECHER ET AL. zur Annotation prinzipieller Lösung mit ontologischen Aussagen in der Konzeptphase, um beabsichtigte Anforderungen explizit und für die weitere maschinelle Verarbeitung verfügbar zu machen [78];
- Die Ontologie tribAIn von KÜGLER ET AL. für eine formale und explizite Spezifikation des Wissens im Bereich der Tribologie, um Versuchsaufbauten und -ergebnissen zu annotieren und zu finden [330].

Ontologien als Wissensrepräsentationssprachen

Die Web Ontology Language (OWL) (vgl. für Details [262]) sowie das Resource Description Framework (RDF) (vgl. für Details [79]) sind zwei der vom World Wide Web Consortium (W3C) **standardisierten Wissensrepräsentationssprachen**. OWL und RDF sind zwei von vielen beispielhaften Ontologiesprachen, die zur Modellierung und Nutzung von Ontologien verwendet werden können [449]. Mit Hilfe dieser beiden Sprachen können vielfältige, in der Entwicklung eingesetzte Datenstrukturen und Formate modelliert werden, wie bspw. relationale Datenbanken mittels Ontology-based Data Access (s. [435]) und die zuvor skizzierten Klassenmodelle (s. [111]). Neben Klassenbeschreibungen können mit Ontologiesprachen auch Regeln und komplexe Beziehungen mit Hilfe von Ontologiemustern (s. z. B. [203], [204]), domänenspezifische Sprachen (s. z. B. [586], [276]) oder Metadaten (s. [520]) repräsentiert und verwaltet werden.

Im Hinblick auf Ontologiesprachen selbst (i. S. v. Beschreibungssprachen) wird i. d. R. zwischen der Ebene der sog. **T-Box** und der **A-Box** differenziert (s. [400] und vgl. u. a. [521], [250]). Nach NARDI & BRACHMAN enthält die T-Box intentionales Wissen in Form einer Terminologie (engl.

„terminology“, daher der Begriff T-Box, wobei auch „Taxonomie“ denkbar ist) und setzt sich aus Aussagen zusammen, die die allgemeinen Eigenschaften von Konzepten beschreiben [400]. Die A-Box enthält erweitertes Wissen, das auch als Assertions- oder Behauptungswissen bezeichnet wird (engl. „assertion“ daher der Begriff A-Box), d. h. Wissen, das spezifisch für die Individuen der jeweiligen Anwendungsdomäne ist [400]. Dabei bilden MÖLLER & HAARSLEV die T-Box und die A-Box gemeinsam die Wissensbasis, wobei Inferenzmechanismen auf die T-Box oder auf T- und A-Box zusammen angewendet werden, um Fragen der Kohärenz (von Begriffen), der Konsistenz (von A- und T-Box), Subsumption (zwischen Begriffen der T-Box) oder Instanziierung (von Begriffen und Relationen) zu klären oder um konjunktive Anfragen zu beantworten [391].

Ontologie als Teil der Anwendungsarchitektur des Semantic Web

Insb. aufgrund der rasanten Entwicklungen im Bereich des Semantic Web wurden Ontologien v. a. für Informatik immer mehr von Interesse. Dabei besteht eine **Semantic-Web-Anwendung** – vereinfacht ausgedrückt – aus einer Wissensbasis, die ein Ontologie-Schema und Aussagen enthält, und einer Inferenzmaschine, die neue Aussagen aus dieser bestehenden Wissensbasis ableitet [101]. Dabei wird durch das Ontologie-Schema determiniert, welche Arten von Aussagen zulässig sind [101] (vgl. Kapitel 3.2.2).

Der Aufbau des Semantic Web erfordert nicht nur die Verwendung der zuvor beschriebenen Ontologiesprachen, wie OWL und RDF, sondern auch einen geeigneten Ansatz zur Anwendungsintegration. Hierfür ist in Bild 132 ein grober Überblick über eine **Anwendungsarchitektur** nach ANTONIOU ET AL. dargestellt, mit der diverse IT-Werkzeuge im Rahmen des Semantic Web integriert werden und deren Interoperabilität durch die Verwendung von Semantic Web-Standards realisiert wird [17].

Die in Bild 132 dargestellte Anwendungsarchitektur ist in [17] ausführlich allgemein und in [543] am Beispiel einer konkreten Ontologie ausführlich erläutert. Daher werden nachfolgend nur die wesentlichen **Bestandteile der Anwendungsarchitektur** (s. Bild 132) kurz nach [17] erläutert:

- Wissensakquisition: Die Akquisition von Wissen kann prinzipiell (semi-)strukturiert oder unstrukturiert erfolgen. Unstrukturierte Dokumente liegen meist in natürlicher Sprache und strukturierte und semistrukturierte Dokumente sind z. B. Datenbanken, HTML-Tabellen und Tabellenkalkulationen;
- Wissensspeicherung im Ontologie-Repository: Das akquirierte Wissen wird mittels RDF und einem RDF-Schema repräsentiert.

Das Repository speichert sowohl die Ontologie (Klassenhierarchie, Eigenschaftsdefinitionen) als auch die Instanzen der Ontologie (z. B. bestimmte Elemente, die zu Klassen gehören);

- Wissenspflege (i. S. v. Ontologie-Bearbeitung): Das Semantic Web-Repository stellt Administratoren grundlegende Funktionen zum Verwalten und Pflegen der Ontologie zur Verfügung (z. B. Änderungsverwaltung und -verwaltung, Zugriffs- und Eigentumsrechte);
- Wissensnutzung (i. S. v. Wissenszugriff und -bereitstellung): Die Ontologien und Daten im Repository müssen über Anwendungen den Nutzer bereitgestellt werden;

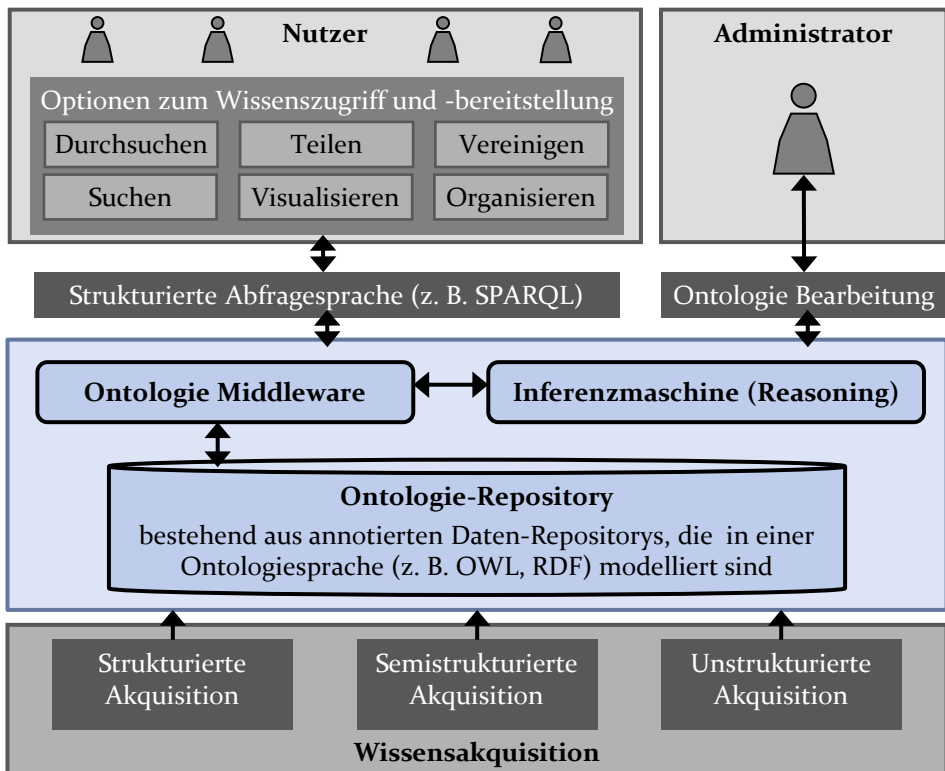


Bild 132: Semantic Web Anwendungsarchitektur (nach [543] und [17])

Dabei ist die **Repräsentation von Wissen als Ontologie** die wesentliche Grundlage von Semantic Web Anwendung. Durch geeignete Anwendungsprogramme kann das modellierte Wissen in der PE mit Hilfe intelligenter Suchanfragen effektiv, effizient und kontextbezogen zur Verfügung gestellt werden (vgl. u. a. [313] und die betreuten studentischen Arbeiten [S15] und [S2] für Beispiele). Dadurch kann Wissen mit Hilfe wissensbasierter Systeme u. a. geteilt und bearbeitbar gemacht werden.

A.18 Grundlagen zu matrixbasierten Modellen

In diesem Abschnitt sind überblicksartig **weitere Arbeiten angeführt, die im Rahmen matrixbasierter Modellierungsansätze relevant sind** und damit für diese Arbeit z. T. hilfreiche Grundlagen und wegweisende Impulse darstellen. In dieser Ergänzung des Stands der Technik werden die Ansätze neben der (teils selbst gewählten) Bezeichnung auch mit einer kurzen Erläuterung mit Fokus auf die für diese Arbeit relevanten Aspekte sowie Literaturhinweise zur Vertiefung der Erläuterung, Anwendung oder Bewertung des entsprechenden Ansatzes zusammengefasst.

Ansatz für ein Produktmodell mit einem Änderungsfortpflanzungsmodell (s. v. a. [414] sowie [416] für das Produktmodell und [415] für die Integration von TRIZ): Der von OIZUMI & AOYAMA erarbeitete Ansatz basiert v. a. auf das von OIZUMI ET AL. in [416] vorgestellte Produktmodell, welches u. a. auch dem Projektmanagement in der PE durch Modellierung von Produkt- und Prozessinformationen dient. In diesem Produktmodell werden Design Parameter mit deren Relationen zu funktionalen Metriken und Strukturelementen abgebildet und auf Grundlage der daraus resultierenden Abhängigkeiten mögliche PEP-Varianten abgeleitet. Dieses Produktmodell dient als Grundlage die modellbasierte Methode zur Bewertung von Änderungsfortpflanzung (change propagation) in Bezug auf die Funktionsstruktur des Produkts, welche in einer DSM abgebildet wird. Der Fokus des Modells liegt darauf, wie funktionale Metriken durch Veränderung(en) der Design Parameter verbessert werden. Dazu werden mittels DSMs die Relationen zwischen funktionalen Metriken und Design Parametern analysiert, um die Änderungsfortpflanzung in den funktionalen Metriken ableiten zu können. Zudem werden mit Hilfe von zwei Zusatzinformationen die Auswirkungen der Design Parameter auf die funktionalen Metriken beschrieben. So werden mit Sensitivitätsangabe die Stärke der Auswirkung (bzgl. einer Verbesserungsfunktion) und mit der Eigenschaftsangabe die Art und Weise (z. B. je größer/kleiner desto besser bzgl. einer Verbesserungsfunktion) der Auswirkung des Design Parameter auf die funktionale Metrik modelliert. Aufbauend auf diesen Arbeiten wird in [415] von OIZUMI & AOYAMA eine Methode zur Unterstützung der Ideenfindung auf Grundlage der Integration von TRIZ vorgestellt, die auf der Analyse von Kompromissen und physikalischen Widersprüchen in einem Produkt basiert.

Ansatz einer Model-Based DSM (MDSM) (s. [508]): Das zentrale Ziel des MDSM-Ansatzes ist die weitgehend automatische Generierung der modellbasierten DSM – also der MDSM – auf Grundlage des auch in [508] vorgestellten OPM-Projekt-Produktmodell (s. Kapitel 3.2.1), welches die relevanten Projekt-, Produkt- und Prozessobjekte enthält. So wird durch Verwendung eines Objekt-Prozess-Diagramm (OPD), welches die Modellierungssprache OPM (s. Kapitel 3.2.1) zugrunde liegt, insb. die Semantik der Relationen verdeutlicht. Das OPD ist in Bild 133 links als objektprozessbasierter Systemgraph und in Bild 133 rechts als entsprechender automatisch generierter Paragraph, welcher in der Object Process Language (OPL) formuliert ist, dargestellt.

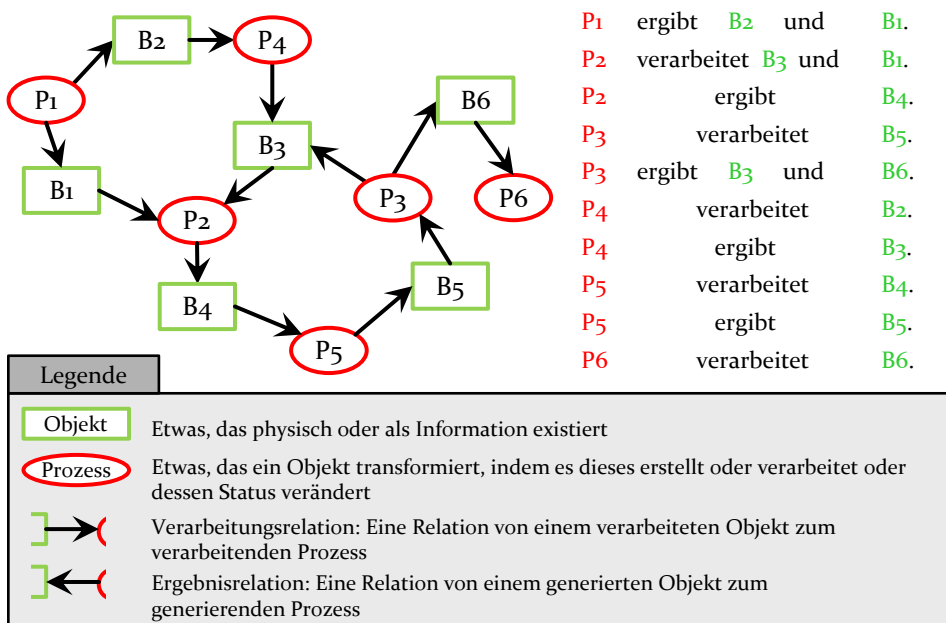


Bild 133: Beispiel für ein Objekt-Prozess-Diagramm (nach [508])

Dagegen ist die **Verwendung von DSMs zur mathematischen Analyse** (s. Kapitel 5.5) und Verbesserung von komplexen, technischen Systemen nach SHARON ET AL. besonders vorteilhaft. Zur Nutzung sowohl der Vorteile von OPM als auch von DSM beschreiben die Autoren ein Transformations-Framework, das eine bidirektionale Transformation zwischen dem OPM-Projekt-Produktmodell und der entsprechenden MDSM ermöglicht. Dabei wird zum einen der objektbasierte Systemgraph (vgl. „OPM-Baustein“ Objekt) in eine komponentenbasierte DSM und zum anderen der prozessbasierte Systemgraph (vgl. „OPM-Baustein“ Prozess) in eine aktivitätenbasierte DSM (vgl. Bild 23) transformiert. In Bild 134 oben ist die DSM der

OPD abgebildet, wobei C eine Art Verarbeitungsrelation (consumption link) von einem Objekt zu einem Prozess und R eine Art Ergebnisrelation (result link) von einem Prozess zu einem Objekt bezeichnen. Um das Muster der Schleife {P₂, B₄, P₅, B₅, P₃, B₃} in der OPD auch in der DSM besser zu erkennen sind in Bild 134 unten die Zeilen und Spalten der DSM neu angeordnet, so dass P₂, B₄, P₅, B₅ und P₃ in einer RCRCR-Sequenz angeordnet sind (vgl. Muster mit abwechselnden Rs und Cs direkt über der DSM-Diagonale). Dabei vervollständigt die DSM-Zelle (B₃, P₂) die Schleife als Scheitelpunkt eines gleichschenkligen Dreiecks, welches als Basis eine aus fünf Zellen bestehende RCRCR-Sequenz besitzt. Anhand eines Anwendungsbeispiels eines unbemannten Luftfahrzeugs werden u. a. Semantik und Analogien zwischen unterschiedlichen Arten von Relationen innerhalb des Systemmodells und der MDSM ausführlich aufgezeigt.

↖	P ₁	B ₁	P ₂	B ₂	P ₃	B ₃	P ₄	B ₄	P ₅	B ₅	P ₆	B ₆
P ₁		R		R								
B ₁			C									
P ₂							R					
B ₂						C						
P ₃					R							R
B ₃			C									
P ₄					R							
B ₄							C					
P ₅								C				
B ₅					C				R			
P ₆												
B ₆											C	

↖	P ₁	B ₁	P ₂	B ₄	P ₅	B ₅	P ₃	B ₃	B ₂	P ₄	P ₆	B ₆
P ₁		R							R			
B ₁			C									
P ₂				R								
B ₄					C							
P ₅						R						
B ₅							C					
P ₃								R				R
B ₃			C									
B ₂									C			
P ₄								R				
P ₆												
B ₆											C	

Bild 134: DSM als Repräsentation des OPD in Bild 133 (nach [508])

Ansatz zur Identifizierung von Produktmodulen basierend auf dem Axiomatic Design und der DSM (s. [104]): Dieser von CHENG ET AL. erarbeitete Ansatz dient zur systematischen und strukturierten Entwicklung modularer Produkte. Dazu wird das Produkt (z. B. eine Klimaanlage) gemäß der Axiomatic Design Methode (s. Kapitel 3.1.2) hierarchisch bzgl. der Funktions-, Physik- und Prozessdomäne zerlegt. Z. B. erfolgt dabei die Quantifizierung der Design Matrix am Beispiel einer Klimaanlage, in dem Mapping-Beziehungen zwischen den funktionalen Anforderungen in der Funktionsdomäne (in diesem Beispiel werden 5 Funktionshierarchieebenen modelliert) und die dazugehörigen Designparameter in der Physikebene bestimmt werden. Die als Ergebnis vorliegende Design Matrix wird in eine DSM transferiert. Anschließend werden die relevanten DSMs mit denjenigen Elementen erstellt, welche die Produktfunktion, -struktur und auch den Herstellungsprozess repräsentieren. Nach Integration dieser DSMs wird ein geeigneter genetischer Algorithmus (z. B. Cluster-Algorithmus mit minimaler Beschreibungslänge) zur Optimierung der modularen Produktarchitektur eingesetzt.

Ansatz zur systematischen Modul- und Schnittstellendefinition auf Basis komponentenbasierter DSMs (s. [254]): Die Zielsetzung des Ansatzes von HELMER ET AL. ist ein systematisches und praktisches Vorgehen zur Gestaltung der Modul- und Schnittstellendefinition von Produktarchitekturen mittels Clustering der komponentenbasierten DSMs. Da im Fokus des Ansatzes die Montage von Modulen steht, sind insb. die räumlichen Relationen (vgl. Bild 32) bei der Identifizierung und Bestimmung von Modulschnittstellen entscheidend. Grundsätzlich werden für die Modellierung der Relationen zwischen den Elementen aber die folgenden fünf Interaktionsmöglichkeiten berücksichtigt: Raum, Struktur/Bewegung, Energie, Signal/Information, Material/Stoff (s. Bild 32). Mit Hilfe einer Art Fallunterscheidung in jeder Zelle einer DSM zwischen diesen fünf Arten von Relationen wird der Umgang mit allen möglichen Interaktionsarten ermöglicht. Diese Fallunterscheidung bildet zusammen mit einem entwickelten Bewertungsschemata, dass die Wichtigkeit zwischen den fünf Interaktionsmöglichkeiten bewertet, die Basis für einen Nachbearbeitungsschritt zur Korrektur und Verfeinerung des automatisierten Cluster-Ergebnisses.

Ansatz zur Modulpartition für 3D-CAD-Baugruppenmodelle (s. [349], s. auch betreute studentischen Arbeit [S64]): Die Zielsetzung des Ansatzes ist mit Hilfe von hierarchischen Clustering-Verfahren auf der Grundlage der Abhängigkeitsstärke zwischen Komponenten bereits bestehende CAD-Baugruppenmodelle v. a. aus Perspektive der Montage in Module zu modularisieren. Dazu werden vier verbreitete hierarchische Clustering-

Verfahren verwendet, um mit Hilfe der Abhängigkeitsstärken zwischen den Komponenten, die in der komponentenbasierten DSM modelliert sind, die Produktstruktur zielgerichtet zu zerlegen, um dadurch v. a. für die Montage zweckmäßige modulare Produktstrukturen zu erzeugen. Für diese Arbeit ist allerdings weniger die Anwendung von Analyseverfahren (s. auch Kapitel 3.3.3), sondern vielmehr die Vorgehensweise zur Überführung von Daten aus bestehenden CAD-Baugruppenmodellen in DSMs interessant. In einem ersten Schritt werden nach dem Ansatz von LI & XIE die Komponenten-, Schnittstellen- und Beziehungsinformationen aus dem CAD-Modell in die DSMs überführt.

Anschließend wird in der DSM die **Stärke der Relation zwischen den Komponenten** modelliert und auf Basis dieser erfolgt die Modulpartition. Dabei wird die Abhängigkeitsstärke zwischen zwei Komponenten auf Basis ihrer relativen Freiheitsgrade (engl. degrees of freedom, kurz: DoF) bestimmt. Generell besitzt jede Komponente im dreidimensionalen Raum mit jeweils drei translatorischen und drei rotatorischen insgesamt maximal sechs Freiheitsgrade. Basierend auf den Ansätzen, die in [563] und [103] zu finden sind und aufgrund der Bewertung von HELMER ET AL., erfolgt die Bestimmung der Abhängigkeitsstärke auf Basis der DoFs zwischen zwei Komponenten auf einer zwischen „0“ und „1“ (normierten) Bewertungsskala (s. Bild 135). Dabei steigt die Abhängigkeitsstärke zwischen zwei in Beziehung stehenden Komponenten mit sinkendem Freiheitsgrad. So sind z. B. bei einer starren Relation zwischen zwei Komponenten (z. B. eine Schweißnaht) alle sechs Freiheitsgrade gesperrt und somit die Abhängigkeitsstärke maximal (d. h. „1“). Weitere mögliche Arten von Relationen mit deren Auswirkungen auf Freiheitsgrade sind in [583] zu finden.

Offene Freiheitsgrade (DoF)	6	$2 < \text{DoF} < 6$	$0 < \text{DoF} \leq 2$	$\text{DoF} = 0$ (fixiert)	„Interne“ Relation
Stärke der Relation	0	0,3	0,5	0,8	1,0

Bild 135: Bewertungsskala für die Stärke der Relation auf Basis der Anzahl der Freiheitsgrade (in Anlehnung an [349])

Ansatz für ein generisches Framework zur Produktmodellierung (s. [624], s. auch betreute studentischen Arbeit [S32]): Die Zielsetzung des Ansatzes ist mittels eines generischen Produktmodellierungs-Frameworks (GPMF) den Informationsaustausch insb. zwischen Montage, Fertigung und Entwicklung zu unterstützen, da bestehende Produktmodellierungstechnologien nach XIE ET AL. dazu nicht in der Lage sind. Das GPMF

verwendet STEP als Standard zum Austausch von Produktmodelldaten als Grundlage. Das GPMF besteht aus vier funktionalen Bestandteilen: einem EXPRESS-Datenmodell (kurz: EDM); einer STEP-basierten Modellierungsumgebung; einer Modellierungsmethode mit fünf Phasen; drei Methoden zum EDM-Datenaustausch- und -freigabe. Für diese Arbeit ist zum einen die grundsätzliche Kategorisierung der Hauptelemente des EXPRESS-Datenmodells (u. a. entity, type, constraint) und zum anderen die Modellierungsmethoden mit den folgenden fünf Phasen interessant: Modularisierung der EDM, Modellierungsobjekte, Relationen und Attribute, Ergänzung um Constraints und Modellintegration.

Actor-Based Signposting (ABS) (s. [249], s. auch [247], [248], [246]): Bei diesem Ansatz nach HASSANNEZHAD ET AL. wird die Entwicklungsorganisation aus Perspektive eines integrierten sozio-technischen Systems modelliert. Ziel ist die Unterstützung von Entscheidern bei Planungsfragen (z. B. Terminplanung, Ressourcenzuteilung) als auch bei Fragen der organisatorischen Planung des Entwicklungsprojekts (z. B. der Identifizierung von Kritikalität, Abstimmung notwendiger Fähigkeiten und Kenntnisse). Mit ABS soll die Organisationen unterstützt werden, anpassungsfähiger auf die VUCA-Welt zu reagieren.

DSM für das Integrated Concurrent Engineering (ICE) (s. [28]): Die Zielsetzung des ICE-Ansatzes ist es bei den agilen, interdisziplinären, einwöchigen PE für Raumfahrtmissionen zu unterstützen. Dazu werden im Rahmen des ICE-Ansatzes von AVNET & WEIGEL v. a. die parameterbasierte und organisationsbasierte DSM (s. Bild 23) zusammen mit verschiedenen Analyseverfahren verwendet. Die in [28] als Fallbeispiel verwendete parameterbasierte DSM eines typischen ICE-PEPs umfasst 172 Entwicklungsparametern und 682 Relationen und ist als alphabetische Abfolge der 16 beteiligten Disziplinen angeordnet (s. Bild 136 oben). Dabei repräsentieren die Cluster entlang der Diagonale die Aufgaben, die für jede Disziplin intern sind, und die nicht diagonalen Matrixeinträge stellen den Informationsfluss zwischen den Disziplinen dar. Nach der Analyse der DSM mittels des Partitionierungsverfahrens wird die partitionierte DSM für den optimierten ICE-PEP generiert (s. Bild 136 unten). Darin stellen die Cluster die 5 optimierten Phasen des ganzen PEPs dar, wobei der hellgraue Bereich die Entwicklung i. e. S. entspricht (dunkelgraue Cluster repräsentieren die Aufgaben jeder einzelnen Disziplin). Die wichtigsten Wechselwirkungen zwischen den technischen Subsystemen und Fachdisziplinen können durch eine tiefere Analyse der Entwicklungsphase gefunden werden. Dafür ist das Analysieren der Schleifen auf Basis der partitionierten DSM

hilfreich. Dadurch lassen sich die Relationen zwischen den Disziplinen, die sich aus diesen Schleifen ergeben, darstellen.

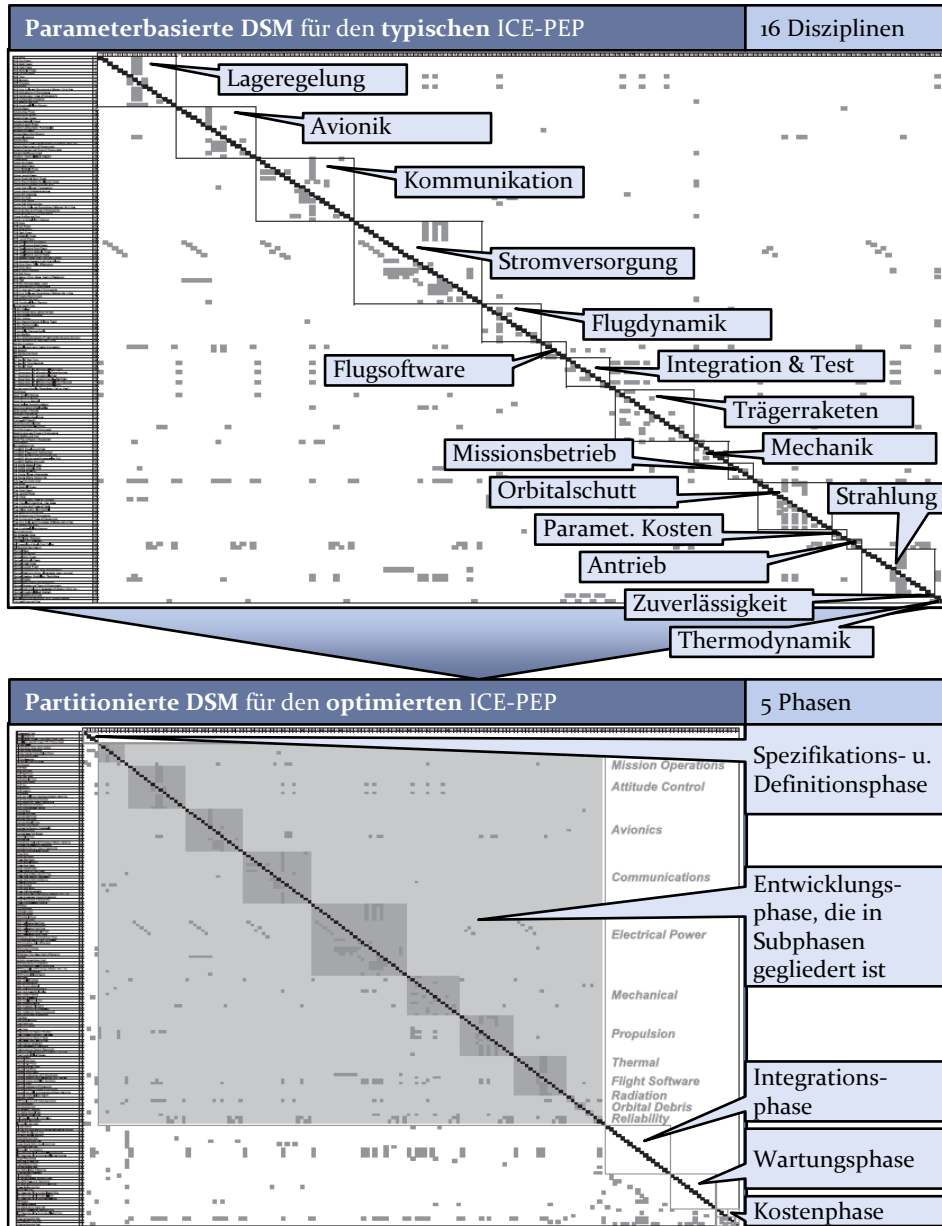


Bild 136: Anwendung der DSM im Rahmen des Ansatz Integrated Concurrent Engineering (zwei Screenshots entnommen aus [28])

A.19 Grundlagen zur graphbasierten Visualisierung

Ergänzend zum Stand der Forschung und Technik werden **Grundlagen zu graphbasierten Visualisierungsansätze** kurz vorgestellt. Dabei wird auch auf die entsprechende Literatur zur Vertiefung aufmerksam gemacht.

Vergleich von matrix- und graphbasierten Repräsentationsformen

TILSTRA ET AL. stellen in [553] eine matrixbasierte einer graphbasierten Repräsentationsform von einem Produktsystem gegenüber, um deren jeweiligen Vor- und Nachteile für bestimmte Anwendungsmöglichkeiten zu erarbeiten (s. Tabelle 16). Nach Auffassung von TILSTRA ET AL. HAT die Wahl zwischen der Verwendung einer **matrix- oder graphbasierten Darstellungsform** Auswirkungen auf die Art und Weise, wie die Produktmodelle erfasst, ausgewertet und genutzt werden. Im Rahmen des Vergleiches verwenden die Autoren als Vertreter für ein matrixbasiertes Modell die HDDSM (s. Bild 30 in Kapitel 3.3.1) sowie für ein graphbasiertes Modell den Component Flow Graph (CFG), welche sie jeweils selbst unabhängig voneinander entwickelt haben. Auf Grundlage ihrer Forschungsergebnisse haben sie einen Fragenkatalog erstellt, der bei der Auswahl eines geeigneten Produktrepräsentationsformats im Rahmen der Entwicklung unterstützt. Der erwartete Nutzen der matrixbasierten (vertreten durch HDDSM) und der graphbasierten (vertreten durch CFG) Repräsentationsform eines Produktmodells ist im Hinblick auf die drei wesentlichen Phasen der Produktmodelllebenszyklus Modellerstellung, Modellanalyse bzw. -bewertung und Modellnutzung und -kommunikation in Bild 137 dargestellt.

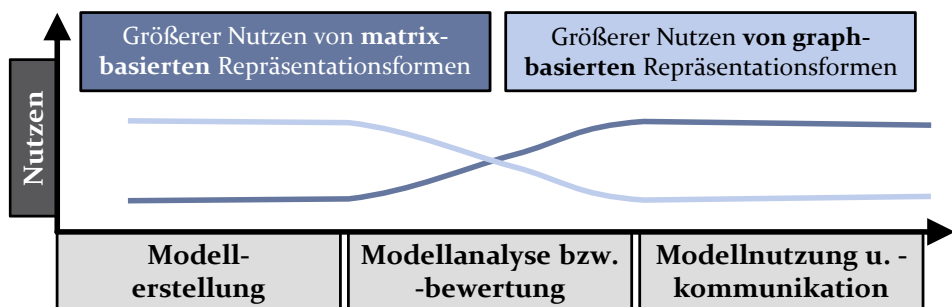


Bild 137: Nutzen von matrix- und graphbasierter Repräsentationsform (in Anlehnung an [553])

Tabelle 16: Vor- und Nachteile von matrix- und graphbasierten Repräsentationsformen (in Anlehnung an [553])

Modellphase	Matrixbasierte Repräsentationsform (z. B. HDDSM)	Graphbasierte Repräsentationsform (z. B. CFG)
Modellerstellung	+ Systematische Berücksichtigung von Elementpaaren und somit deren Relationen + Führt Modellersteller durch den Datenerfassungsprozess - Kann zu einer sehr hohen Anzahl an zu überprüfenden Elementpaaren führen	+ Erlaubt es Modellerstellern, leicht Ausnahmen von Regeln zu machen - Kann sehr schwierig sein, Relationen zwischen Knoten in einem großen Graphen zu lokalisieren, zu erfassen und zu modellieren
Modellanalyse bzw. -bewertung	+ Matrizen können mit einfachen mathematischen Analysemethoden analysiert werden - Aufgrund der schnell steigenden Größe der Matrizen ist die Verwendung fortschrittlicher Analysemethoden oft sehr zeitaufwendig	+ Muster in Graphen können leicht gefunden und automatisch manipuliert werden - Einfache Operationen, wie das Zählen von Ein- und Ausgängen, müssen von einem Suchalgorithmus ausgeführt werden
Modellnutzung u. -kommunikation	+ Matrizen sind für die visuelle Erkennung von Modulen und Strukturen nützlich - Verzweigungen sind in Matrizen schwer zu erkennen - Modifikationen, die an der Matrix auf der Basis der Bewertung vorgenommen wurden, sind oft schwer auf Grundlage der Matrixdarstellung zu erkennen und nachzuvollziehen	+ Das visuelle Layout des Graphen kann von einem Benutzer schnell geändert und manipuliert werden + Generative Grammatiken und Konfigurationstransformationen ermöglichen das automatische Einfügen oder Entfernen von Komponenten und Verbindungen

A.20 Grundlagen zur mathematischen Analyse

Zur Vervollständigung des Stands der Technik werden nachfolgend einige **Grundlagen zu mathematischen Analyseansätzen** kurz vorgestellt. Zur Weiteren Vertiefung ist jeweils die entsprechende Fachliteratur angegeben.

Kategorisierung von Clusterverfahren

Die Vielzahl verschiedener Clusterverfahren kann auf mannigfache Art und Weise kategorisiert werden. Die in Bild 138 dargestellte **Kategorisierung unterschiedlicher Arten von Clusterverfahren** stammt von GAN ET AL. aus der Grundlagenpublikation „Data Clustering“ [202].

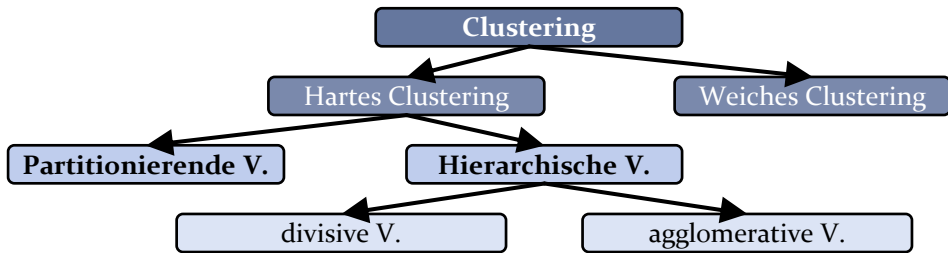


Bild 138: Kategorisierung von Clusterverfahren (nach [202])

Gemäß GAN ET AL. erfolgt die **Kategorisierung von Clusterverfahren** grundsätzlich in hartes und weiches Clustering [202]. Während hartes Clustering jedes Element bzw. jeden Datenpunkt in exakt ein Cluster einordnet, kann beim weichen Clustering ein Element bzw. ein Datenpunkt auch mehreren Clustern zugeordnet werden, was zu Clusterüberschneidungen führt. Innerhalb des harten Clustering wird zwischen partitionierenden und hierarchischen Verfahren differenziert. Partitionierende Verfahren (z. B. k-Means-Algorithmus) zerlegen die Menge an Elementen bzw. Daten in eine zuvor festgelegte Clusteranzahl, welches mind. ein Element und ein (Cluster-)Mittelpunkt zugeordnet wird. Hierarchische Verfahren gliedern sich in divisive (d. h. teilende) und agglomerative (d. h. zusammenschließende) Verfahren. Divisive Verfahren gehen Top-down vor und starten mit einem großen Cluster, das alle Elemente enthält, und spaltet dieses fortlaufend in immer kleinere Cluster auf. Agglomerative Verfahren gehen Bottom-up vor und starten mit Clustern, die aus je einem Element bestehen und schließen diese schrittweise zu größeren Clustern zusammen. Das Ergebnis dieser zwei Verfahren ist hierarchische Repräsentation der Elemente mit Hilfe von Clustern (s. für Details [168], [202], [166]).

Überblick über ausgewählte Clusterverfahren

Dieser Abschnitt gibt einen groben **Überblick über die Vielzahl von Clusterverfahren** mit ihren diversen Funktionsweisen, Besonderheiten und Anwendungsbereichen. In der nachfolgenden Übersicht in Bild 139 sind neben der Bezeichnung des Clusterverfahrens, einer knappen Erläuterung auch Literaturhinweise zur Vertiefung der Erklärung und/oder zur Anwendung oder Bewertung des jeweiligen Clusterverfahrens zu finden.

Cluster- verfahren	Erläuterung mit z. T. kurzer Interpretation, Bewertung	Ver- weis
k-Means- Algo- rithmus	▪ Verwendung für binäre Matrizen ▪ Anzahl k an Cluster muss am Anfang vorgegeben werden ▪ Wiederholte Durchführung der folgenden beide Schritte nach der Definition von k Clustermittelpunkte (CMP): Klassifikation aller Elemente zum nächsten CMP sowie Neuberechnung der CMP ▪ Anpassung des Verfahrens für DSMs, indem es nicht anhand des Abstands die Elemente einteilt sondern anhand der Relationen	[166], [41]
Expectation- Maximiza- tion (EM)	▪ Verbesserung von k-Means-Algorithmus ▪ Im Vergleich zum IGT-Algorithmus, Ermittlung besserer Lösungen durch weiches und flexibleres Clustering ▪ Wiederholte, abwechselnde Durchführung der beiden Schritte „Expectation“ und „Maximization“: Expectation berechnet für jedes Element, mit welcher Wahrscheinlichkeit es zu jedem Cluster gehört, Maximization verwendet diese Wahrscheinlichkeiten zur Neuberechnung der Verteilungsparameter	[166], [168]
Clustering- Algorithmus von ZAKARIAN	▪ Verwendung für binäre, nichtbinäre und gewichtete Matrizen ▪ Einsatz von separaten Algorithmen für $n \times n$ und $m \times n$ Matrizen ▪ Durchführung von Analysen für einfache und komplexe Matrizen möglich	[629]
IGT- Algorithmus	▪ Verwendung auch für numerische DSMs ▪ Benennung des IGT-Algorithmus nach seinen Entwicklern Idicula-Gutierrez-Thebeau-Algorithmus (IGTA) ▪ Überschneidung der Cluster sind möglich, da ein Element mehr als einem Cluster zugeordnet werden kann ▪ Basisalgorithmus des IGTA ist der sog. Bergsteigeralgorithmus, der sukzessive die gefundene Lösung bis zu einem lokalen (nicht globalen) Optimum verbessert. ▪ Integration als C- und Matlab-Code z.B. in eine Excel-Datei möglich	[269], [234], [548]
IGTA-plus	▪ Verbesserungen des IGT-Algorithmus ▪ Im Vergleich zum IGT-Algorithmus, Ermittlung besserer Lösungen mit geringerem Total-Cost-Wert v. a. bei großen Iteration ▪ Im Vergleich zum IGT-Algorithmus, computerunterstützte Anwendung ist konsistenter und schneller	[72], [73]

Cluster- verfahren	Erläuterung mit z. T. kurzer Interpretation, Bewertung	Ver- weis
Genetischer Algorithmus mit Nutzung d. Minimum Description Length	▪ Verwendung für das Clustering einer DSM ▪ Entwicklung der Methode auf Grundlage des genetischen Algorithmus (GA) mit Nutzung der Minimum Description Length ▪ Nutzung der DSM zur Visualisierung der Produktarchitektur und zur Entwicklung von Basis bildenden Blöcken von Elementen, die der Identifizierung von Produktmodulen dienen ▪ Anwendung erfordert eine relativ lange Laufzeit	[72], [585], [627], [628]
Hybrider genetischer Clustering- Algorithmus	▪ Verbesserung des vorangehenden Verfahren auf Basis des GA mit Nutzung der Minimum Description Length ▪ Einbindung von heuristischen Elemente in den GA Prozess zur Reduzierung der Laufzeit des Algorithmus und Verbesserung des Ergebnisses	[73]
Cuckoo Search (CS)	▪ Verwendung für das Clustering einer DSM zur Unterstützung der Modularisierung von Produkten ▪ Anwendung des Optimierungsalgorithmus Cuckoo Search, um die optimale Anzahl von Clustern und die optimale Zuordnung jedes Produktelements zu einem bestimmten Cluster zu finden, um den Gesamtkoordinationsaufwand zu minimieren ▪ Zielsetzung sind möglichst keine Überschneidungen von Cluster	[585]
Kombination von Neural Network Algorithmen und DSM	▪ Kombination von Algorithmen für neuronale Netze und DSMs ▪ Reorganisation der Produktelemente in Cluster, so dass die Interaktionen innerhalb der Cluster maximiert und außerhalb der Cluster minimiert werden ▪ Verwendung eines multikriteriellen Entscheidungsansatzes, um die Effizienz der verschiedenen vom neuronalen Netz abgeleiteten Clustering-Alternativen zu bewerten	[423], [585]
Systematic Clustering Methode	▪ Verwendung für organisationsbasierte, numerische DSMs ▪ Zielsetzung ist die Analyse der Auswirkungen von Überlappungen auf die Interaktionsstärke zwischen Teams, die diese überlappenden Aktivitäten durchführen ▪ Darstellung von Evolutionsgrad mittels einer Evolutions-DSM und von Sensitivitätsgrad mittels einer Sensitivitäts-DSM ▪ Messung der Interaktionsstärke zwischen Teams, die überlappende Aktivitäten durchführen, mit Hilfe eines quantitativen Modell auf Basis des Überlappungsverhältnisses, der relativen Koordinationsfrequenz sowie der Evolutions- und Sensitivitäts-DSM ▪ Vorgehen nach 2-stufige Clustering-Kriterien-Modell, bei dem das Clustering-Kriterium der 1. Stufe die Maximierung der addierten durchschnittlichen Interaktionsstärke der ausgewählten Organisationseinheiten und das Clustering-Kriterium der 2. Stufe die Minimierung der Gesamtkoordinierungszeit des PE-Projekts ist	[585], [625]

Bild 139: Übersicht über ausgewählte Clusterverfahren

Change Prediction Index zur Abschätzung der Änderungsweitergabe

Da der verbreitete und nützliche **Change Prediction Index (CPI)** sowohl im Analysemodul des Assistenzsystems zum Komplexitätsmanagement im Rahmen dieser Arbeit implementiert wurden als auch die Grundlage vieler der im folgenden Abschnitt beschriebenen Ansätze zur Analyse von Änderungen ist, wird dieser in diesem Abschnitt erläutert. Der CPI ist in der Publikation von SUH ET. AL. detailliert erläutert und wird nachfolgend gemäß dieser zusammenfassend und verkürzt skizziert [542].

Der CPI dient für einen ersten **Überblick bzw. eine grobe Abschätzung der Änderungsweitergabe einzelner Elemente** eines matrixbasierten Systemmodells. Darauf aufbauende und detailliertere Analysemethoden für die Änderungsverbreitung bzw. -fortpflanzung sind in Bild 140 skizziert.

Die **Berechnung des CPI eines Elements i** erfolgt gemäß Gleichung (20) (vgl. [542] auch für Ergänzungen zur Berechnung von entsprechenden Unsicherheiten). Der CPI-Wert eines Elements i gibt anhand der Relationen des Systems (unabhängig von deren Stärke) die Änderungspropagation an, die von diesem Element verursacht wird, wenn das System eine (externe) Veränderung (z. B. eine modifizierte Anforderung) erfährt. Dabei stellt ΔE eine binäre Änderungspropagation-Matrix dar, wobei $\Delta E_{j,i}$ bzw. $\Delta E_{i,k}$ eine Binärzahl ist, die angibt, ob das i -te Element aufgrund des Elements j bzw. k verändert wird (n ist die Anzahl aller Elemente). Dabei gibt die Aktiv- bzw. Reihensumme $\Delta E_{out,i}$ die Anzahl der Änderungen an, die von einem Element i ausgehen. Die Passiv- bzw. Spaltensumme $\Delta E_{in,i}$ gibt die Anzahl der Änderungen an, die ein Element i beeinflussen. Der CPI-Wert des Elements i ist die Differenz der Aktiv- und Passivsumme des Elements i .

$$CPI_i = \sum_{j=1}^n \Delta E_{j,i} - \sum_{k=1}^n \Delta E_{i,k} = \Delta E_{out,i} - \Delta E_{in,i} \quad (20)$$

Für die prinzipielle **Interpretation der CPI-Werte** werden die Elemente gemäß ihrem CPI-Wert einer der folgenden vier Kategorien zugeordnet:

- Verstärker (engl. multiplier) mit $CPI > 0$, d. h. verursachen mehr Änderungen als von anderen Elementen beeinflusst zu werden
- Absorber (engl. absorber) mit $CPI < 0$, d. h. Gegenteil v. Verstärker
- Träger (engl. carrier) mit $CPI = 0$, d. h. die Anzahl an beeinflussten und beeinflussenden Elementen ist identisch

Konstante (engl. constant) mit $CPI = 0$, wobei $\Delta E_{out,i} = \Delta E_{in,i} = 0$ ist, d. h. diese Elemente sind von den Veränderungen des Systems bzw. im System komplett unbeeinflusst.

Übersicht von Ansätzen zur Analyse von Änderungen

In Ergänzung zu der Übersicht über grundlegende Analysemethoden von Änderungen in Bild 40 werden nachfolgend weitere relevante Ansätze zur Analyse von Änderungen kurz vorgestellt. Dazu werden in Bild 140 zusätzlich zur Bezeichnung des Ansatzes, einer kurzen Erläuterung Literaturhinweise zur Vertiefung angeführt (vgl. auch die betreute studentische Arbeit [S54]).

Analyse- methode	Erläuterung mit z. T. kurzer Interpretation, Bewertung	Ver- weis
Change Fa- vorable Re- presentation (C-FAR)	▪ Verwendung von vorhandenen Produktinformationsmodel zur Unterstützung der Darstellung, Verbreitung und qualitativen Bewertung von Änderungen ▪ C-FAR nutzt das STEP-Datenmodell und das EXPRESS-Schema ▪ Bestandteile des Informationsmodells sind Entitäten, Relationen zwischen Entitäten und Attribute, die die Entitäten beschreiben, wobei C-FAR zur Unterstützung von Änderungen und deren Bewertung auf Attributebene dient ▪ Anwendung nur für einfache Produkte, da Methode relativ komplex	[110], [106]
Change Prediction Methode (CPM)	▪ Bekanntester Lösungsansatz zur Analyse der Änderungsweitergabe ▪ Ermittlung der jeweiligen Auftretenswahrscheinlichkeit und des Ausmaßes der Änderungsweitergabe zwischen zwei Elementen auf Grundlage von initialen Erfahrungswerten in zwei DSMs ▪ Produkt aus DSM „Direkte Wahrscheinlichkeit“ und DSM „Direkte Auswirkung“ ergibt DSM „Direktes Risiko“ und Risikograph und damit die Grundlage für die Berechnung der Produkt-Risiko-Matrix ▪ Unterstützung bei der Abschätzung des Risikos einer Änderungsausbreitung, um Auswirkungen auf (Teil-)Systeme oder mögliche entstehende Änderungskosten vorab besser kalkulieren zu können	[106], [107]
CPM and House of Quality (HoQ)	▪ Kombination von CPM und HoQ (nach der Methode Quality Function Deployment (QFD)), um die Auswirkungen der Änderungsweitergabe auf die Produktattribute zu bewerten ▪ Modellierung einer MDM bestehend aus den DSMs Produktkomponenten, Änderungsoptionen und Produkthanforderungen ▪ Unterstützung bei der Abschätzung wie gut jede Änderungsoption die Produkthanforderungen erfüllen kann, indem die potenzielle Ausbreitung von Änderungen berücksichtigt wird	[244], [308], [309]
Function- Behaviour- Structure Linkage Methode (FBS- Linkage)	▪ Erweiterung des CPM durch FBS, um durch die zusätzlichen Detail-ebenen die Gründe für die Weitergabe von Änderungen zu erhalten und ein proaktiveres Management von Änderungen zu ermöglichen ▪ Modifikation des (FBS)-Framework, so dass es für komplexe Produkte anwendbarer und zur Analyse der Änderungsmodellierung und -ausbreitung verwendbar ist ▪ Modifikationen umfassen insb. physikalische und technische Funktionen, die anstatt durch allgemeiner Verb-Objekt-Funktionen durch In- und Output-Relationen von Flüssen beschrieben werden ▪ Integration des modifizierten FBS-Framework in die CPM	[244]

Analyse- methode	Erläuterung mit z. T. kurzer Interpretation, Bewertung	Ver- weis
Change Prediction Index (CPI) als Teil des flexiblen Produktplatt- form-Design- prozess (FPDP)	- Integration 4 verschiedener Produktplattform Design-Methoden in den FPDP, welcher insgesamt 7 Phasen umfasst - Beachtung exogener Unsicherheiten und sog. flexibler Elemente durch den FPDP, wobei flexible Elemente als Elemente definiert sind, die den verschiedenen Anforderungen jeder Produktvariante durch Modifikation bei niedrigeren zusätzlichen Investitionsniveaus im Vergleich zu anderen einzigartigen Elementen, die denselben Zweck erfüllen können, gerecht werden - Messung des Grads der Änderungsausbreitung für ein einzelnes Element durch den Änderungsausbreitungsindex (CPI), der den Grad der physischen Änderungsausbreitung misst (ohne Beachtung der Stärke), der durch dieses Element verursacht wird, wenn das System in Folge einer externen Änderung beeinflusst wird - Berechnung des CPI als Differenz aus Aktiv- und Passivsumme, wobei der CPI-Wert zur Klassifizierung der Elemente dient	[143], [542], [505],
Methode zur Analyse von Änderungen in der Ent- wicklung	- Zielsetzung ist Aufdeckung der Änderungsweitergabe innerhalb und zwischen der funktionellen Anforderungsdomäne und der physikalischen Strukturdomäne (z.B. Komponenten) - Integration von Konzepten aus dem modellbasierten SE, der ontologie-, matrixbasierten und wissensbasierten Modellierung in eine Methode zur Analyse von Entwicklungsänderungen - Modellierung der kritischen Wechselwirkungen zwischen Funktionen und Komponenten mittels MBSE-Diagrammen - Analyse der Änderungsausbreitung und Identifikation von Entwurfskonflikten mittels matrixbasierter Analysemethoden, indem die Relationen zwischen Funktionen, Verhalten und räumlichen Anordnungen der Komponenten verfolgt werden - Formalisierung von Entwurfskonflikten auf der Grundlage einer vordefinierten Ontologie, wodurch das Schlussfolgern von Wissen (Reasoning) in der Wissensbasis möglich wird - Sammlung von Wissen über Entwicklungsänderungen aus früheren PE-Projekten - Identifikation geeigneter Lösungen aus der Wissensbasis mittels Reasoning und Bewertung potentieller Auswirkungen	[172], [244]
Multilayer network model (MNM) Mehrschich- tiges Netz- werkmodell	- Integration der Produkt-, Änderungs- und Organisationsdomäne in das MDM, wobei die Produktdomäne Komponenten, die Änderungsdomäne Änderungswünsche (change requests) und die Organisationsdomäne Personen umfasst - Management und Analyse der Änderungsausbreitung durch das MNM, welches verschiedene Werkzeuge und Metriken bereit stellt, inkl. der beiden neuen Ansätze Engineer Change Propagation Index (Engineer-CPI) und Propagation Directness (PD) - Quantifizierung der Leistung von Person bzgl. Ausbreitungseffekte ihrer Implementierung von Änderungen mittels des Engineer-CPI - Ermittlung der Anzahl der Produktschnittstellen, die sich aus einer Instanz der Eltern-Kind-Weitergabe ergeben, wird als PD benannt	[244], [424]

Bild 140: Ausgewählte Ansätze zur Analyse von Änderungen

A.21 Vorgehensweise zur Erstellung, Visualisierung und Analyse von matrixbasierten Modellen

Die vorhandenen **Vorgehensweisen zur Erstellung, Visualisierung und Analyse von matrixbasierten Modellen** variieren v. a. aufgrund deren verschiedener Zielsetzungen. Nachfolgend wird die Vorgehensweise nach LINDEMANN ET AL. [355], nach EPPINGER & BROWNING [161] und nach RINKEVICH & SAMSON [453] überblicksartig zusammengefasst (vgl. auch [378]).

Vorgehensweise nach EPPINGER & BROWNING

Die Vorgehensweise von EPPINGER & BROWNING [161] verfolgt als **Zielsetzung die Modellierung und Analyse der Systemarchitektur** und ist damit im Vergleich zum nachfolgenden Vorgehen nach LINDEMANN ET AL. etwas allgemeingültiger bzw. generischer und damit grundsätzlich umfassender. Die Vorgehensweise gemäß EPPINGER & BROWNING besteht aus den fünf in Bild 141 skizzierten Schritten, welche auf Grundlage von geeigneten Planungs- und Vorbereitungstätigkeiten bzgl. des Zwecks und Ziels sowie der Randbedingungen aufbauen. So erfolgt bspw. auf der Planung des Projekts sowie der Definition der Aufgaben- oder Problemstellung, wie z. B. der Neu- oder Anpassungskonstruktion eines Rohrs (s. Kapitel 1.5) im ersten Schritt Zerlegen (vgl. auch Bild 14 in Kapitel 2.3.2) die Aufteilung des PE-Projekts „Rohr“ (vgl. Bild 5 in Kapitel 1.5) in die einzelnen Domänen und Teilsysteme (z. T. über mehrere hierarchische Ebenen). So kann z. B. das Teilsystem „Rohrgeometrie“ in die einzelnen, geometrischen Merkmale, wie bspw. Innen- und Außendurchmesser und Länge, zerlegt werden (s. Bild 22 in Kapitel 3.3.1). Durch dieses Zerlegen wird das Systemverständnis i. d. R. deutlich verbessert. Im zweiten Schritt Identifizieren sind die zwischen den zuvor zerlegten Elementen bestehenden Relationen zu modellieren. So stehen der Innen- und Außendurchmesser des Rohrs wechselseitig zueinander in Beziehung (s. Bild 22), da der Außendurchmesser größer als der Innendurchmesser sein muss. Im Gegensatz dazu ist die Rohrlänge in diesem Beispiel von den zwei Rohrdurchmessern unabhängig. Im Rahmen der Analyse, dem dritten Schritt, werden die modellierten Systeme analysiert und die Elemente in eine zweckmäßige Reihenfolge angeordnet (s. auch Kapitel 3.3.3), um strukturelle Muster und deren Auswirkungen auf das Systemverhalten zu verstehen. Für das einfache Teilsystem „Rohrgeometrie“ ist es bspw. sinnvoll, die zwei in Abhängigkeiten stehenden Elemente Innen- und Außendurchmesser des Rohrs zusammenzustellen und die von beiden unabhängige Länge nicht dazwischen anzuordnen (s. Bild 22). Im vierten Schritt Darstellen bzw. Visualisieren sind hilfreiche

Repräsentation der DSM zu erstellen (s. Bild 34 und Kapitel 3.3.2), um im Hinblick auf die Zielsetzung besonders relevanten Eigenschaften des Systems geeignet darzustellen, wie z. B. alle Relationen zwischen den geometrischen Merkmalen des Rohrs und – falls von Interesse – auch zu den jeweils entsprechenden Prozessschritten. Im abschließenden Schritt Verbessern sind mögliche Verbesserungspotentiale abzuleiten. Als Ergebnis der Anwendung dieser Schritte liegt i. d. R. sowohl ein besseres Systemverständnis als auch viele Verbesserungsmaßnahmen für einzelne Teilsysteme sowie das Gesamtsystem vor. So kann im Hinblick auf das Rohrbeispiel die Länge unabhängig von den beiden Durchmessern angepasst werden und – auf Basis des Anhang A.4 – wird deutlich, welche geometrischen Merkmale wie genau mit den Rohreigenschaften und -verhalten zur Erfüllung der gewünschten Anforderungen in Beziehung stehen. Zudem sollte im Sinne eines kontinuierlichen Verbesserungsprozesses die Lessons learned und die gewonnenen Erkenntnisse nach dem fünften Schritt als Input in den ersten Schritt zurückfließen. In [161] sind eine Vielzahl an Anwendungsbeispielen zu finden, die sich an diesem Vorgehen in Bild 141 orientieren.

1	Zerlegen (engl. decompose)
2	Identifizieren (engl. identify)
3	Analysieren (engl. analyze)
4	Darstellen (engl. display)
5	Verbessern (engl. improve)

Bild 141: Vorgehen zur Modellierung und Analyse einer DSM (in Anlehnung an [161])

Vorgehensweise nach LINDEMANN ET AL.

Das Vorgehen nach LINDEMANN ET AL. [355] hat das gesamte **strukturelle Komplexitätsmanagement als zentrale Zielsetzung**. Hierfür beschreiben die Autoren den in Bild 142 dargestellten Prozess, der zum mit dem Vorgehen in Bild 141 einige Gemeinsamkeiten und Unterschied hat. In diesem wird in einem ersten Schritt die Aufgaben- oder Problemstellung definiert (z. B. Neu- oder Anpassungskonstruktion eines Rohrs s. Kapitel 1.5). Im zweiten Schritt ist das dafür erforderliche und zu modellierende System als MDM zu definieren. Die hierfür benötigten Daten und Informationen bzgl. der zu modellierenden Systemelemente und -relationen sind in einem dritten Schritt zu beschaffen, wodurch die direkten Abhängigkeiten des Systems bestimmt werden (vgl. Bild 22 in Kapitel 3.3.1). Als vierter Schritt

schließt sich der insb. für das strukturelle Komplexitätsmanagement wichtige Schritt der Ableitung der indirekten Abhängigkeiten an. Hierfür dient zur Veranschaulichung folgendes vereinfachtes Beispiel (s. Bild 5 in Kapitel 1.5 und Bild 22 in Kapitel 3.3.1): Durch das Prozesselement P1 „3D-CAD-Modell modellieren“ wird das Produktelement d_i „Rohrinnendurchmesser“ definiert und da dieses wiederum das Prozesselement P4 „Festigkeitsnachweis durchführen“ beeinflusst (i. S. v. als Input dient), sind die zwei Prozesselemente P1 und P2 indirekt voneinander abhängig. Im fünften Schritt erfolgt die Analyse der Systemstruktur, um u. a. Cluster zu identifizieren (s. Kapitel 3.3.3) und auch [55]. Die daraus gewonnenen Ergebnisse und Erkenntnisse dienen im Rahmen des strukturellen Komplexitätsmanagements zur Verbesserung der Systemstruktur. Die Schritte zwei bis sechs sind in jeweils einem Kapitel in [355] ausführlich beschrieben. Zudem werden in den beiden sich in [355] anschließenden Kapiteln einige dieser Schritte anschaulich mittels zweier Anwendungsbeispiele erläutert.

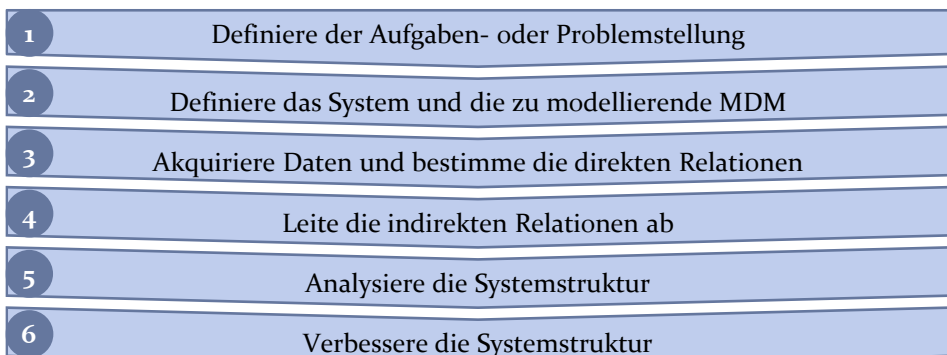


Bild 142: Vorgehen des strukturellen Komplexitätsmanagement (in Anlehnung an [355])

Vorgehensweise nach RINKEVICH & SAMSON

Das Vorgehen von RINKEVICH & SAMSON [453], welches u. a. auch JIANG ET AL. [279] (s. auch [278]) weitgehend identisch übernehmen, hat auch das **Ziel die komplexe PE insb. mit Hilfe von DSMs zu unterstützen**. Diese Vorgehensweise besteht aus den in Bild 143 dargestellten neun Schritten, welche zur Handhabung von allen DSM-Kategorien (s. Bild 23, s. auch Bild 38) verwendet werden können. Da dieses Vorgehens den vorhergehenden Vorgehensweisen ähnlich ist, wird auf eine detaillierte Erläuterung anhand des Anwendungsbeispiels Rohr verzichtet.

1	Definiere das System und des Randbedingungen
2	Liste alle Elemente (der relevanten Domänen) auf
3	Studiere die Relationen zwischen den Elementen
4	Modelliere eine Matrix zur Repräsentation der Relationen
5	Verifiziere die Matrix mit Hilfe von (Fach-)Experten
6	Analysiere die Matrix
7	Optimiere die Matrix mit Domänenwissen
8	Übergebe die Matrix mit den Ergebnissen an die Verantwortlichen
9	Diskutiere die Ergebnisse und implementiere diese sofern sinnvoll

Bild 143: Vorgehen zur Nutzung der DSM-Methode (in Anlehnung an [453])

Schlussfolgerungen aus diesen drei Vorgehensweisen

EPPINGER & BROWNING geben einen Überblick über die fünf prinzipiellen Tätigkeiten im Rahmen der matrixbasierten Modellierung und Analyse. Die einzelnen Schritte werden nicht detaillierter vorgestellt, so dass Nutzer nicht bei deren konkreten Anwendung unterstützt werden. LINDEMANN ET AL. erläutern das Vorgehen zur Handhabung von MDMs im Hinblick auf das gesamte strukturelle Komplexitätsmanagement sehr detailliert und ausführlich. Das Vorgehen nach RINKEVICH & SAMSON ist dagegen auch wieder allgemeingültiger und kann als eine geringfügige Konkretisierung des Vorgehens von EPPINGER & BROWNING angesehen werden. Daher lässt sich festhalten, dass sich zwar diese drei Vorgehensweisen etwas voneinander unterscheiden, deren **generelle Gemeinsamkeiten überwiegen**.

A.22 Ergänzungen zur Produktmodellierung

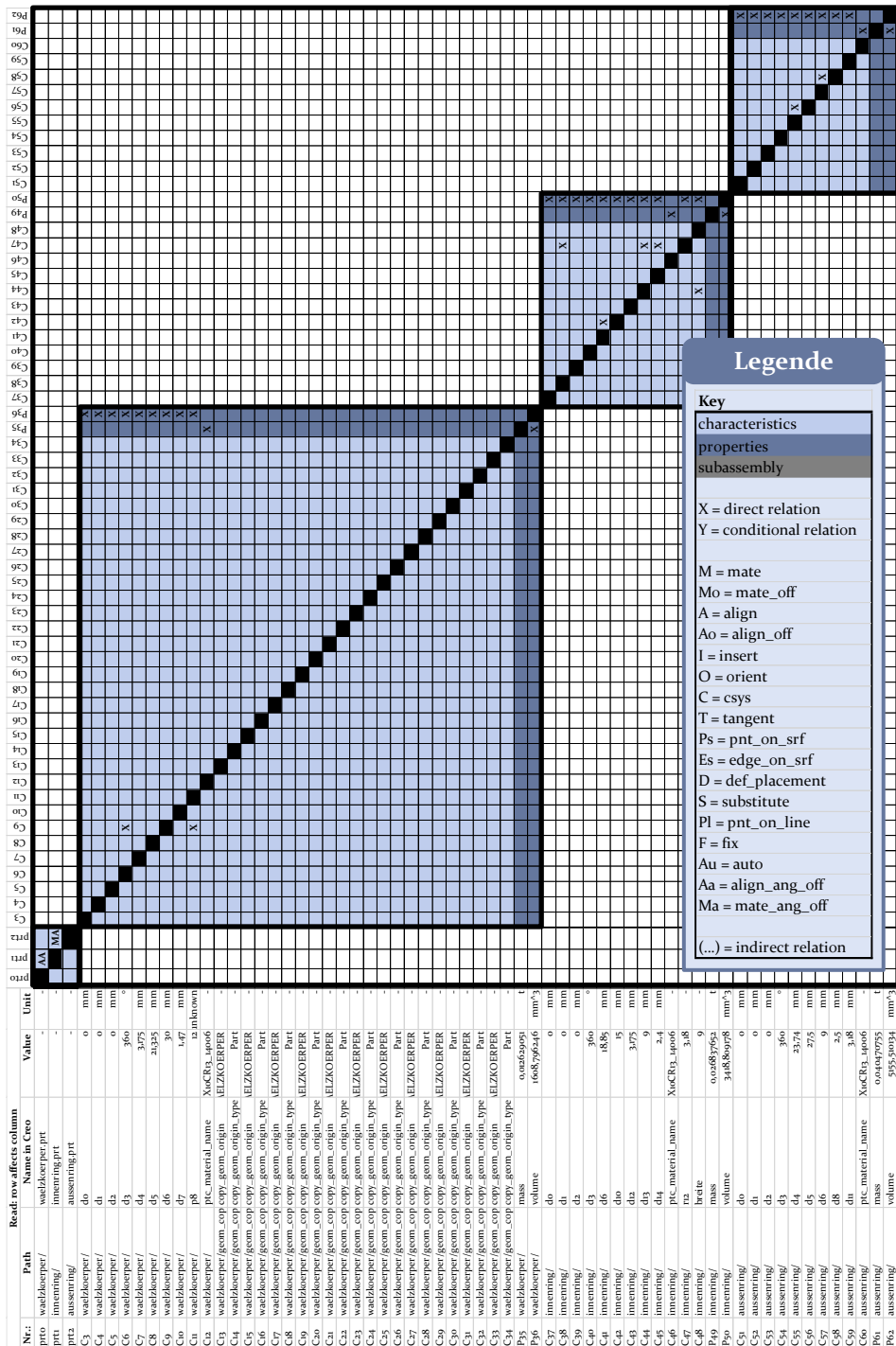
Im Folgenden werden **weitere ausgewählte Anwendungsbeispiele zur Demonstration der automatisierten Produktmodellierung mit Hilfe der IT-Schnittstelle** zwischen CAD-Modellen und dem Assistenzsystem dargestellt und kurz erläutert. Als Erstes wird das Anwendungsbeispiel aus Kapitel 6.1 für die matrixbasierte Produktmodellierung ergänzt. Danach werden weitere, automatisiert erstellte Produkt-DSMs kurz erläutert.

Modellierung einer DSM für eine einfache Baugruppe

In Kapitel 6.1 ist die Modellierung der DSM für ein einfaches Bauteil am Beispiel der Lagerhalbschale und für eine komplexe Baugruppe am Beispiel Getriebe aufgezeigt. Am Beispiel des im Getriebe verwendeten Rillenkugellagers wird die **Modellierung einer DSM für eine einfache Baugruppe** nachfolgend veranschaulicht und dabei weitere Besonderheiten erläutert.

Das gewählte **CAD-Modell des Rillenkugellagers** (s. Bild 144) ist auch stark vereinfacht und setzt sich aus den drei Einzelteilen Wälzkörper, Innenring und Außenring zusammen, welche jeweils als ein Einzelteil modelliert sind, um die Anzahl an Einzelteilen auf drei zu begrenzen. Die mit Hilfe der IT-Schnittstelle automatisiert erstellte DSM für das Rillenkugellager mit der Bezeichnung „*rillenkugellager_16006.asm*“ ist in Bild 144 abgebildet. Dabei sind in der ersten Spalte „Nr.“ wird der *elementtype* (s. Kapitel 5.2.1) und somit die Art des Parameters erfasst (z. B. „*prto*“, „*C3*“ und „*P35*“). In der Spalte „*Path*“ werden die Bauelemente angeführt, zu welchem der entsprechende Parameter gehört (z. B. „*P35*“ gehört zu „*waelzkoerper*“). Die in Creo vergebenen Bezeichnungen der Parameter werden in die Spalte „*Name in Creo*“, die dazugehörigen Werte in die Spalte „*Value*“ und Einheiten in die Spalte „*Unit*“ transformiert.

In der DSM_{Constraints} sind die **Relationen zwischen den drei Einzelteilen** zu finden. Die Wälzkörper sind zum Innenring mittels zwei *align*-Relationen (s. „*AA*“) ausgerichtet und die Lagebedingungen vom Innenring zum Außenring sind über die zwei Bedingungen *mate* („*M*“) und *align* („*A*“) festgelegt. Innerhalb des Einzelteils Wälzkörper beeinflusst bspw. der Parameter „*C6*“ den Parameter „*C9*“ über eine direkte Relation „*X*“. Dagegen sind in dieser DSM indirekte Relationen ausgeblendet. Im Einzelteil Wälzkörper ist ein großer Bereich ohne Relationen entstanden, da die Musterebene der gemusterten Wälzkörper bei der Akquisition von Beziehungen nicht berücksichtigt wird. Die auch automatisiert erstellte Legende zur Erklärung der Abkürzungen und Farben ist aus Platzgründen links unten eingefügt.



Modellierung einer DSM für die Baugruppe Korkenzieher

In diesem Abschnitt erfolgt die automatisierte Modellierung einer DSM für die etwas vereinfachte **Baugruppe Korkenzieher**, die aus den folgenden acht Einzelteilen besteht: Glocke, Spindel, Hebel, zwei Bolzen, zwei Hebel und zwei Sicherungsringe (DIN 471). Insgesamt umfasst diese Baugruppe rund 200 Bauelemente. Die auf Basis des CAD-Modells „Korkenzieher“ automatisiert erstellte DSM in Bild 145 hat daher neben der DSM_{Constraints} acht weitere DSMs für die jeweiligen Einzelteile. Dabei fällt auf, dass alle Einzelteile – bis auf die Ausnahme für die beiden DSMs für den Sicherungsring – im Hinblick auf die geometrischen Merkmale keine Relationen innerhalb der jeweiligen DSM des Einzelteils aufweisen. Zudem befinden sich auch keine Relationen außerhalb der farblich eingefärbten Zellen.

In der DSM_{Constraints} wird ersichtlich, dass bspw. das Einzelteil „glocke“ die Einzelteile „prt3“, „prt4“, „prt5“ über jeweils zwei direkte Relationen, das Einzelteil „prt6“ über zwei indirekte Relationen und das Einzelteil „prt7“ über eine direkte Relation „M“ und über zwei indirekte Relationen „(IM)“ beeinflusst. Die Legende für Bild 145 ist identisch der Legende in Bild 144. Außerdem wird sichtbar, dass die Einzelteile „prt1“ und „prt2“ mit keinem Einzelteil in Beziehung stehen, was möglicherweise auf ein Verbesserungspotential hindeuten kann.

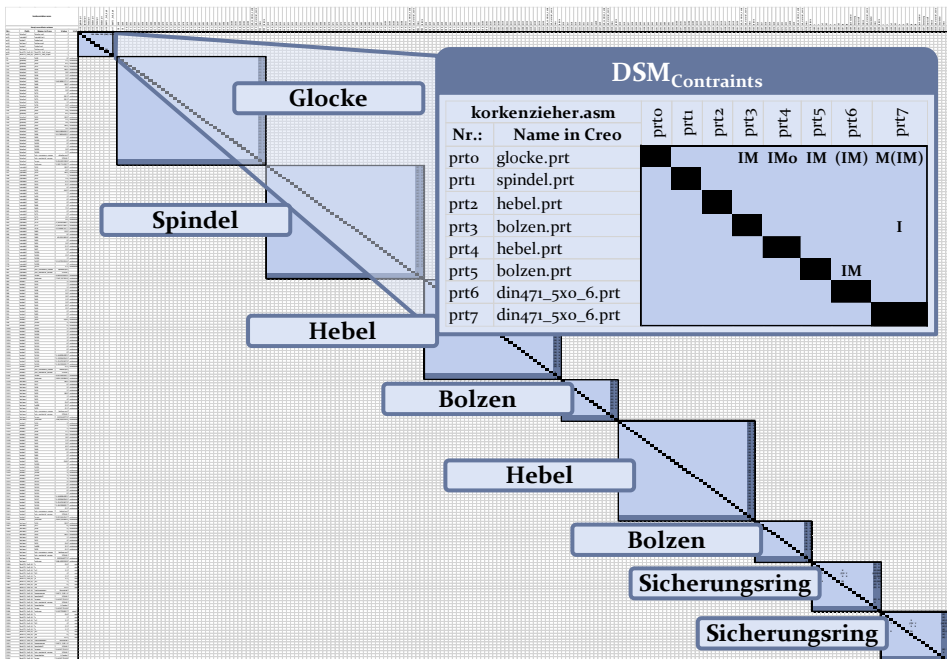


Bild 145: Automatisierte Modellierung einer DSM für die Baugruppe Korkenzieher

Die beiden DSMs für den Sicherungsring in Bild 145 sind identisch. Zur Veranschaulichung eines weiteren Ergebnisses einer automatisiert erstellten DSM wird die **DSM Sicherungsring** (DIN 471) in Bild 146 abgebildet. In dieser Darstellung sind einige direkte, bedingte Relationen als auch entsprechende indirekte und indirekt bedingte Relationen zu finden. Ferner wird ersichtlich, dass einige Parameter nur aufgrund deren Bezeichnung (s. in Spalte „Name in Creo“) nicht ohne weiteres interpretiert werden können. Daher sind entweder zweckmäßige und verständliche Bezeichnungen im CAD-Modell vorzunehmen oder DSMs für Einzelteile zusammen mit dem dazugehörigen CAD-Modell zu interpretieren. Außerdem werden vom Assistenzsystem auch Parameter (z. B. C208 bis C211) vom CAD-Modell in die DSM transformiert, die nicht für jeden Zweck hilfreich sind. Allerdings kann bspw. die Erfassung des Parameters „C211 Bearbeiter“ dazu verwendet werden, um zwischen der DSM für den Sicherungsring und einer weiteren DSM mit Mitarbeitenden aus der Domäne Organisation eine Verknüpfung zu modellieren.

korkenzieher.asm		C197	C198	C199	C200	C201	C202	C203	C204	C205	C206	C207	C208	C209	C210	C211	P212	P213
Nr.:	Name in Creo																	
C197	s																(X)	X
C198	x																(X)	X
C199	a2								X								(X)	(X)X
C200	d3			Y				(Y)Y	Y								(X)(Y)	(Y)X
C201	a							X	X								(X)	(X)X
C202	b																(X)	X
C203	d5		X	X					(X)								(X)	(X)X
C204	xb																(X)	X
C205	yb																(X)	X
C206	sachnummer																	
C207	benennung																	
C208	werkstoff																	
C209	masse																(X)	X
C210	ptc_material_name																X	
C211	bearbeiter																	
P212	mass																	
P213	volume																X	

Bild 146: Automatisierte Modellierung einer DSM für das Einzelteil Sicherungsring

Modellierung einer DSM für die Baugruppe Flugmotor

Abschließend wird zur Demonstration der automatisierten Modellierung der DSM mit Hilfe der IT-Schnittstelle zwischen CAD-Modellen und dem Assistenzsystem **als Anwendungsbeispiel ein vereinfachter Flugmotor** ausgewählt. Dieser besteht aus 29 Einzelteilen, wovon die Kolbengruppe drei dieser Einzelteile in eine Baugruppe gruppiert (s. Bild 147). Insgesamt

umfasst die automatisiert erstellte DSM für die Baugruppe Flugmotor exakt 1.047 Parameter und kann daher – wie in Bild 148 – nur als Ausschnitt in dieser Arbeit abgebildet werden.

In der DSM_{Constraints} in Bild 147 wird bei näherer Betrachtung offensichtlich, dass Einzelteil Motorgehäuse in der ersten Zeile der DSM_{Constraints} zu allen Einzelteilen und zur Baugruppe direkte oder indirekte Relationen aufweist. Außerdem wird ersichtlich (allein an automatisiert erstellten Spaltenbreite aufgrund der vielen Relationen), dass insb. die drei Einzelteile Propeller-nabe und -scheibe sowie die Sechskantmutter(n) (DIN 934) über relativ viele Relationen beeinflusst werden (in Bild 147 mit drei Pfeilen dargestellt).

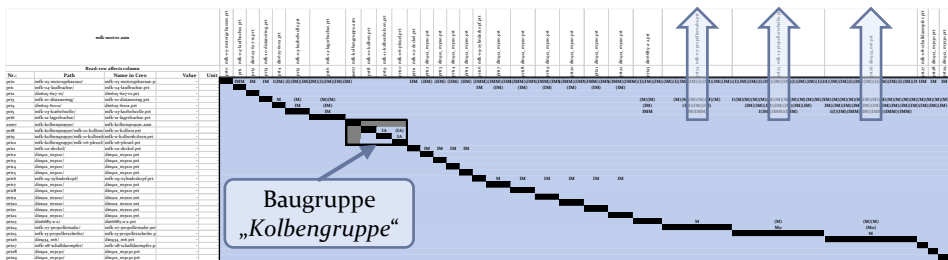


Bild 147: Automatisierte Modellierung der DSM_{Constraints} für die Baugruppe Flugmotor

Im Ausschnitt der automatisiert erstellten DSM für den Flugmotor, in dem nur 8 der 20 Einzelteile dargestellt sind sowie das sehr umfangreiche Einzelteil Motorgehäuse ausgeblendet ist, wird deutlich, dass selbst bei komplexen Produkten die DSM durch die gewählte Methode noch übersichtlich bleibt. Dies liegt daran, dass alle Relationen zwischen den Einzelteilen in den entsprechenden DSM_{Constraints} zusammengefasst sind.

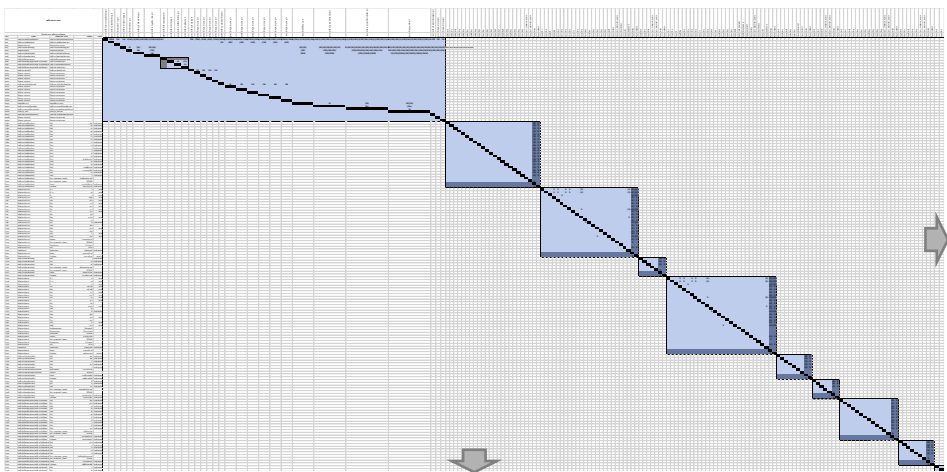


Bild 148: Ausschnitt der automatisiert erstellten DSM für die Baugruppe Flugmotor

A.23 Ergänzungen zur Prozessmodellierung

In diesem Abschnitt werden wesentliche **Grundlagen und Ergänzungen zur Prozessmodellierung** angeführt. Zuerst werden wichtige Grundlagen zu prinzipiellen Prozesselemente und -relationen überblicksartig erläutert. Darauf aufbauend sind wesentliche Grundlagen zu Modellierungssprachen zur Erstellung von Prozessmodelle zusammengefasst. Anschließend wird beispielhaft die Modellierungssprache Business Process Modeling Notation vorgestellt.

Grundlagen zu prinzipiellen Prozesselemente und -relationen

Im Rahmen der **Erstellung von Prozessmodellen dienen die vielen verschiedenen Relationen zwischen den Prozesselementen** auf der einen Seite zur Darstellung von Abhängigkeiten zwischen den Prozessschritten und auf der anderen Seite auch zur Beschreibung von Objekt-, Kontroll-, Steuerungs- und Informationsflüssen. Somit stehen zur Modellierung der Ablauflogik in Prozessmodellen insb. logische, zeitliche und technologische Relationen (i. S. v. Beziehungen, Bedingungen) zur Verfügung (s. [201] für tiefergehende Erläuterungen). Daher sind Konnektoren (i. S. v. logischen Verknüpfungsoperatoren) ein essenzieller Bestandteil von Prozessmodellen.

In den folgenden beiden Darstellungen in Bild 149 und Bild 150 ist eine **Übersicht von oft verwendeten Symbolen für Prozesselemente und -relationen** dargestellt. Diese zwei Übersicht Bild 149 und Bild 150 sind in Anlehnung an [307] erstellt (vgl. auch [96], [97], [98]). Die in dieser Übersicht verwendeten Bezeichnungen und Symbole der Elemente und Relationen sind in ähnlicher Art und Weise und auch mit einem ähnlichen Verständnis in den verbreiteten Modellierungssprachen, wie z. B. BPMN, EPK und UML, wiederzufinden. Da es allerdings zwischen Modellierungssprachen für Prozessmodelle im Hinblick auf die Notation, Syntax und Semantik (s. Anhang A.14) sowie auch auf die Metamodellierungssprache (s. Anhang A.15) Unterschiede gibt, ist eine definierte Prozessmodellierungssprache zu verwenden. Für die Realisierung einer möglichst unternehmensweit einheitlichen und allgemeingültigen Verständlichkeit von Prozessmodellen sind insb. standardisierte Modellierungssprachen hilfreich, die sich durch geeignete Softwareunterstützung (z. B. Modelio, Adonis) auch in formal-strukturierter Art und Weise modellieren lassen.

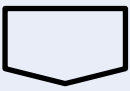
Bezeichnung	Symbol	Erläuterung
Elemente		
Start / Ende		Erstes /letztes Prozesselement i. S. v. Anfang / Ende des Prozessmodells
Ereignisse / Zustände		Ereignisse i. S. v. zeitbezogene Zustände; das Eintreten eines Ereignisses bzw. Erreichen eines Zustands hat eine bestimmte Folge
Funktion / Tätigkeit (Prozessschritt)		Durch eine Funktion / Tätigkeit wird an einem Objekt die Transformation eines bestimmten Inputs zu einen bestimmten Output durchgeführt (im engeren Sinne der Prozessschritt)
Informationsobjekt		Ein Informationsobjekt ist ein Element des Datenflusses (z. B. Auftrags- und Kundendaten)
Dokumente in Papierform		„Analoge“ Dokumente, wie z. B. in Papierform vorliegende Liste, Eingangsrechnung
Dokument in digitaler Form		„Digitale“ Dokumente, wie z. B. elektronische Bestellung, Datei auf dem Laufwerk u. im PDM
Datenspeicher (Persistenz)		Datenspeicherung im direkten Zugriff, wie z. B. eine IT-Schnittstelle zwischen PDM und ERP
Zusammengehörnde, digitale Dokumente		Digitaler Zwilling, elektronischer Schriftwechsel, wie z. B. E-Mail Verkehr zu einem Teilprozess (analog Dokumente in Papierform)
IT-System Anwendung		z. B. Prozessmodellierungssoftware, Assistenzsystem zum Komplexitätsmanagement
Organisationseinheit		Die Organisationseinheit gibt an, welcher Aufgabenträger oder welche Rolle bzw. Stelle eine Funktion übernimmt; dadurch erfolgt die Klärung von Zuständigkeiten in einem Prozess
Sprungmarke (beide Darstellungsformen sind üblich)		Prozessmodelle, die sich über mehrere „Seiten“ erstrecken, werden über Sprungmarken mit Seitenzahl bzw. Pfadlinks verbunden
Verweis auf Teilprozess / Prozessschritt		Das Symbol dient auch als Verweis auf einen an anderer Stelle dokumentierten Teilprozess / Prozessschritt; die Kennzeichnung erfolgt mit Seitenzahl, Pfadlink oder Teilmodellnamen

Bild 149: Übersicht der oft verwendeten Symbole für Elemente der Prozessmodellierung

Bezeichnung	Symbol	Erläuterung
Relationen		
Konnektoren / Verzweigungs- stellen		Exklusive-ODER-Verzweigung: Nur einer der folgenden Teilprozesse / Prozessschritte kann eintreten
		ODER-Verzweigung: Einer, mehrere oder alle folgenden Teilprozesse / Prozessschritte können eintreten
		UND-Verzweigung: Alle folgenden Teilprozesse / Prozessschritte müssen eintreten
Flusslinie		Diese Relationen dienen der Darstellung von Abhängigkeiten zwischen Prozessschritten und der Beschreibung von Objekt-, Kontroll-, oder Steuerungsflüssen, z. B. logische, zeitliche und technologische Relationen
Datenfluss		Mit dieser Relation wird v. a. der Wissens-, Informations- und Datenfluss dargestellt
Verbindung der Organisations- einheit		Mit dieser Relation werden die Organisations- einheiten mit den entsprechenden Elementen verbunden

Bild 150: Übersicht der oft verwendeten Symbole für Relationen der Prozessmodellierung

Grundlagen zu Modellierungssprachen für Prozessmodelle

Sowohl in der wissenschaftlichen Forschung als auch in der industriellen Praxis stehen eine Vielzahl von Modellierungssprachen zur Erstellung von Unternehmens- bzw. Prozessmodellen zur Verfügung. Dabei lassen sich Modellierungssprachen zum einen in domänenübergreifende und in domänenspezifische Sprachen (s. Anhang A.12) und zum anderen in Skript- und Diagrammsprachen gliedern. Da Skriptsprachen gute Programmierkompetenzen voraussetzen, sind **Diagrammsprachen zu reinen Prozessmodellierung** deutlich verbreiteter. Ausgewählte Diagrammsprachen sind gemäß GADATSCH in Bild 151 nach datenfluss-, kontrollfluss- und objektorientierten Ansätzen gegliedert dargestellt (s. [201]):

- Mit datenflussorientierten Ansätzen wird v. a. der Wissens, Informations- und Datenfluss zwischen den verschiedenen Prozessschritten und entsprechenden Tätigkeiten modelliert

- Mit kontrollflussorientierten Ansätzen werden v. a. die zeitlich-logischen Abhängigkeiten und damit die zeitliche Abfolge der einzelnen Prozessschritte modelliert
- Mit objektorientierten Ansätze werden v. a. Personen, Sachmittel, Dokumente oder Ereignisse als Objekte, die mit Attributen charakterisiert werden können, als In- und Output für Prozessschritte modelliert.

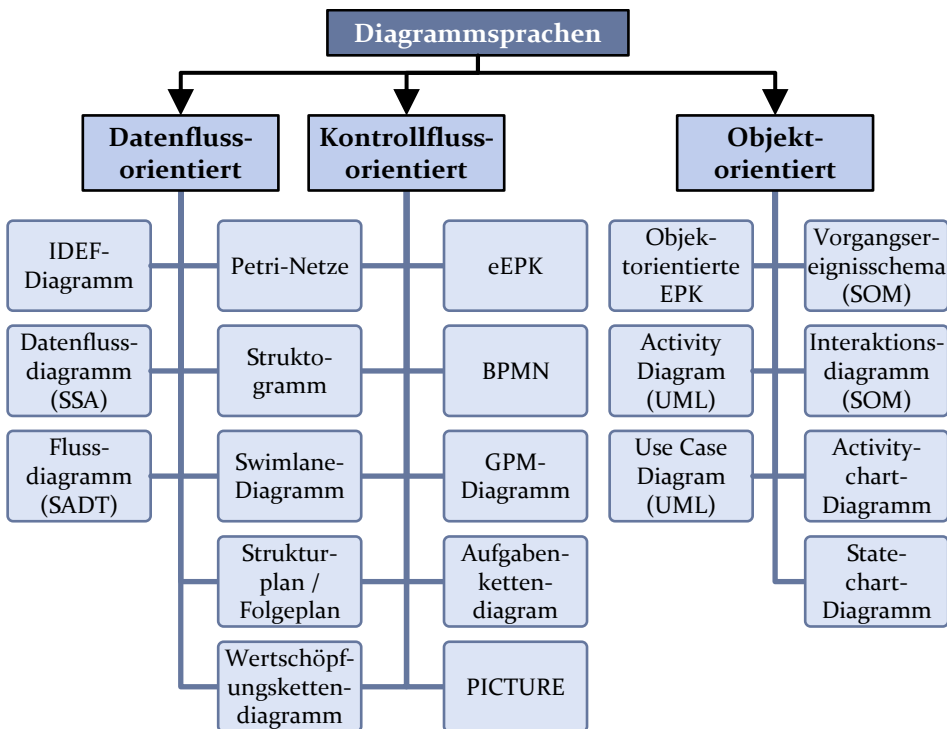


Bild 151: Übersicht ausgewählter Diagrammsprachen (gemäß [201])

Eine **empirische Befragung** von 219 Unternehmen und Institutionen im deutschsprachigen Europa hat u. a. gezeigt, dass 63% der Befragten Flussdiagramme, 49% BPMN, 47% EPK und 28% UML zur Modellierung von Prozessen verwenden (Mehrfachnennungen bei dieser Befragung waren möglich) [389]. Allen anderen Modelliersprachen wurden nur von bis zu 3% der Befragten genannt.

Die Verwendung von Flussdiagrammen erfolgt häufig ohne eine definierte Notation, Syntax und Semantik (vgl. Anhang A.14), die für die jeweilige Zielsetzung und Aufgabe individuell festgelegt werden. Somit lassen sich unter Flussdiagramme alle nicht formalisierten Modellierungsmethoden und -sprachen subsumieren, die aus z. B. aus beliebigen Bestandteilen von Diagrammen bestehen (vgl. [201]). Auch wenn Flussdiagramme für die Modellierung von komplexen Systemen hilfreich sein können, ist die **Verwendung von international standardisierten und formalisierten Modellierungssprachen** hilfreicher und auch für die Zielsetzung in dieser Arbeit zweckdienlicher. Dies liegt u. a. daran, dass auf Basis der allgemein definierten Notation, Syntax und Semantik (vgl. Anhang A.14) sowie dazugehöriger Referenzmodelle und Metamodelle (s. Anhang A.15) diese Modellierungssprachen allgemeingültig verständlich sind. Außerdem kann in Folge der einheitlichen Formalisierung auch die Modellierung, Visualisierung und Analyse besser mit entsprechenden IT-Werkzeugen unterstützt werden. Zudem sind formalisierte Prozessmodelle bspw. durch das in dieser Arbeit vorgestellte Assistenzsystem besser als z. B. individuelle Flussdiagramme nutzbar. Zu den verbreiteten Modellierungssprache der oben skizzierten Umfragen gehören insb. BPMN, EPK und UML.

Vorstellung der Business Process Modeling Notation

Die Modellierungssprache **Business Process Model and Notation (BPMN)** ist in der industriellen Praxis stark verbreitet, ist gegenwärtig in der Version BPMN 2.0 international standardisiert und dient v. a. der visuellen Modellierung von Prozessmodellen. Dabei liegt der Fokus der BPMN auf der **Prozessgestaltung i. S. des zeitlich-logischen Ablaufs** von Prozessschritten (i. S. v. Tätigkeiten und Aufgaben). Im Gegensatz dazu wird die Modellierung von Konstrukten u. a. der Domänen Produkt, Wissen und Organisation weitgehend nicht unterstützt [40]. Die in der BPMN erstellten Diagramme werden oft im *.xml*-Dateiformat gespeichert.

Für eine detaillierte Erläuterung der Notation, Syntax und Semantik sowie zur Vertiefung der Anwendung und für Anwendungsbeispiele wird u. a. auf [40], [192], [201] verwiesen. Nachfolgend werden lediglich die **Basiselemente der BPMN** vorgestellt (s. Bild 152), die in folgende fünf Kategorien unterteilt werden können (vgl. [40], [192], [201]):

- Flussobjekte: Flussobjekte stellen den Prozess (Aktivitäten) dar, der ggf. bestimmte Bedingungen erfordert (Gateways) und zu einem bestimmten Zustand führt (Ereignisse),

- Verbindende Objekte: Verbindende Objekte verbinden Flussobjekte mittels Sequenzflüsse, Nachrichtenflüsse und Assoziationen,
- Pools und Lanes: Pools umfassen die zu einem Prozess gehörenden Lanes, die Prozessschritte Aufgabenträger zuordnen,
- Artefakte: Artefakte ergänzen das Prozessmodell um Informationen und werden mittels Assoziationen mit Flussobjekte verbunden,
- Daten: Daten dienen der Darstellung von In- und Output und werden mittels Assoziationen mit Flussobjekte verbunden.

Kateg.	Bezeichnung	Symbol	Erläuterung
Flussobjekte	Aktivitäten	 	Aktivitäten stellen den Prozess i. e. S. dar und können in zusammengesetzte und elementare Aktivitäten differenziert werden; Unterprozesse setzen sich aus mind. zwei Arbeitsschritten zusammen und werden mit einem „+“ repräsentiert
	Gateways	 XOR  AND  OR	Gateways stellen Entscheidungen in Form von Entscheidungspunkten dar, die z. B. in Prozessablaufalternativen und Prozesszusammenführungen resultieren
	Ereignisse	 S  Z  E	Ereignisse stellen Zustandsänderungen dar, die den Ablauf von Prozessen beeinflussen; unterscheiden lassen sich das Start- (S), Zwischen- (Z) und Endereignis (E)
Verbindende Objekte	Sequenzfluss		Der Sequenzfluss stellt den Ablauf von Flussobjekten innerhalb eines Pools dar
	Nachrichtenfluss		Der Nachrichtenfluss stellt den Austausch von Nachrichten zwischen Akteuren aus verschiedenen Pools dar
	Assoziation		Mittels der Assoziationen werden beliebige Artefakte an Flussobjekte annotiert

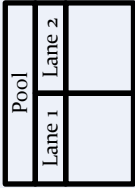
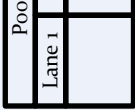
Kateg.	Bezeichnung	Symbol	Erläuterung
Pools und Lanes	Pools		Pools stellen einen eigenständigen Prozess dar und umfassen dessen Grenzen von Anfang bis Ende
	Lanes		Lanes stellen die Aufgabenträger einzelner Teilprozesse eines Pools dar und zwar z. B. differenziert nach Organisationseinheiten, Stellen, Rollen oder IT-Systemen
Artefakte	Anmerkung		Durch Anmerkungen können beliebige Texte an alle Basiselemente annotiert werden und dadurch die Dokumentation des Modells verbessert werden
	Gruppierung		Durch Gruppierungen können beliebige Basiselemente zusammengefasst werden und dadurch die Dokumentation und Analyse des Modellen vereinfacht werden
	Eigene Symbole		Durch eigene Symbole besteht die Möglichkeit zusätzliche Notationen zu verwenden
Daten	Datenobjekt		Datenobjekte dienen der Darstellung von Input und Output der Aktivitäten und werden durch Assoziationen mit den Flussobjekten verbunden
	Dateninput		
	Datenoutput		
	Daten-speicher		

Bild 152: Übersicht der Basiselemente von BPMN (in Anlehnung an [40], [192])

Grundlagen zu Modellierungssoftware für Prozessmodelle

Für die Prozessdomäne hat sich gezeigt, dass grundsätzlich alle Prozesse auf verschiedenen Hierarchieebenen computerunterstützt in das matrixbasierte Prozessmodell zu übertragen sind, da nur dadurch ein entwicklungsbegleitendes, ganzheitliches Komplexitätsmanagement sinnvoll möglich ist (z. B. Iterationsmanagement). Untersuchungen von IT-Werkzeugen zur Prozessmodellierung in der Industrie haben gezeigt, dass vor allem auf den Prozessmodellierungssprachen BPMN, EPK und UML basierte Programme Anwendung finden (s. Bild 153). Daher sind v. a. IT-Schnittstellen zwischen

Prozessmodellierungstools und der matrixbasierten Prozessbeschreibung zweckdienlich. Da der Entwicklungsaufwand für die **IT-Schnittstellen zu den verbreiteten Prozessmodellierungstools** vom Autor dieser Arbeit als vergleichbar eingeschätzt wird, ist in Kapitel 5.2.2 die automatisierte Modellierung der DSM-Prozess am Beispiel ADONIS erläutert. In Bild 153 ist eine Übersicht von Prozessmodellierungssoftware mit den jeweils unterstützten Modellierungssprachen, der Kundenanzahl sowie der Information, ob diese Software von Autor verwendet wurde, dargestellt.

Software	Anbieter	Kunden	BPMN	EPK	UML
ADONIS (verwendet)	BOC Group	1.000 bis 10.000	x	x	x
Aeneis (verwendet)	Intellior	1.000 bis 10.000	x		
ARIS (verwendet)	Software AG	über 10.000	x	x	x
BIC Platform	GBTEC / arvato Systems	100 bis 1.000	x	x	x
BPM inspire	Inspire	100 bis 1.000	x		x
ConSense IMS QMS PMS	ConSense	100 bis 1.000	x	x	
Enterprise Architect (verwendet)	Sparx Systems	nicht bekannt			x
iGrafx Suite (verwendet)	iGrafx	über 10.000	x		
Modelio (verwendet)	Modeliosoft	nicht bekannt	x		x
Process Modeler for MS Viso	itp commerce	1.000 bis 10.000	x		
ViFlow	ViCon	1.000 bis 10.000	x	x	

Bild 153: Übersicht ausgewählter Prozessmodellierungssoftware (in Anlehnung an [134])

Vorstellung der Prozessmodellierungssoftware ADONIS

ADONIS ist eine Software **für Prozessmodellierung und -management**, die von der BOC Information Technologies Consulting GmbH (BOC) in Wien entwickelt wurde. Die BOC ist eine Ausgründung der Universität Wien, die neben ADONIS auch ADOit, einer Lösung für IT-Architekturmodellierung und -management, vertreibt. Diesen beiden IT-Lösungen liegen zum einen das grundlegende Begriffsverständnis von Modellierungsmethode und -sprache nach Karagiannis (s. Anhang A.14) und zum anderen das Begriffsverständnis von Metamodell nach Karagiannis (s. Anhang A.15) zugrunde.

ADONIS stellt Anwendern einen sehr großen Funktionsumfang zur Verfügung. Die Software umfasst Module zur Prozessmodellierung, -analyse, -evaluation, -simulation, -ausführung und -überwachung (s. für Details [134]). Außerdem werden die wesentlichen Modellierungssprachen und deren Notationen, wie insb. BPMN 2.0, EPK und UML unterstützt (s. Bild 153). **ADONIS** unterstützt verschiedene, domänenspezifische Fachexperten, insb. Prozessexperten, und Geschäftsprozessanalysten zur Optimierung der Unternehmensprozesse.

Für die Prozessmodellierungssoftware **ADONIS stehen unterschiedliche Versionen bereit**. So lassen sich einige kommerzielle sowie eine kostenlose Version (Community Edition) auf der Homepage von BOC finden. Alle Varianten unterstützen die Modellierung von BPMN-Prozessmodellen sowie die wesentlichen Funktionalitäten. Im Vergleich zur kommerziellen hat die kostenlose Variante einige funktionale, technische und organisatorische Einschränkungen.

ADONIS bietet einige **IT-Schnittstellen zum Import und Export**. Jedoch kann in der kostenlosen Variante nur das ADONIS eigene Format „ADL“ genutzt werden. Daher müssen *.bpmn*-Dateien online konvertiert werden. Im Gegensatz dazu können in der kommerziellen Variante *.bpmn*-Dateien direkt importiert oder exportiert werden.

Da u. a. die Studie zu IT-Werkzeugen zum Geschäftsprozessmanagement in [134] gezeigt hat, dass **ADONIS eine der führenden Prozessmodellierungssoftware** ist, hat sich der Autor für ADONIS als ein Anwendungssystem für die Prozessdomäne entschieden. Als eine weitere Modellierungssoftware wurde in seinen Forschungsarbeiten v. a. auch die Software Modelio und darüber hinaus auch Aeneis und Aris u. a. auch für die Prozessmodellierung genutzt.

A.24 Ergänzungen zur Visualisierung

In diesem Anhang sind wesentliche **Ergänzungen zur Visualisierung** zu finden. Im ersten Abschnitt Ergänzungen zum graphbasierten Visualisierungsmodul des Assistenzsystems und im zweiten Abschnitt Ergänzungen zum Visualisierungsprogramm Tulip zusammengefasst.

Ergänzungen zum graphbasierten Visualisierungsmodul

Im **graphbasierten Visualisierungsmodul des Assistenzsystems** sind gegenwärtig die drei Visualisierungsstile *Value*, *Level* und *Correlation* als auch die fünf Graphlayouts *hierarchisch*, *kreisförmig*, *radialer Baum*, *kräftebasiert* und *spektral* verfügbar. Darüber hinaus lassen sich aufgrund des modularen Aufbaus des Assistenzsystems und der weiter zunehmenden Vielfalt an Python-Bibliotheken bei Bedarf auch weitere Visualisierungsmöglichkeiten implementieren. Für vertiefende und weitere im Rahmen der Forschungsarbeiten des Autors entwickelte Visualisierungsansätze s. insb. die beiden Publikationen [P32], [P34] sowie die von ihm betreuten, studentischen Arbeiten [S8], [S16], [S23], [S24], [S25].

Bei allen Visualisierungsstilen des graphbasierten Visualisierungsmoduls wird neben der jeweils spezifischen Legende und diverser Optionen auch eine tabellarisch strukturierte **Liste mit kontextspezifischen Inhalten** dem Anwender zur Verfügung gestellt. In dieser Liste sind in Abhängigkeit der vorgenommenen Einstellungen zu den jeweils relevanten Elementen bspw. deren *Kurzbezeichnung* bzw. *ID*, *Path*, *Name*, *Value*, *Unit* sowie die *Aktiv-* und *Passivsumme* dargestellt. Zudem sind in Detailansichten weitere, hilfreiche Informationen (Attribute) zu Elementen und Relationen zu finden, wie z. B. ein- und ausgehenden Relationen von einem selektierten Element sowie deren Art und Stärke. Dadurch wird die Lesbarkeit und Übersichtlichkeit benutzerspezifisch und je nach Aufgaben- und Problemstellung verbessert.

Aufgrund der stark eingeschränkten Lesbarkeit von zwei der drei in Bild 83 in Kapitel 6.4.1 dargestellten **Visualisierungsstile werden diese beiden nachfolgend vergrößert** abgebildet. So ist in Bild 154 der Visualisierungsstil *Level* und in Bild 155 der Visualisierungsstil *Value* und dargestellt. Die Erläuterungen zu den beiden Visualisierungen sind in Kapitel 6.4.1 zu finden.

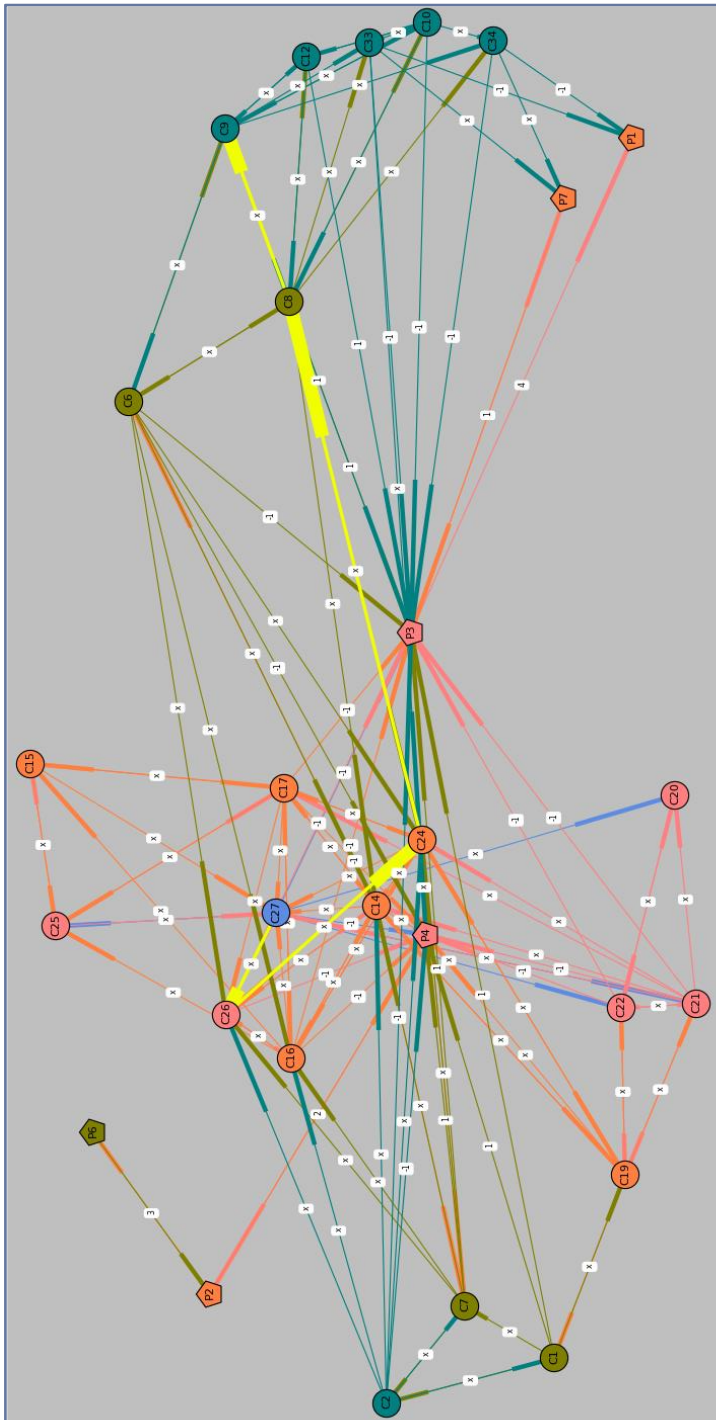


Bild 154: Visualisierungsstil *Level* im Graphlayout *Spectral* mit fünf *Levels*

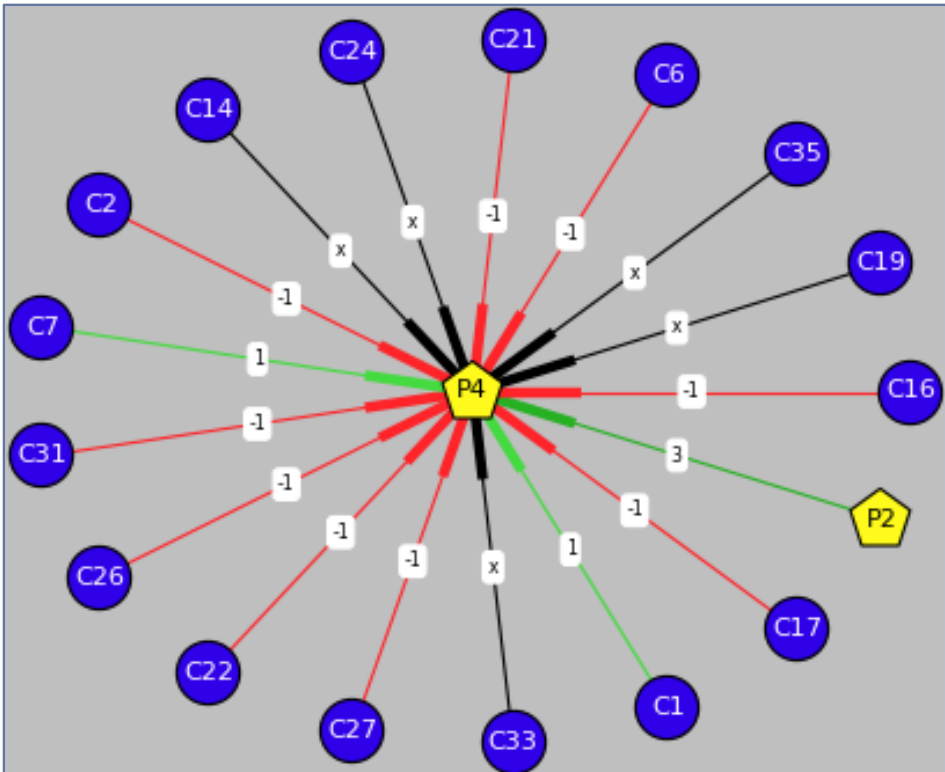


Bild 155: Visualisierungsstil *Value* im Graphlayout *Circle* mit der Option *Reverse Direction* (in Anlehnung an [S23])

Ergänzungen zum externen Visualisierungsprogramm Tulip

Tulip ist eine wissenschaftliche Visualisierungssoftware, die ursprünglich an der Universität Bordeaux entwickelt wurde [25]. Unter Tulip verstehen die Autoren ein **Framework zur Informationsvisualisierung**, das sich insb. zur Analyse und Visualisierung von relationalen Daten eignet [25]. Demgemäß wird mit Tulip das Ziel verfolgt, Entwicklern eine umfassende Bibliothek zur Verfügung zu stellen, die für den Entwurf von interaktiven Informationsvisualisierungsanwendungen für v. a. relationale Daten dient (z. B. Entwicklung von Algorithmen, visuellen Kodierungen, Interaktionstechniken, Datenmodellen und domänenspezifischen Visualisierungen) [25]. Somit soll Tulip als Framework – ähnlich wie auch Gephi – Entwickler unterstützen selbst Anwendungen zu erstellen, die auf die anwenderspezifischen Aufgaben- und Problemstellungen zugeschnitten sind [488].

Das quellcodeoffene Programm Tulip ist in der Programmiersprache C++ implementiert [25]. Auf Basis der Tulip-Python-Bibliothek, d. h. der Vielzahl von Tulip-Python Paketen, wird annähernd der gesamte Inhalt der Tulip C++ API in Python verfügbar. In dieser Arbeit sind u. a. die Pakete *tulip-python*, *tulipgui-python* und *tulipogl-python* verwendet worden. Neben C++ können daher **Modifikationen und Erweiterungen von Tulip auch in Python** von Entwicklern und Anwendern vorgenommen werden.

Tulip ist **die Visualisierung von Modellen äußerst komplexer Systeme geeignet**. So lassen sich mittels Tulip in Abhängigkeit der verwendeten Hardwaresystems mir rund zehn Millionen Elemente darstellen [488]. Insgesamt sind in Tulip über 40 verschiedene Graphlayouts verfügbar, wovon v. a. die sechs hierarchischen und zwölf stärkebasierten Layouts für diese Arbeit besonders hilfreich sind. Diese Graphlayouts können mit Hilfe von verschiedenen Sichten (sog. Panels) je nach Zielsetzung unterschiedlich betrachtet werden. Zudem sind auch dreidimensionale Visualisierung möglich. Im Gegensatz zu anderen Visualisierungssoftwaresystemen kann Tulip auch Adjazenzmatritzen darstellen. Ein weiterer Vorteil von Tulip ist, dass relativ einfach der kürzeste Pfad zwischen zwei selektierten Elementen ermittelt und grafisch abgebildet werden kann.

Tulip stellt neben einer Vielzahl von Visualisierungsmöglichkeiten auch viele **Funktionalitäten zur Erstellung, Manipulation und Analyse von Graphen** zur Verfügung (s. für Details [25], [488]). So kann z. B. die Farbgebung von Elementen und deren Relationen sowohl manuell als auch computerunterstützt auf Basis zuvor definierter Kriterien erfolgen. Zudem stehen verschiedene Funktionen zum Suchen und Filtern von Elementen und Relationen zur Verfügung, die auch nutzerspezifisch angepasst werden können. Außerdem stellt Tulip auch Möglichkeiten zur hierarchischen Navigation in Graphen bereit. So lässt sich in Tulip analog der Systemhierarchiemodellierung in Bild 30 ein Graph in seine einzelne Subgraphen (i. S. v. Clustern) z. B. für eine bessere Übersichtlichkeit aufteilen (vgl. auch Vorgehensweise zur systematischen Zerlegung von DSMs in Submatrizen in Bild 31). Dabei können diese Subgraphen mit denselben Funktionalitäten wie Graphen visualisiert und manipuliert werden.

Tulip unterstützt **alle für das Assistenzsystem erforderlichen Import- und Exportformate**. Als Importformate können neben dem Tulip-Format v. a. auch .csv-Dateien und .*graphml*-Dateien vom Visualisierungsprogramm verwendet werden. Als Exportformate stehen insb. GraphML- und Tulip und zudem u. a. auch Jpeg, BMP, PNG und Tiff zur Verfügung. Für weitere Details und Erläuterungen siehe die Homepage [25].

A.25 Ergänzungen zur Analyse

Nachfolgend sind **Ergänzungen zur Analyse** und Bewertung zu finden. Zuerst wird das Fachkonzept für die Implementierung des Clustering mit dem ZAKARIAN-Algorithmus vorgestellt. Danach werden zusätzliche Anwendungsbeispiele für verschiedene Module zur Analyse dargestellt.

Fachkonzept für das Clustering mit dem ZAKARIAN-Algorithmus

Im Aktivitätsdiagramm in Bild 156 ist das **Fachkonzept für das Clustering mit dem ZAKARIAN-Algorithmus** dargestellt. Dieses ist gemäß dem Grundschemata bzw. Pseudocodes von ZAKARIAN in [629] erstellt.

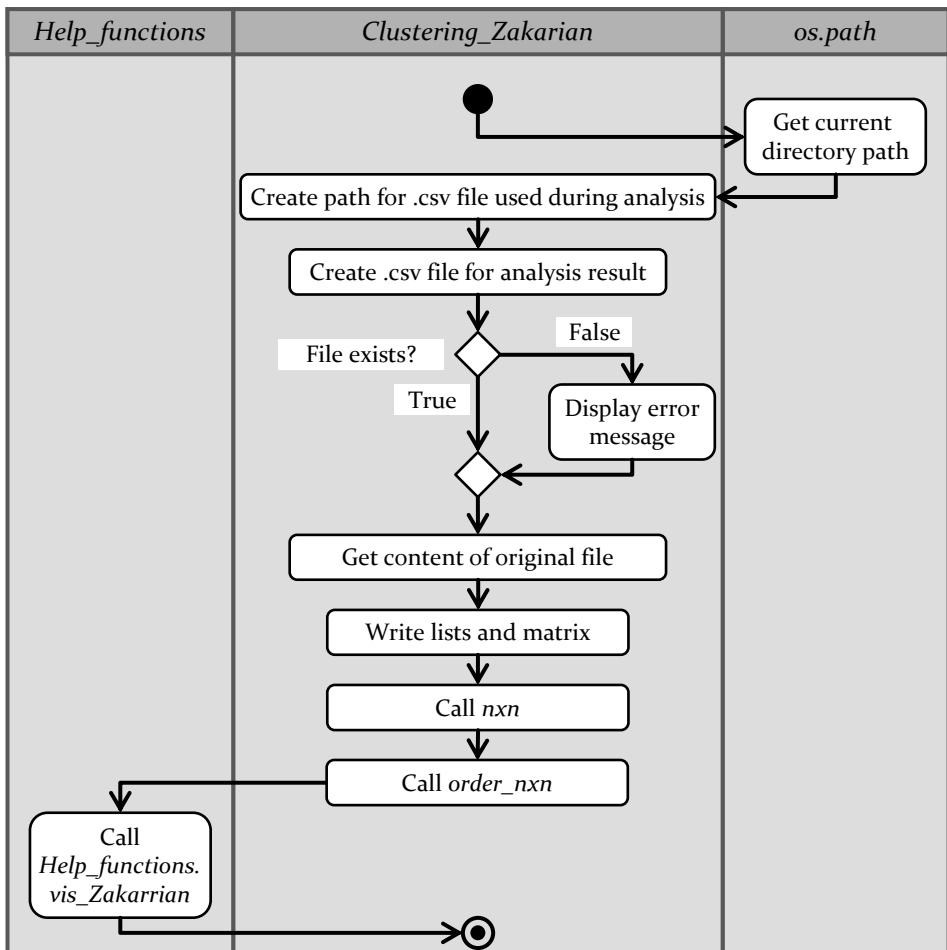


Bild 156: Aktivitätsdiagramm zum Clustering mit dem Modul *Clustering_Zakarian* (in Anlehnung an [629] und [S54])

In Bild 156 handelt es sich um das Clusterverfahren für eine $n \times n$ -Matrix. Das Clustern mit dem **Modul *Clustering_Zakarian*** erfolgt analog auch für eine $n \times m$ -Matrix. Das Vorgehen bzgl. der Dateipfade (mit *os.path*), das Einlesen und Erstellen der Matrizen und Namenslisten ist ähnlich wie bei anderen implementierten Analysemethoden (s. Bild 165). Danach wird die *nxn*-Funktion (s. Bild 157) und *order_nxn*-Funktion (s. Bild 158) aufgerufen.

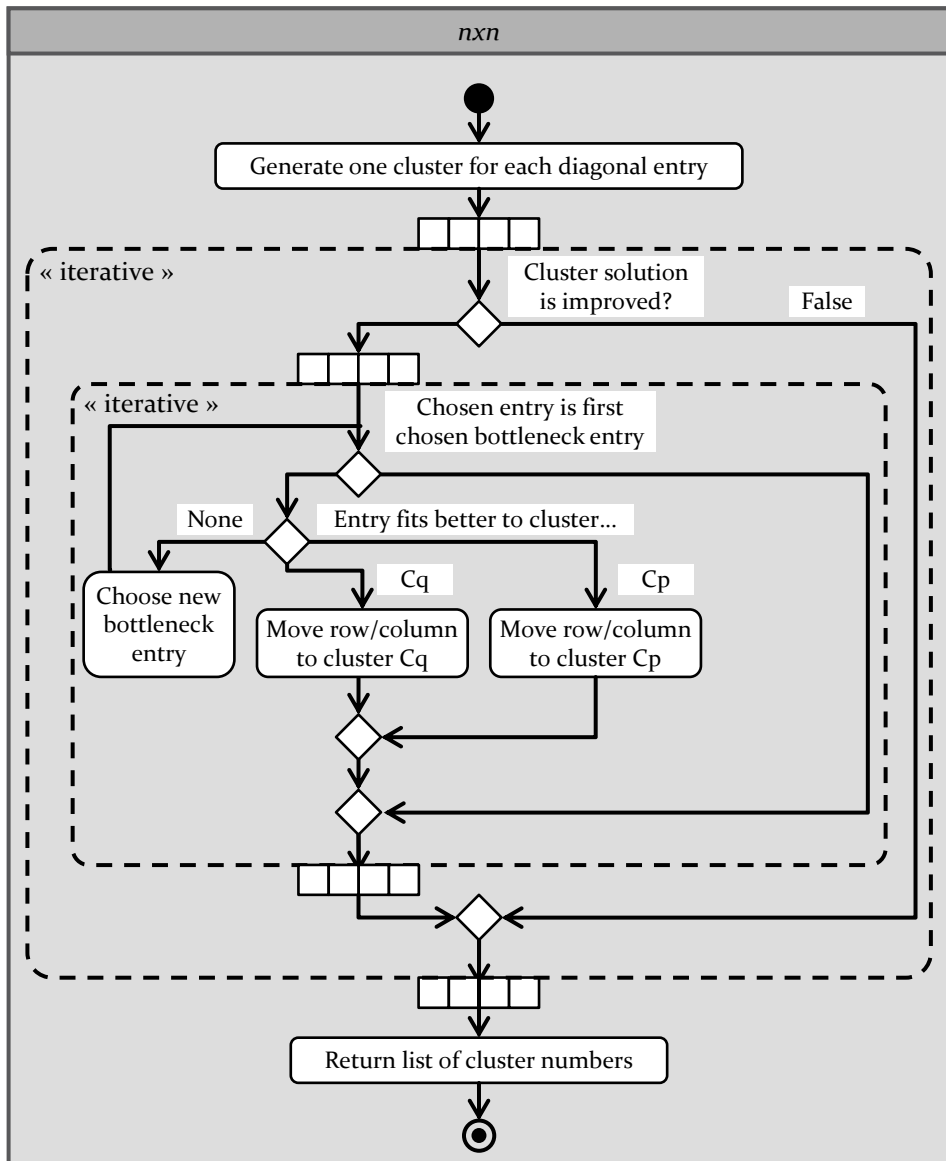


Bild 157: Aktivitätsdiagramm *nxn*-Funktion des Moduls *Clustering_Zakarian* (in Anlehnung an [629] und [S54])

Nach dem Aufruf der *nxn*-Funktion aus dem Modul *Clustering_Zakarian*, wird der **Cluster-Algorithmus** gemäß dem Aktivitätsdiagramm in Bild 157 durchlaufen. Das Resultat dieser *nxn*-Funktion ist die Speicherung des Clusterergebnisses, das an die Sortierfunktion, der *order_nxn*-Funktion, übergeben wird. Diese *order_nxn*-Funktion, die als Aktivitätsdiagramm in Bild 158 modelliert ist, liefert als Resultat geordnete Zeilen und Spalten der Matrix. Diese Matrix wird anschließend in der Wissensbasis des Assistenzsystems gespeichert und steht bspw. zur Visualisierung zur Verfügung. Das Modul *Clustering_Zakarian* ist eines von vielen Teilmodulen des Analysemoduls des Assistenzsystems (vgl. Bild 165)

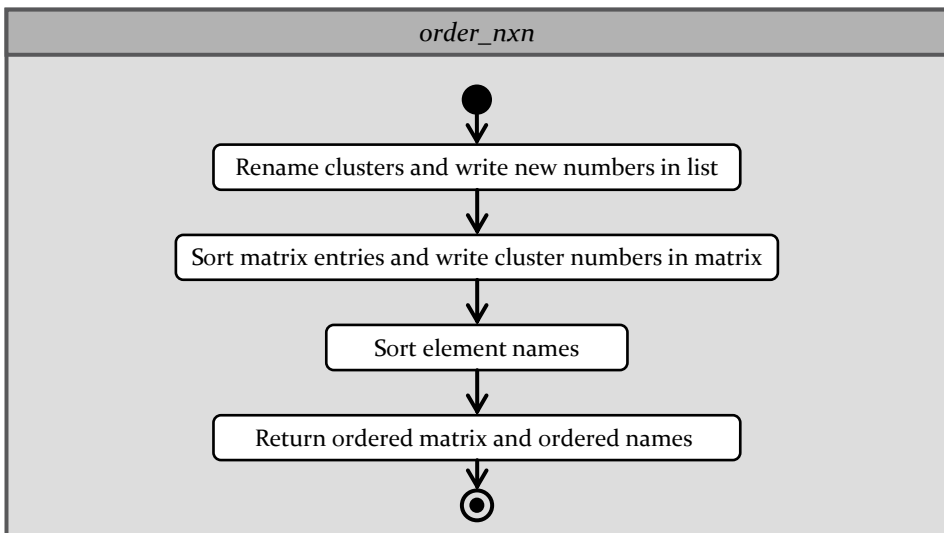


Bild 158: Aktivitätsdiagramm *order_nxn*-Funktion des Moduls *Clustering_Zakarian* (in Anlehnung an [629] und [S54])

Anwendungsbeispiele zum *Change Prediction Index*

Nach Auswahl der vereinfachten DSM des Wälzlagers wird für dieses die CPI-Methode (s. Kapitel 5.5.1) angewendet, die auf der Methode *Banding – Aktiv- und Passivsumme* aufbaut. Das Resultat der **Anwendung der CPI-Methode** ist in Bild 159 dargestellt. Darin ist zu erkennen, dass für alle 20 berücksichtigten Produktelemente des Wälzlagers die Aktiv- und Passivsumme berechnet ist (in Bild 159 hellblau eingefärbt). Dabei ist bspw. zu erkennen, dass das Element *Volume* eine Aktivsumme von 1 hat, da es nur das Element *Mass* beeinflusst, sowie eine Passivsumme von 15 hat, da es von fast allen anderen Elementen beeinflusst wird. Somit handelt es sich bei dem Element *Volume* um ein sog. *Absober* Element (s. links in Bild 159).

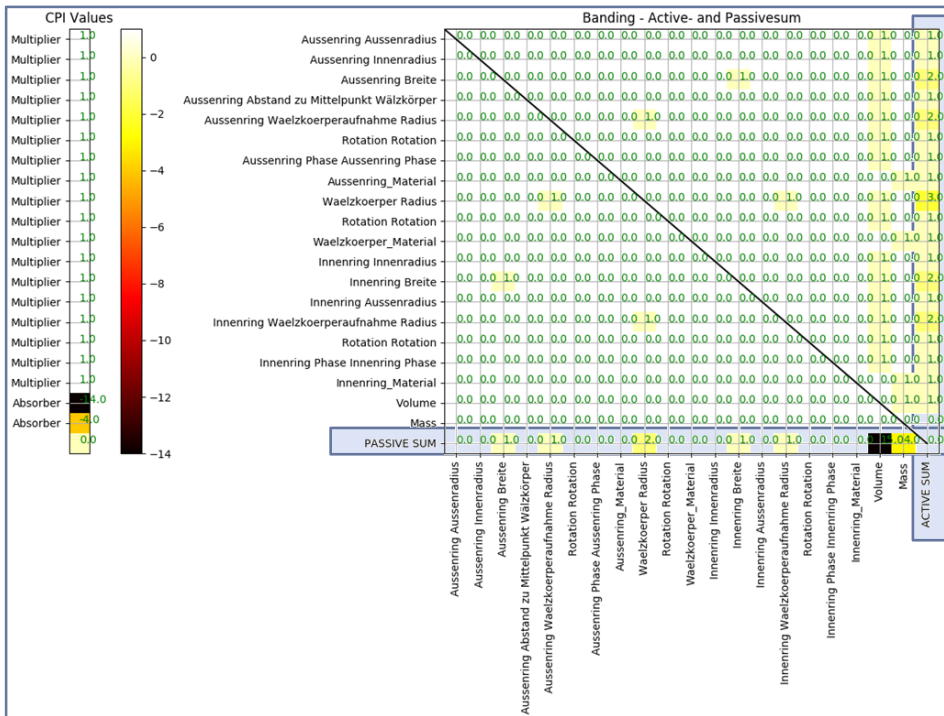


Bild 159: Resultat des *Change Prediction Index* Methode für vereinfachtes Wälzlager

Links in Bild 159 sind neben den **CPI-Werten** (*CPI Values*) auch die damit einhergehenden Kategorien der 20 Produktelement des Wälzlagers gezeigt (s. Kapitel 5.5.1 und Anhang A.20). In Bild 159 wird sichtbar, dass die meisten Elemente eine Aktivsumme von 1 haben, da diese häufig das Volumen beeinflussen. Dagegen haben die Elemente *Aussenring Breite*, *Aussenring Waelzkörperaufnahme Radius*, *Innenring Breite*, *Innenring Waelzkörperaufnahme Radius* eine Aktivsumme von 2 sowie das Element *Waelzkörper Radius* sogar von 3. Diese Elemente sind vom Komplexitätsingenieur im Rahmen bspw. des Änderungsmanagements kritischer zu bewerten, da diese bei einer einwirkenden Änderung eine Änderung von 2 bzw. 3 weiteren Elemente zur Folge haben und somit die Änderungsweitergabe sich im betrachteten System verstärken.

Anwendungsbeispiele zum *Cost Analysis Module*

In Ergänzung zum in Kapitel 6.5.3 skizzierten **Anwendungsbeispiel für das Modul Cost Analysis** werden im Folgenden weitere und detaillierte Beispiele dargestellt. So sind im Folgenden Bild 160 die CAD-Dokumente im Gegensatz zum Bild 92 nicht kombiniert bzw. aggregiert dargestellt. Dadurch wird dieses Kostenmodell zwar deutlich detaillierter und somit informativer, aber auch unübersichtlicher.

In Bild 161 ist ein vereinfachtes **Kostenmodell für die Produktdomäne** visualisiert. Auch in diesem Fenster *Cost Visualization* kann sich der Komplexitätsingenieur mittels des Mouseovers weitere Informationen anzeigen lassen. So können bspw. Informationen zu Eigenfertigungs- und Kaufteilen und zur gesamten Anzahl einzelner Produktelemente im Gesamtprodukt angezeigt werden (z. B. wird die Lagerhalbschale vier Mal pro Produkt im eigenen Unternehmen produziert und verursacht damit $4 \times 5 \text{ €}$ Kosten).

Natürlich können auch **kombinierte Kostenmodelle von zwei oder mehr Domänen** dargestellt, analysiert und somit verbessert werden. So ist bspw. in Bild 162 das Kostenmodell der zwei Domänen Wissen und Produkt visualisiert. Auf Grundlage der Relationen zwischen Elementen der beiden Domänen können auch die Auswirkungen bzgl. der Änderung von Kosten in einer Domäne auf die jeweils andere Domäne nachvollzogen werden.

Cost Visualization			
Knowledge			
ID	Element (Kandidaten)	Single Costs Accumulated Costs	Ratio
KD0	Leifensienbewertung.xls	40.01 €	1105.73 € 98.58 %
KD1	Leifensienbewertung.xls	15.01 €	690.2 € 51.38 %
RO	Lastschrift (Anforderungsliste).xlsx	10.04 €	155.87 € 13.79 %
KD2	Prokura- und Lieferantenauftrag.pdf	16.01 €	622.87 € 51.54 %
RO	Prokura- und Lieferantenauftrag.pdf	11.01 €	400.12 € 32.45 %
KD3	Zulieferer-Liste.xlsx	11.31 €	680.2 € 54.98 %
KD4	Markt-Konkurrenz-Analyse.docx	38.01 €	140.67 € 12.46 %
KD5	Trendanalyse.docx	102.67 €	102.67 € 8.07 %
KD6	Bundesgesetz.pdf	5.2 €	5.2 € 0.44 %
KD7	DH Norm Lieferhabschalen.pdf	303.01 €	658.97 € 40.79 %
KD8	Prokura- und Lieferantenauftrag.pdf	15.01 €	559.97 € 45.19 %
KD9	Funktionstruktur.pdf	18.67 €	174.53 € 15.48 %
KD10	Morphologischer Kosten.xlsx	6.67 €	585.2 € 52.05 %
KD11	abtrabonelle.art.12	7.01 €	613.87 € 54.54 %
KD12	sicherungsart.1	16.47 €	635.07 € 56.49 %
KD13	passabonelle.art.1	2.31 €	620.53 € 55.18 %
KD14	passabonelle.art.1	1.87 €	622.4 € 55.41 %
KD15	inmrcart.5	1.97 €	620.07 € 55.16 %
KD16	laerdiedel.art.18	5.61 €	612.47 € 54.45 %
KD17	laerdiedel.offen2.art.6	6.51 €	613.4 € 54.54 %
KD18	laerdiedel.art.12	7.47 €	665.33 € 41.46 %
KD19	laerdiedel.art.12	7.47 €	665.33 € 41.46 %
KD20	laerdiedel.art.12	7.47 €	665.33 € 41.46 %
KD21	abtrabonelle.dhw.1	9.31 €	620.2 € 55.43 %
KD22	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD23	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD24	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD25	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD26	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD27	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD28	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD29	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD30	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD31	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD32	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD33	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD34	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD35	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD36	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
KD37	laerdiedel.dhw.1	9.31 €	1047.27 € 93.15 %
Total Costs		1124.73 €	
Target Costs		1200 €	

Bild 160: Darstellung eines Kostenmodells der Wissensdomäne (in Anlehnung an [S13])



Bild 161: Darstellung eines vereinfachten Kostenmodells der Produktdomäne (in Anlehnung an [S13])

Prozess- und Funktionskostenrechnung mit Modul *Cost Analysis*

Außerdem kann das *Cost Analysis Module* des Assistenzsystem auch bei der **Prozesskostenrechnung** unterstützen. Hierfür dient als Grundlage die z. B. mit dem *Domain Merger* modellierte DMM-Prozess-Organisation (s. Kapitel 5.3.2). Dabei muss der Komplexitätsingenieur sicherstellen, dass allen Prozesselementen, welche in der Prozesskostenrechnung zu berücksichtigen sind, mind. ein für deren Verrichtung verantwortliches Organisationselement zugewiesen wird (vgl. Bild 59).

Die **Berechnung der Kosten von einzelnen Prozesselementen** erfolgt in ähnlicher Art und Weise wie die skizzierte Kostenberechnung für die Wissensselemente in Kapitel 5.5.3. So wird die für die Durchführung eines Prozesselements abgeschätzte bzw. tatsächlich benötigte Zeit mit dem Stundenlohn der verantwortlichen Person multipliziert und ggf. um zeitunabhängige Kosten erhöht. Dadurch werden die Kosten der einzelnen Prozessschritte (cost process step) ermittelt, wie bspw. die 245 € Kosten für den process step *Erstellung Gesamtentwurf CAD* in Bild 163. Die Addition aller Prozesskosten in der Spalte cost process step ergeben die gesamten Prozesskosten (*total costs*) für die Entwicklung des Getriebes.

Mit dem *Plot*-Button (vgl. Bild 163) wird ein **Parto-Diagramm** erzeugt, das die einzelnen Prozesselemente anhand deren Kosten absteigend sortiert. Dabei werden in absteigender Reihenfolge diejenigen Prozesselemente rot eingefärbt, die etwa 80% der kumulierten Prozesskosten umfassen. Diese Darstellung dient dem Komplexitätsingenieur bei der Identifizierung der Prozesselemente, welche einen vergleichsweise hohen Einfluss auf die Prozesskosten haben. Für diese Prozesselementen sollten zuerst Kostenreduzierungs- und Optimierungsmaßnahmen entwickelt werden.

Darüber hinaus kann das *Cost Analysis Module* des Assistenzsystem auch bei der **Funktionskostenrechnung** unterstützen. Dazu können mehrere Funktionen des Produkts (z. B. Getriebe) gewählt werden. Als Beispiel für die exemplarisch in Bild 163 dargestellte Funktionskostenrechnung kann die Funktion *F1 Übertragung Drehmoment* und die Funktion *F2 Reduktion Geräuschemission* verwendet werden. Da sich die Kosten der Funktionen anteilig aus den Kosten der Prozesselemente zusammensetzen, sind diese Prozentsätze jeweils in die Spalten *F1 [%]* und *F2 [%]* der Tabelle in Bild 163 einzutragen. Danach werden die anteiligen Kosten der Prozesselemente für die Funktionen in Spalte *F1 [€]* und *F2 [€]* automatisch berechnet und durch Addition der Werte in den jeweiligen Spalten die Kosten für die beiden Funktionen ermittelt.

Anschließend kann der Komplexitätsingenieur das jeweilige Verhältnis der Funktionskosten zum Kundennutzen berechnen und bewerten. Auf dieser Basis lassen sich i. S. der Wertanalyse gezielt Maßnahmen zur gleichzeitigen, kombinierten Optimierung der Kosten und Kunden(nutz)werte der einzelnen Funktion identifizieren. daher unterstützt das Assistenzsystem den Komplexitätsingenieur auch bei der Vorbereitung und **Durchführung von Wertanalyseprojekten**.

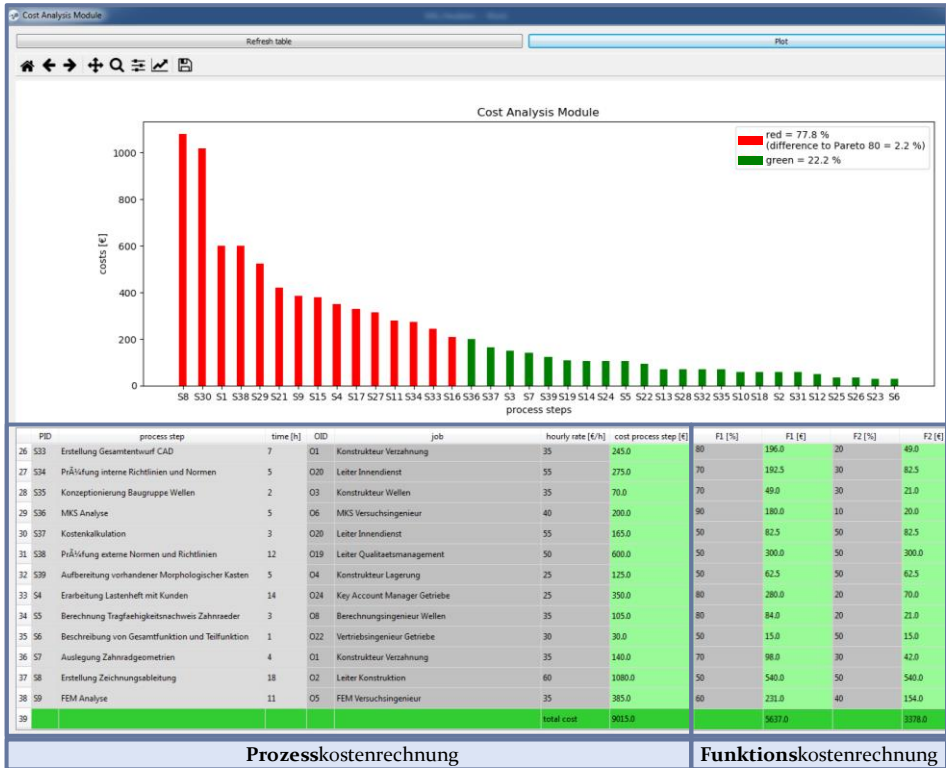


Bild 163: Stark vereinfachtes Beispiel für eine Prozess- und Funktionskostenrechnung (in Anlehnung an [S20])

A.26 Ergänzungen zum Assistenzsystem

In diesem Anhang sind hilfreiche Ergänzungen und **Grundlagen zum Fachkonzept des Assistenzsystems und zu dessen Implementierung** zusammengefasst. So werden zuerst die softwaretechnischen Grundlagen mit den wesentlichen Python-Bibliotheken für das Assistenzsystem im Überblick vorgestellt. Darauf aufbauend wird die IT-Architektur mit den wesentlichen Modulen des Assistenzsystems sowie ausgewählten externen und integrierten IT-Werkzeugen vereinfacht und überblicksartig erläutert. Abschließend werden einige, nützliche Exportformate bzw. -möglichkeiten des Assistenzsystems vorgestellt.

Softwaretechnische Grundlagen

Das Assistenzsystem hat **keine besonderen Anforderungen** an die Hardware, so dass dieses generell auf jeder Hardware eingesetzt werden kann. Zudem ist das Assistenzsystem auch weitgehend unabhängig vom Betriebssystem, da für verschiedene Betriebssysteme nur einfache Anpassungen vorzunehmen sind.

Für die **Nutzung des Assistenzsystems ist die Programmiersprache Python erforderlich**. Das Assistenzsystem ist weitgehend mit Python 3.6 programmiert worden und ist bei Bedarf (z. B. zur Nutzung bestimmter Bibliotheken) auf ältere oder neuere Versionen von Python anzupassen. Zur Integration, Verwaltung und Nutzung von z. B. Bibliotheken und Packages wird das Python-Bundles Miniconda genutzt. Als Programmierwerkzeug zur Entwicklung des Assistenzsystems wird Eclipse als integrierte Entwicklungsumgebung zusammen mit dem PyDev Modul verwendet.

Ein großer Vorteil von Python ist die **Vielzahl verfügbarer Bibliotheken und Packages**. Eine Auswahl von Bibliotheken, die im Rahmen der Programmierung des Assistenzsystems verwendet wurden, sind in Bild 164 mit einer kurzen Erläuterung zu finden. Für detaillierte Erläuterungen wird zum einen auf die vom Autor betreuten studentischen Arbeiten sowie zum anderen auf die umfangreiche Dokumentation im Internet verwiesen. Für die Erstellung der GUI wurden v. a. *PyQt4* und bei Bedarf auch auf *Tkinter* verwendet. Von besonderer Bedeutung sind auch die Module *Matplotlib*, *Numpy* und *Panda* für die Formatierung von Daten und Modellierung von Matrizen sowie *NetworkX* für die graphenbasierten Visualisierung (vgl. [240]). Zudem ist z. B. das Modul *pickle* zum Management der *.p*-Dateien und somit auch der Wissensbasis des Assistenzsystems wichtig, wobei Pickle-Dateien serialisierte Python-Objekte darstellen.

Bibliothek	Erläuterung
ctypes	Erstellung von Informationsfenstern mit unterschiedlichen Antwortmöglichkeiten für Nutzer
Matplotlib	Visualisierung von Netzwerkgraphen und Matrizen
NetworkX	Erstellung und Bearbeitung von Netzwerkgraphen, wie z. B. für den Import und Export von <i>.graphml</i> -Dateien
NumPy	Bereitstellung von Funktionen zur Matrixoperationen und zur Bearbeitung von multidimensionalen Arrays und Matrizen
os (z. B. os.path)	Nutzung bestimmter Funktionen des Betriebssystems, wie z. B. Pfaderkennung, Existenzprüfung von Dateien
Panda	Erstellung einer Matrix durch das Auswerten und Bearbeiten von tabellarische Daten
pickle	Verwaltung von <i>.p</i> -Dateien und damit der Wissensbasis des Assistenzsystems
PyQt4	Bereitstellung der graphischen Benutzeroberfläche (GUI) mit Hilfe von verschiedenen Visualisierungselementen, wie z. B. Buttons, Checkbox und Gitter
pywinauto	Steuerung von Programmen, wie z. B. AML Studio
re	Ermöglichung des Erkennens von Zahlen in einem string
selenium	Ermöglichung der Steuerung von Browsern
sys	Bereitstellung von Informationen des Python-Interpreters in Form von Funktionen und Methoden
tkinter	Bereitstellung der graphischen Benutzeroberfläche (GUI), wobei weitgehend PyQt4 für das Assistenzsystem bevorzugt wird; aufgrund der einfacheren Integration bestimmter Funktionen, z. B. eines Scrollbars, teilweise verwendet
tulip-python	Bereitstellung von Inhalt der Tulip C++ API
win32clipboard	Ermöglichung des Zugriffs auf die Windows Zwischenablage
win32com.client. dynamic	Ermöglichung einer Programmierschnittstelle (API) für Windows-Anwendungen
xlrd (z. B. xlswriter)	Ermöglichung von Interaktionen mit <i>.xlsx</i> -Dateien, z. B. Erstellung von <i>.xlsx</i> -Dateien
xml.etree.Element Tree	Ermöglichung des Durchlaufens sowie Analysierens von <i>.xml</i> -Dateien

Bild 164: Übersicht ausgewählter Python-Bibliotheken des Assistenzsystems

IT-Architektur des Assistenzsystems

Die Architektur des Assistenzsystems AS4MAVEN² mit seinen zentralen Modulen ist in Bild 48 schematisch und überblicksartig dargestellt. Im nachfolgenden Bild 165 wird die **IT-Architektur etwas konkretisiert**, aber aufgrund der Vielzahl verschiedener (Teil-)Module immer noch stark vereinfacht und lediglich mit einer Auswahl wesentlicher (Teil-)Module skizziert. In Bild 165 ist der modulare Aufbau des Assistenzsystems gezeigt.

Darin ist zu erkennen, dass sich das Assistenzsystem neben grundlegenden Methoden und Funktionen und der Wissensbasis aus den folgenden drei **übergeordneten Modulen zusammensetzt**: Matrixbasiertes Modellierungsmodul, graphbasiertes Visualisierungsmodul und mathematisches Analysemodul. Zu den grundlegenden Methoden und Funktionen gehören insb. Methoden zur Konvertierung, Integration, Manipulation und sonstige Methoden (z. B. „IMP_xml_to_csv“, „CRE_matrix“). Diese und viele weitere Methoden und Funktionen werden i. d. R. aus dem Hauptprogramm des Assistenzsystems „__Main“ aufgerufen und im Teilmodul der Wissensbasis „temp“ geladen und/oder gespeichert. Neben dem Teilmodul „temp“ v. a. für temporäre Daten setzt sich die Wissensbasis insb. aus den drei weiteren Teilmodulen zusammen: „source data“ für die Quelldaten aus externen bzw. integrierten Programmen, „saved data“ für diverse gespeicherte Daten sowie der „database“, die die diversen DSM und DMMs enthält. Das Modul zur matrixbasierten Modellierung dient insb. zur Modellerstellung und Manipulation von Matrizen im Assistenzsystem, wie z. B. mit Hilfe des in Kapitel 6.3.2 vorgestellten Domain Mergers. Das Visualisierungsmodul stellt vielfältige Möglichkeiten zur grafischen Visualisierung von Matrizen bereit. Hierzu gehören sowohl interne (z. B. mittels der Python-Bibliothek NetworkX s. Bild 164) als auch externe Optionen (z. B. mittels Tulip, yEd). Das Analysemodul unterstützt bei der Analyse mit vielen verschiedenen, spezifischen Analysealgorithmen einerseits von DSMs (z. B. Clustering, Banding, Partitioning) und andererseits von MDMs (z. B. Feedforward, Traceback, Closed Loops). Darüber hinaus sind noch einige, integrierte und externe Programme dargestellt, aus welchen domänenspezifische Modelle in das Assistenzsystem automatisiert akquiriert werden (vgl. Kapitel 5.2). Dabei sind yEd (s. z. B. DMM-Modellierung in Kapitel 5.3) und MS Excel besonders wichtig. So ist es bspw. bei der automatischen Akquisition von Elementen und deren Relationen hilfreich und zweckmäßig, dass sowohl die Möglichkeit der direkten Akquisition vom Anwendungsprogramm in das Assistenzsystem als auch die Möglichkeit eines Zwischenschritt mittels MS Excel besteht. Dadurch können Anwender bei Bedarf in MS Excel noch Anpassungen und Ergänzungen vornehmen.

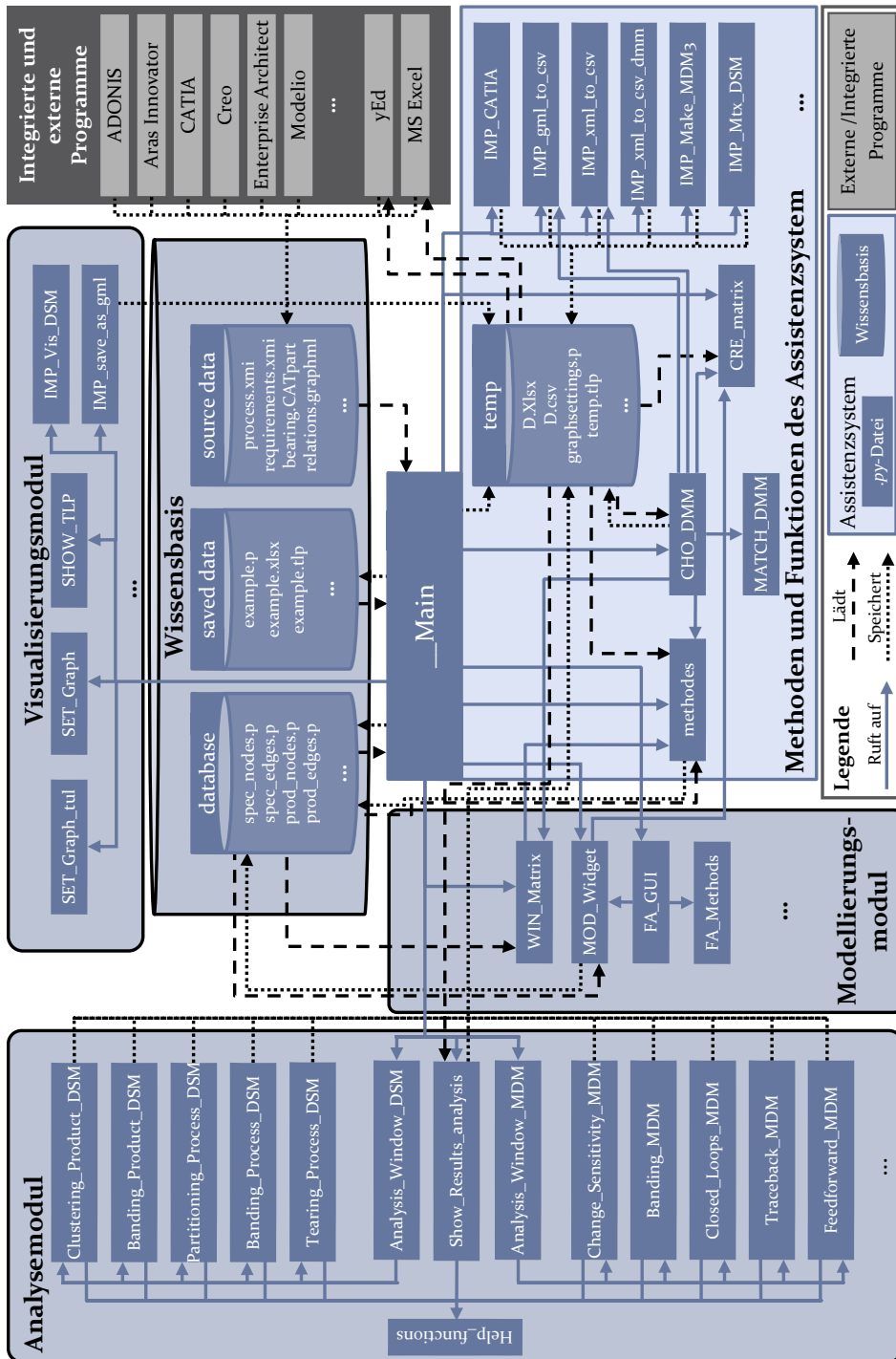


Bild 165: Überblick über die vereinfachte IT-Architektur des Assistenzsystems (Ausschnitt)

Exportmöglichkeiten des Assistenzsystems

In der Darstellung der Architektur des Assistenzsystems AS₄MAVEN² sind einige externe Programme und somit auch die i. d. R. bidirektionale Schnittstellen abgebildet (s. in Bild 48 und Bild 165). Nachfolgend liegt der Fokus auf **ausgewählten Exportmöglichkeiten**, da diese die eine weitere Bearbeitung, Analyse und Visualisierung mit anderen Programmen ermöglichen, die für bestimmte Aufgaben und Zwecke besser als das Assistenzsystem geeignet sind. Dabei lassen sich die bereits bestehenden Exportmöglichkeiten des Assistenzsystems aufgrund dessen offener, modularer Architektur beliebig erweitern. Daher stellen die im Bild 166 angeführten Exportmöglichkeiten mit beispielhaften Nutzungsmöglichkeiten nur eine kleine, beispielhafte Auswahl dar.

MS Excel ist eines der beliebtesten IT-Werkzeuge in Unternehmen und mittels eines definierten .csv-Formats können **bidirektional vielfältige Informationen zwischen dem Assistenzsystem** und z. B. MS Excel ausgetauscht werden. Zudem dienen die beiden Visualisierungsprogramme Tulip und yEd sowohl zur grafischen Darstellung als auch zur Erstellung von DSMs und v. a. DMMs (s. Kapitel 5.3.1) sowie zur Unterstützung des Änderungsmanagements. Sofern bspw. in yEd Modifikationen an Modellen vorgenommen oder neue Modelle (z. B. im UML-Layout) erstellt wurden, können diese in das Assistenzsystem automatisiert übernommen werden. Dadurch werden die Anforderungen in Kapitel 4 erfüllt, wie z. B. passende Schnittstellen zu anderen IT-Werkzeugen zur Analyse, geeignete Exportmöglichkeiten in Visualisierungstools zur Darstellung der Änderungsfortpflanzung, einfache Integrierbarkeit in die Unternehmensarchitektur.

Analyse	Knoten-/ Kantenliste im .csv-Format	Excel	Änderungsmanagement, Manipulation in Excel
	Matrizen im .csv-Format	Excel	Bearbeitung in Excel, Weiterverarbeitung in anderen IT-Werkzeugen
	Matrizen im .tlpx-Format	Tulip	Visualisierung, Änderungsmanagement, Auswirkungen von Änderungen
	Matrizen im .graphml-Format	yED	Visualisierung, Änderungsmanagement, Bearbeitung (z. B. für DMMs)
	Matrizen als Graph in den Formaten .raw, .rgba, .pdf, .png, .eps, .ps, .pgf, .svgz	PowerPoint	Präsentation, Kommunikation
Visualisierung			

Bild 166: Einige Exportmöglichkeiten des Assistenzsystems

Hintergrund zur Bezeichnung des Assistenzsystems AS₄MAVEN²

Das Wort „Maven“ (alternativ auch Mavin) stammt vom jüdischen Wort „מַבִּין meyn“ (d. h. ein Experte, Kenner, Wissender), welches aus dem Hebräischen „מֵבִין mēvīn“ (d. h. Mensch mit Verständnis, Lehrer) abgeleitet ist und vom Partizip des Verbs „הֵבִין hēvīn“ (d. h. (er) verstand) stammt [412]. Dadurch wird sowohl der **unterstützende Charakter des Assistenzsystems für das generelle Vorgehen zum Management komplexer Systeme** (s. Kapitel 5.1.1) als auch die wesentlichen Module zur Unterstützung für die Modellierung, Analyse, Visualisierung, Bewertung ausgedrückt.

Literaturverzeichnis

- [1] Abu el Ata, N.; Perks, M. J.: Solving the Dynamic Complexity Dilemma: Predictive and Prescriptive Business Management. Answering the Need for a New Paradigm. Berlin: Springer-Verlag, 2014
- [2] Abulawi, J.: Ansatz zur Beherrschung der Komplexität von vernetzten 3D-CAD-Modellen. Hamburg: Fachgebiet Maschinenelemente und Rechnergestützte Produktentwicklung, Helmut-Schmidt-Universität, Universität der Bundeswehr, 2012
- [3] Adam, D.; Johannwille, U.: Die Komplexitätsfalle. In: Adam, D. (Hrsg.): Komplexitätsmanagement. Wiesbaden: Gabler-Verlag, 1998, S. 5-28
- [4] Adept Management Ltd: Adept Management, 2020, URL: <http://www.adeptmanagement.com> (Abrufdatum: 20.07.2020)
- [5] Adunka, R.: Rechnerunterstützter Bewertungsprozess im Umfeld methodischer Produktentwicklung. Düsseldorf: VDI-Verlag, 2003
- [6] Ahmed, S.; Štorga, M.: Merged ontology for engineering design: Contrasting empirical and theoretical approaches to develop engineering ontologies. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. (23)2009, Nr. 4, S. 391-407
- [7] Albers, A.; Braun, A.: A generalised framework to compass and to support complex product engineering processes. International Journal of Product Development, (15)2011, 15, Nr. 1-3, S. 6-25
- [8] Albers, A.; Matthiesen, S.: Konstruktionsmethodisches Grundmodell zum Zusammenhang von Gestalt und Funktion technischer Systeme. Das Elementmodell „Wirkflächenpaare & Leitstützstrukturen“ zur Analyse und Synthese technischer Systeme. Konstruktion, 54(2002), Nr. 7-8, S. 55-60
- [9] Albers, A.; Wintergerst, E.: The Contact and Channel Approach (C&C2-A): Relating a System's Physical Structure to Its Functionality. In: Chakrabarti, A.; Blessing, L. (Hrsg.): An Anthology of Theories and Models of Design. London: Springer-Verlag, 2014, S. 151-171

- [10] Albers, A.; Braun, T.; Clarkson, P. J.; Enkler, H.-G.; Wynn, D. C.: Contact and channel modelling to support early design of technical systems. In: Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann, U. (Hrsg.): 17th International Conference on Engineering Design (ICED 2009). Glasgow: Design Society, 2009, S. 61-72
- [11] Albers, A.; Matthiesen, S.; Bursac, N.; Moeser, G.; Schmidt, S.; Lüdcke, R.: Abstraktionsgrade der Systemmodellierung – von der Sprache zur Anwendung. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering (TdSE 2014). München: Carl Hanser-Verlag, 2014, S. 183-192
- [12] Albers, A.; Matthiesen, S.; Bursac, N.; Moeser, G.; Klingler, S.; Schmidt, S.; Munker, F.; Scherer, H.; Kurrle, A.: Model-Based Systems Engineering (MBSE) in der Karlsruher Schule- Fünf Jahre Forschung für die Anwendung. In: K. Kohlhammer, K. (Hrsg.): develop³ systems engineering – Interdisziplinäre Produktentwicklung in der Praxis. Leinfelden-Echterdingen: Konradin-Verlag, 2016, S. 38-41
- [13] Alpar, P.; Grob, H. L.; Weimann, P.; Winter, R.: Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen. Wiesbaden: Springer Vieweg Verlag, 2014
- [14] Alt, O.: Modellbasierte Systementwicklung mit SysML. München: Hanser-Verlag, 2012
- [15] Andreasen, M. M.: Complexity of Industrial Practice and Design Research Contributions – We Need Consolidation. In: Meerkamm, H. (Hrsg.): 20. Symposium Design for X (DfX 2009). Neukirchen/Erlangen: Lehrstuhl für Konstruktionstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2009, S. 1-9

- [16] Angermeier, D.; Bartelt, C.; Bauer, O.; Beneken, G.; Bergner, K.; Birowicz, U.; Bliß, T.; Breitenstrom, C.; Cordes, N.; Cruz, D.; Dohrmann, P.; Friedrich, J.; Gnatz, M.; Hammerschall, U.; Hidvegi-Barstorfer, I.; Hummel, H.; Israel, D.; Klingenberg, T.; Klugseder, K.; Küffer, I.; Kuhrmann, M.; Kranz, M.; Kranz, W.; Meinhardt, H.-J.; Meisinger, M.; Mittrach, S.; Neußer, H.-J.; Niebuhr, D.; Plögert, K.; Rauh, D.; Rausch, A.; Rittel, T.; Rösch, W.; Saas, E.; Schramm, J.; Sihling, M.; Ternité, T.; Vogel, S.; Weber, B.; Wittmann, M.: V-Modell XT: Das deutsche Referenzmodell für Systementwicklungsprojekte. Version: 2.3. München: Verein zur Weiterentwicklung des V-Modell XT e.V. (Weit e.V.); Berlin: Informationstechnikzentrum Bund (ITZBund), 2019
- [17] Antoniou, G.; Groth, P. E.; Harmelen, F. V.; Hoekstra, R.: A Semantic Web Primer. Cambridge: The MIT Press, 2012
- [18] Archambault, D., Tóth, C. D.: Graph Drawing and Network Visualization. Cham: Springer-Verlag, 2019
- [19] Aristoteles: Aristoteles' Metaphysik. In: Seidl, H. (Hrsg.): Aristoteles' Metaphysik: Bücher VII(Z)-XIV(N). Griechisch-Deutsch. Hamburg: Meiner Felix Verlag, 2009
- [20] Ashby, W. R.: An introduction to cybernetics. London: Chapman & Hall Verlag, 1956
- [21] Aßmann, G.; Gerst, M.; Stetter, R.; Allmansberger, G.; Kleedörfer, R.; Riedel, D.; Voigt, P.; Conrat, J.-I.: Aktionsfelder des Integrierten Änderungsmanagements. In: Lindemann, U.; Reichwald, R. (Hrsg.): Integriertes Änderungsmanagement. Berlin: Springer-Verlag, 1998, S. 107-256
- [22] Aßmann, G.: Gestaltung von Änderungsprozessen in der Produktentwicklung. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2000
- [23] Atzberger, A.; Paetzold, K.: Current challenges of agile hardware development: What are still the pain points nowadays. In: Wartzack, S.; Schleich, B.; Gon (Hrsg.): 22nd International Conference on Engineering Design (ICED 2019). Cambridge: Cambridge University Press, 2019, S. 2209-2218

- [24] Atzberger, A.; Nicklas, S. J.; Schrof, J.; Weiss, S.; Paetzold, K.: Agile Entwicklung physischer Produkte. Eine Studie zum aktuellen Stand in der industriellen Praxis. Neubiberg: Institut für Produktentwicklung, Universität der Bundeswehr München, 2020
- [25] Auber, D.; Mary, P. (u. a.): Tulip - Better Visualization Through Research - Data Visualization Software, 2020, URL: <https://tulip.labri.fr> (Abrufdatum: 06.09.2020)
- [26] Aurich, J. C.; Grzegorski, A.: Vielfaltsinduzierte Komplexität in Ingenieurprozessen. ZWF – Zeitschrift für wirtschaftlichen Fabrikbetrieb, (103)2008, Nr. 5, S. 316-321
- [27] große Austing, S.: Komplexitätsmessung von Produktmodellen. Oldenburg: Abteilung Systemanalyse- und optimierung, Carl von Ossietzky Universität Oldenburg, 2012
- [28] Avnet, M. S.; Weigel, A. L.: An application of the design structure matrix to integrated concurrent engineering. Acta Astronautica, (66)2010, Nr. 5, S. 937-949
- [29] Bach, N.; Brehm, C.; Buchholz, W.; Petry, T.: Organisation. Gestaltung wertschöpfungsorientierter Architekturen, Prozesse und Strukturen. Wiesbaden: Springer Gabler Verlag, 2017
- [30] Backhaus, K.; Voeth, M.: Internationales Marketing. Stuttgart: Schäffer-Poeschel-Verlag, 2010
- [31] Badow, G.; Holzmüller, H. (Hrsg.): „Das ist gar kein Modell!“ – Unterschiedliche Modelle und Modellierungen in Betriebswirtschaftslehre und Ingenieurwissenschaften. Wiesbaden: Gabler Verlag, 2010
- [32] Bartolomei, J.: Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure. Cambridge: Engineering Systems Division, Massachusetts Institute of Technology (MIT), 2007
- [33] Bartolomei, J.; Cokus, M.; Dahlgren, J.; de Neufville, R.; Maldonado, D.; Wilds, J.: Analysis and applications of design structure matrix, domain mapping matrix, and engineering system matrix frameworks. Cambridge: Engineering Systems Division, Massachusetts Institute of Technology (MIT), 2007
- [34] Bartuschat, M.: Beitrag zur Beherrschung der Variantenvielfalt in der Serienfertigung. Essen: Vulkan-Verlag, 1995

- [35] Battenfeld, D.: Behandlung von Komplexitätskosten in der Kostenrechnung. Controlling und Management, 45(2001), Nr. 3, S. 137-143
- [36] Bauer, S.: Entwicklung eines Werkzeugs zur Unterstützung multikriterieller Entscheidungen im Kontext des Design for X. Düsseldorf: VDI-Verlag, 2009
- [37] Bauert, F.: Methodische Produktmodellierung für den rechner-unterstützten Entwurf. Berlin: Institut für Maschinenkonstruktion – Konstruktionstechnik, Technische Universität Berlin, 1991
- [38] Bayer, T.: Integriertes Variantenmanagement. Variantenkostenbewertung mit faktorenanalytischen Komplexitätstreibern. München: Rainer Hampp Verlag, 2010
- [39] Becker, J.; Pfeiffer, D.: Konzeptionelle Modellierung – ein wissenschaftstheoretischer Forschungsleitfaden. In: Lehner, F.; Nösekabel, H.; Kleinschmidt, P. (Hrsg.): Multikonferenz Wirtschaftsinformatik (MKWI 2006), Vol. 2. Berlin: Gito-Verlag, 2006, S. 3-19
- [40] Becker, J.; Probandt, W.; Vering, O.: Grundsätze ordnungsmäßiger Modellierung. Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement. Berlin: Springer-Verlag, 2012
- [41] van Beek, T. J.; Erden, M. S.; Tomiyama, T.: Modular design of mechatronic systems with function modeling. Mechatronics, (20)2010, Nr. 8, S. 850-863
- [42] Beier, G.; Figge, A.; Marwedel, S.: A MBSE Approach for the Development of Complex Technical Systems. International Journal of Engineering Research and Application. (7)2017, Nr. 9, S. 24-36
- [43] Beitz, W.: Systemtechnik in der Konstruktion. DIN-Mitteilungen 49 (1970) 8, S. 295-302
- [44] Beneke, F.: Konzeptionelle Ansätze einer prozessorientierten Produktentwicklung. Aachen: Shaker Verlag, 2003
- [45] Bennett, N.; Lemoine, G. J.: What VUCA Really Means for You. Harvard Business Review 92(2014), Nr. 1-2, S. 27

- [46] Bensberg, F.; Buscher, G.: Anforderungen der Praxis an das Geschäftsprozessmanagement. Berufsbilder, Kompetenzen und Tätigkeitsfelder. In: Barton, T.; Müller, C.; Seel, C. (Hrsg.): Geschäftsprozesse. Von der Modellierung zur Implementierung. Wiesbaden: Springer Vieweg, 2017, S. 221-237
- [47] Berens, W.; Schmitting, W.: Controllinginstrumente für das Komplexitätsmanagement: Potentiale des internen Rechnungswesens. In: Adam, D. (Hrsg.): Komplexitätsmanagement, Wiesbaden: Gabler Verlag, 1998, S. 97-110
- [48] von Bertalanffy, L.: General System Theory. Foundations, Development, Applications. New York: George Braziller, 1969
- [49] von Bertalanffy, L.: Vorläufer und Begründer der Systemtheorie. In: Kurzrock, R. (Hrsg.): Systemtheorie. Berlin: Colloquium Verlag, 1972, S. 17-28
- [50] von Bertalanffy, L.: Zu einer allgemeinen Systemlehre. In: Bleicher, K. (Hrsg.): Organisation als System. Wiesbaden: Gabler Verlag, 1972, S. 31-45 (aus: von Bertalanffy, L.: Zu einer allgemeinen Systemlehre. Biologia Generalis, 19(1949), Nr. 1, S. 114-129
- [51] Bhaskara, S.: Re-engineering legacy knowledge based engineering systems using DSM. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (Hrsg.): 10th International Design Structure Matrix Conference (DSM 2008). München: Carl Hanser Verlag, 2008, S. 273-284
- [52] Bhaskara, S.: DSM Based Approach for Managing Requirements, Rules and Design Parameters in Knowledge Based Design Process. In: Wynn, D. C.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, J. (Hrsg.): 12th International Design Structure Matrix Conference (DSM 2010). München: Carl Hanser Verlag, 2010, S. 237-248
- [53] Bhaskara, S.: Analysis and visualization of complex computer aided design models as a Design Structure Matrix. In: Eppinger, S. D.; Maurer, M.; Eben, K.; Lindemann, U. (Hrsg.): 13th International Design Structure Matrix Conference (DSM 2011). München: Carl Hanser Verlag, 2011, S. 61-75
- [54] Bhaskara, S.; Chakraborty, G. (Erfinder): Modernization of legacy software systems based on modelled dependencies. In: The Boeing Company (Anmelder): United States Patent, US 8,930,919 B2, 2015

- [55] Biedermann, W.: A minimal set of network metrics for analysing mechatronic product concepts. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2015
- [56] Biedermann, W.; Lindemann, U.: Cycles in the Multiple-Domain Matrix – Interpretation and Applications. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (Hrsg.): 10th International DSM Conference (DSM 2008). München: Carl Hanser Verlag, 2008, S. 25-34
- [57] Biedermann, W.; Lindemann, U.: On the applicability of structural criteria in complex management. In: Culley, S. J.; Hicks, B. J.; McAloone, T. C.; Howard, T. J.; Lindemann, U. (Hrsg.): 18th International Conference on Engineering Design (ICED 2011). Glasgow: Design Society, 2011, S. 21-32
- [58] Biedermann, W.; Kreimeyer, M.; Lindemann, U.: Measurement System to Improve Data Acquisition Workshops. In: Kreimeyer, M.; Maier, J.; Fadel, G.; Lindemann, U. (Hrsg.): 11th International DSM Conference (DSM 2009). München: Carl Hanser Verlag, 2009, S. 119-130
- [59] Bimazubute, R.: Die Nachbereitung von Experteninterviews im expertenzentrierten Wissensmanagement. Erlangen: Lehrstuhl Informatik 8 (Künstliche Intelligenz), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2005
- [60] Birta, L.; Arbez, G.: Modelling and Simulation. Exploring Dynamic System Behaviour. London: Springer Verlag, 2019
- [61] Bischof, N.; Eppler, M. J.: Caring for Clarity in Knowledge Communication. Journal of Universal Computer Science, (17)2011, Nr. 10, S. 1455-1473
- [62] Black, T. A.; Fine C. H.; Sachs, E. M.: A Method for Systems Design Using Precedence Relationships: An Application to Automotive Brake Systems. Cambridge: Sloan School of Management, Massachusetts Institute of Technology (MIT), 1990
- [63] Blanchard, B.; Blyler, J.: System Engineering Management. Hoboken: Wiley Verlag, 2016
- [64] Blass, E.: Entwicklung verfahrenstechnischer Prozesse. Methoden, Zielsuche, Lösungssuche, Lösungsauswahl. Heidelberg: Springer-Verlag, 1997
- [65] Bleicher, K. (Hrsg.): Organisation als System. Gabler Verlag, 1972

- [66] Bliss, C. Management von Komplexität. Ein integrierter, system-theoretischer Ansatz zur Komplexitätsreduktion. Wiesbaden: Springer Gabler Verlag, 2000
- [67] Böcher, W.: Natur, Wissenschaft und Ganzheit. Über die Welterfahrung des Menschen. Opladen/Wiesbaden: Westdeutscher Verlag, 1992
- [68] Bodendorf, F.: Daten- und Wissensmanagement. Berlin: Springer-Verlag, 2006
- [69] Boelter, A.: Ein Beitrag zur Kontrolle und Darstellung des Leistungsgrades von Projektleitern unter Einbeziehung der Projektkomplexität. Siegen: Universität Siegen, 1992
- [70] Boerner, S.: Führungsverhalten und Führungserfolg. Beitrag zu einer Theorie der Führung am Beispiel des Musiktheaters. Wiesbaden: Deutscher Universitäts-Verlag, 2002
- [71] Bohne, F.: Komplexitätskostenmanagement in der Automobilindustrie. Identifizierung und Gestaltung vielfaltsinduzierter Kosten. Wiesbaden: Springer Gabler Verlag, 1998
- [72] Borjesson, F.; Hölttä-Otto, K.: Improved Clustering Algorithm for Design Structure Matrix. In: ASME (Hrsg.): International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2012). New York: ASME, 2012, S. 921-930
- [73] Borjesson, F.; Sellgren, U.: Fast Hybrid Genetic Clustering Algorithm for Design Structure Matrix. In: ASME (Hrsg.): International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2013). New York: ASME, 2013, S. 1-10
- [74] Brachman, R.; Levesque, H.: Knowledge representation and reasoning. San Francisco: Morgan Kaufmann Verlag, 2004
- [75] Braun, A.: Modellbasierte Unterstützung der Produktentwicklung: Potentiale der Modellierung von Produktentstehungsprozessen am Beispiel des integrierten Produktentstehungsmodells (iPeM). Karlsruhe: Institut für Produktentwicklung, Karlsruher Institut für Technologie (KIT), 2014

- [76] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M.; Maler, E.; Yergeau, F. (Hrsg.): Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation 26 November 2008, 2008, URL: <https://www.w3.org/TR/xml> (Abrufdatum: 31.07.2020)
- [77] Breiing, A.; Knosala, R.: Bewerten technischer Systeme. Theoretische und methodische Grundlagen bewertungstechnischer Entscheidungshilfen. Berlin: Springer-Verlag, 1997
- [78] Breitsprecher, T.; Codescu, M.; Jucovschi, C.; Kohlhase, M.; Schröder, L.; Wartzack, S.: Towards Ontological Support for Principle Solutions in Mechanical Engineering. In: Garbacz, P.; Kutz, O. (Hrsg.): 8th International Conference on Formal Ontology in Information Systems (FOIS 2014). Amsterdam: IOS Press, 2014, S. 427-432
- [79] Brickley, D.; Guha, R. V.: RDF Schema 1.1. W3C Recommendation 25 February 2014, 2014, URL: <http://www.w3.org/TR/2014/REC-rdf-schema-20140225> (Abrufdatum: 30.06.2020)
- [80] Brimble, R.; Sellini, F.: The MOKA Modelling Language. In: Dieng R.; Corby, O. (Hrsg.): Knowledge Engineering and Knowledge Management Methods, Models, and Tools. Berlin: Springer-Verlag, 2000, S. 49-56
- [81] Brockhoff, K.: Dynamics of technological competencies. In: Brockhoff, K.; Chakrabarti, A. K.; Hauschildt, J. (Hrsg.): The Dynamics of Innovation. Strategic and Managerial Implications. Berlin: Springer-Verlag, 1999, S. 31-56
- [82] Brockhoff, K.: Produktinnovation. In: Albers, S.; Herrmann, A. (Hrsg.): Handbuch Produktmanagement. Strategieentwicklung - Produktplanung - Organisation - Kontrolle, Springer Gabler Verlag, 2002, 25-54
- [83] Bronner, A.: Wertanalyse als integrierte Rationalisierung. Werkstattstechnik – Zeitschrift für industrielle Fertigung 58(1968) Nr. 1, S. 16-21
- [84] Brosch, M.: Eine Methode zur Reduzierung der produktvarianteninduzierten Komplexität. Hamburg: Institut für Produktentwicklung und Konstruktionstechnik, Technischen Universität Hamburg-Harburg (TUHH), 2014

- [85] Broßmann, M.; Mödinger, W.: Praxisguide Wissensmanagement. Qualifizieren in Gegenwart und Zukunft. Planung, Umsetzung und Controlling in Unternehmen. Berlin: Springer-Verlag, 2011
- [86] Browning, T. R.: Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Transactions on Engineering Management, (48)2001, Nr. 3, S. 292-306
- [87] Browning, T. R.: Process Integration Using the Design Structure Matrix. Systems Engineering, (5)2002, Nr. 3, S. 180-193
- [88] Browning, T. R.: Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. IEEE Transactions on Engineering Management, (63)2016, Nr. 1, S. 27-52
- [89] Browning, T.; Clarkson, J.; Eckert, C.; O'Donovan, B.: Design planning and modelling. In: Clarkson, J.; Eckert, C. (Hrsg.): Design process improvement - A review of current practice. London: Springer-Verlag, 2005, S. 60-87
- [90] Browning, T. R.; Fricke, E.; Negele, H.: Key concepts in modeling product development processes. Systems Engineering, 9(2006), Nr. 2, S. 104-128
- [91] Budde, L.: Integriertes Komplexitätsmanagement in produzierenden Unternehmen. St. Gallen: Lehrstuhl für Produktionsmanagement, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften sowie Internationale Beziehungen, Universität St. Gallen (HSG), 2015
- [92] Bullinger, H.-J.; Scheer, A.-W. (Hrsg.): Service Engineering. Entwicklung und Gestaltung innovativer Dienstleistungen. Berlin: Springer-Verlag, 2006
- [93] Bullinger, H.-J.; Wörner, K.; Pietro, J.: Wissensmanagement heute. Daten, Fakten, Trends. Stuttgart: Fraunhofer-Institut Arbeitswirtschaft und Organisation (IAO), 1997
- [94] Bullinger, H.-J.; Fähnrich, K.-P.; Meiren, T.: Service engineering - methodical development of new service products. International Journal of Production Economics, 85(2003), Nr. 3, S. 275-287
- [95] Bullinger, H.-J.; Warnecke, H.; Westkämper, E. (Hrsg.): Neue Organisationsformen im Unternehmen. Ein Handbuch für das moderne Managemen. Berlin: Springer-Verlag, 2003

- [96] Bundesverwaltungsamt, Kompetenzzentrum Prozessmanagement (Hrsg.): Verzahnung von Fachbereichen und IT: Aufbau eines IT Managements zur effizienten Kommunikation zwischen Fachbereichen und internen/externen IT Dienstleistern. Köln: Bundesverwaltungsamt, 2012
- [97] Bundesverwaltungsamt, Kompetenzzentrum Prozessmanagement (Hrsg.): Konventionenhandbuch Teil 1 für eine einheitliche Prozessmodellierung. Köln: Bundesverwaltungsamt, 2016
- [98] Bundesverwaltungsamt, Kompetenzzentrum Prozessmanagement (Hrsg.): Konventionenhandbuch Teil 2 für eine einheitliche Modellierung von IT-Prozessen und Diensten. Köln: Bundesverwaltungsamt, 2016
- [99] Bursac, N.: Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung. Karlsruhe: Institut für Produktentwicklung (IPEK), Karlsruher Institut für Technologie (KIT), 2016
- [100] Büsing, C.: Graphen-und Netzwerkoptimierung. Springer-Verlag, 2010
- [101] Busse, J.; Humm, B.; Lübbert, C.; Moelter, F.; Reibold, A.; Rewald, M.; Schlüter, V.; Seiler, B.; Tegtmeier, E.; Zeh, T.: Was bedeutet eigentlich Ontologie? Ein Begriff aus der Philosophie im Licht verschiedener Disziplinen. Informatik-Spektrum. (37)2014, Nr. 4, S. 286-297
- [102] Buur, J.; Andreasen, M. M.: Design models in mechatronic product development. Design Studies, (10)1989, Nr. 3, S. 155-162
- [103] Chen, X.; Gao, S.; Guo, S.; Bai, J.: A flexible assembly retrieval approach for model reuse. Computer-Aided Design, (44)2012, Nr. 6, S. 554-574
- [104] Cheng, Q.; Zhang, G.; Gu, P.; Shao, X.: A product module identification approach based on axiomatic design and design structure matrix. Concurrent Engineering, (20)2012, Nr. 3, S. 185-194
- [105] Cilliers, P.: Complexity and postmodernism. understanding complex systems. New York: Routledge-Verlag, 1998

- [106] Clarkson, P. J.; Simons, C.; Eckert, C.: Predicting Change Propagation In Complex Design. In: ASME (Hrsg.): Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2001). New York: ASME, 2001, S. 1-10
- [107] Clarkson, P. J.; Simons, C.; Eckert, C.: Predicting Change Propagation in Complex Design. Journal of Mechanical Design, (126)2004, Nr. 5, 788-797
- [108] Clement, S.; Kittel, K.; Meyer, A.: Creo Parametric 2.0 für Einsteiger - kurz und bündig. Grundlagen mit Übungen. Wiesbaden: Springer-Verlag, 2013
- [109] Codreanu, A.: A VUCA action framework for a VUCA environment. Leadership challenges and solutions. Journal of Defense Resources Management, 7(2016) Nr. 2, S. 31-38
- [110] Cohen, T.; Navathe, S.; Fulton, R.: C-FAR, change favorable representation. Computer-Aided Design, (32)2000, Nr. 5-6, S. 321-338
- [111] Colombo, G.; Mosca, A.; Sartori, F.: Towards the design of intelligent CAD systems: An ontological approach. Advanced Engineering Informatics. (21)2007, Nr. 2, S. 153-168
- [112] Coulibaly, A.; Mutel, B.; Ait-Kadi, D.: Product modeling framework for behavioral performance evaluation at design stage. Computers in Industry, (58)2007, Nr. 6, S. 567-577
- [113] Coulibaly, A.; Houssin, R.; Mutel, B.: Maintainability and safety indicators at design stage for mechanical products. Computers in Industry, (59)2008, Nr. 5, S. 438-449
- [114] Coulibaly, A.; De Bertrand De Beuvron, F.; Renaud, J.: Maintainability assessment at early design stage using advanced CAD systems. In: Fischer, X.; Nadeau, J.-P. (Hrsg.): Virtual, Interactive and Integrated Product Design and Manufacturing for Industrial Innovations (IDMME - Virtual Concept 2010), Berlin: Springer-Verlag, 2010, S. 1-6
- [115] Crostack, A.: Grundlagen einer Konstruktionsmethodik für Hybride Intelligente Konstruktionselemente (HIKE). Stuttgart: Institut für Konstruktionstechnik und Technisches Design, Universität Stuttgart, 2018

- [116] Crostack, A.; Binz, H.: In: Binz, H. (Hrsg.): Formalisierte Wissensrepräsentation zur Unterstützung des methodischen Konstruierens von und mit hybriden intelligenten Konstruktionselementen (HIKE) und Systeme. In: Binz, H. (Hrsg.): Hybride Intelligente Konstruktionselemente (HIKE) – Abschlusskolloquium der DFG-Forschergruppe 981. Stuttgart: Institut für Konstruktionstechnik und Technisches Design, Universität Stuttgart, 2016, S. 9-12 und 52-65
- [117] Daenzer, W. F.; Huber, F. (Hrsg.): Systems Engineering. Methodik und Praxis. Zürich: Verlag Industrielle Organisation, 2002
- [118] Dahlgren, J. W.: Developing a Framework for Exploring Clustering Coefficient to Evaluate System Coupling. International Command and Control Research Technology Symposium (ICCRTS 2007). McLean: The MITRE Corporation, 2007
- [119] Danilovic, M.; Browning, T. R.: Managing complex product development projects with design structure matrices and domain mapping matrices. International journal of project management, 25(2007), Nr. 3, S. 300-314
- [120] Dehnen, K.: Strategisches Komplexitätsmanagement in der Produktentwicklung. Hamburg: Dr. Kovac Verlag, 2004
- [121] Denkena, B.; Tönshoff, H. K.; Apitz, R.; Granthien, M.: Wissensmanagement im integrierten Produktlebenszyklus. ZWF – Zeitschrift für wirtschaftlichen Fabrikbetrieb, 97(2002), Nr. 9, S. 428-430
- [122] Dhir, S.; Sushil (Hrsg.): Flexible Strategies in VUCA Markets. Singapore: Springer Nature, 2018
- [123] Diagne, S.; Coulibaly, A.; Sene, M.: Complex Mechatronic Product Modeling Using a Multi-Solution, Multi-instance Extended Conceptual Design Semantic Matrix. In: Marle, F.; Jankovic, M.; Maurer, M.; Schmidt, D. M.; Lindemann, U. (Hrsg.): 16th International DSM Conference (DSM 2014). München: Carl Hanser Verlag, 2014, S. 85-94
- [124] Diagne, S.; Coulibaly, A.; De Bertrand De Beuvron, F.: Complex product modeling based on a Multi-solution eXtended Conceptual Design Semantic Matrix for behavioral performance assessment. Computers in Industry, (75)2016, Nr. 1, S. 101-115

- [125] Di Battista, G.; Eades, P.; Tamassia, R.; Tollis, I. G.: Algorithms for drawing graphs: an annotated bibliography. Computational Geometry: Theory and Applications, (4)1994, Nr. 5, S. 235-282
- [126] Diestel, R.: Graph Theory. Berlin: Springer-Verlag, 2017
- [127] DIN EN ISO 10209:2012-11: Technische Produktdokumentation - Vokabular - Begriffe für technische Zeichnungen, Produktdefinition und verwandte Dokumentation. Berlin: Beuth Verlag, 2012
- [128] DIN EN 10088-3:2014-12: Nichtrostende Stähle – Teil 3. Technische Lieferbedingungen für Halbzeug, Stäbe, Walzdraht, gezogenen Draht, Profile und Blankstahlerzeugnisse aus korrosionsbeständigen Stählen für allgemeine Verwendung. Berlin: Beuth Verlag, 2014
- [129] DIN EN 10027-2:2015-07: Bezeichnungssysteme für Stähle – Teil 2. Nummernsystem. Berlin: Beuth Verlag, 2015
- [130] Doheny, M.; Nagali, V.; Weig, F.: Agile operations for volatile times. McKinsey Quarterly 3(2012), S. 126-131
- [131] Dong, Q.: Predicting and Managing System Interactions at Early Phase of the Product Development Process. Cambridge: Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), 2002
- [132] Dori, D.: Model-Based Systems Engineering with OPM and SysML. New York: Springer-Verlag, 2016
- [133] Dörner, D.: Die Logik des Misslingens. Strategisches Denken in komplexen Situationen. Reinbek/Hamburg: Rowohlt Verlag, 2011
- [134] Drawehn, J.; Kochanowski, M.; Kötter, F.; Weisbecker, W.; Drawehn, J. (Hrsg.): Business Process Management Tools 2014. Stuttgart: Fraunhofer-Verlag, 2014
- [135] Düchting, C.: Aufbau eines freigabe- und kommunikationsbasierten Assistenzsystems im Produktentstehungsprozess. Dortmund: Lehrstuhl für Fabrikorganisation, Universität Dortmund, 2005
- [136] Dudenredaktion (o. J.): „Domäne“ auf Duden online. URL: <https://www.duden.de/node/33952/revision/33981> (Abrufdatum: 30.05.2020)

- [137] DWDS – Digitales Wörterbuch der deutschen Sprache: „komplex“. In: Pfeifer, W. (Hrsg.): Etymologisches Wörterbuch des Deutschen (1993), digitalisierte und von Wolfgang Pfeifer überarbeitete Version im DWDS, URL: <https://www.dwds.de/wb/etymwb/komplex> (Abrufdatum: 30.05.2020)
- [138] Dworschak, F.; Zirngibl, C.; Schleich, B.; Wartzack, S.: Konzept für den MBSE-Einsatz zur automatisierten Individualisierung von komplexen Produkten. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 30. DfX-Symposium (DfX2019). Hamburg: TuTech, 2019, S. 279-290
- [139] Eades, P.: A heuristic for graph drawing. *Congressus Numerantium*, (42)1984, Nr. 11, S. 149-160
- [140] Eades, P.; Gutwenger, C.; Hong, S.-H.; Mutzel, P.: Graph drawing algorithms. In: Atallah, M.; Blanton, M. (Hrsg.): *Algorithms and theory of computation handbook: Special Topics and Techniques*. London: Chapman & Hall Verlag, 2010, S. 6-1-6-27
- [141] Eberle, R.: Varianten-Controlling in der Automobil-Industrie. *Controlling und Management*, 44(2000), Nr. 6, S. 343-350
- [142] Ebers, M.; Arbeitskreis Organisation der Schmalenbachgesellschaft für Betriebswirtschaft e.V.: Organisationsmodelle für Innovation. *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung*, (69)2016, Nr. 1, S. 81-109
- [143] Eckert, C.; Clarkson, P. J.; Zanker, W.: Change and customisation in complex engineering domains. *Research in Engineering Design*, (15)2004, Nr. 1, S. 1-21
- [144] Ehms, K.; Richter, A.: Vielfalt und Überforderung. In: Richter, A. (Hrsg.): *Vernetzte Organisation*. München: Oldenbourg Wissenschaftsverlag, 2014, S. 49-54
- [145] Ehrig, M.; Studer, R.: Wissensvernetzung durch Ontologien. In: Pellegrini, T.; Blumauer, A. (Hrsg.): *Semantic Web: Wege zur vernetzten Wissensgesellschaft*. Berlin: Springer-Verlag, 2006, S. 469-484
- [146] Ehrlenspiel, K.: Möglichkeiten zum Senken der Produktkosten. *Erkenntnisse aus einer Auswertung von Wertanalysen*. *Konstruktion* 32(1980) Nr. 5, S. 173-178.

- [147] Ehrlenspiel, K.: Industrieprobleme in Entwicklung und Konstruktion sowie Folgerungen gemäß einer Umfrage. Konstruktion (45)1993, 12, S. 389-396
- [148] Ehrlenspiel, K.: Integrierte Produktentwicklung. Methoden für Prozeßorganisation, Produkterstellung und Konstruktion. München: Carl Hanser Verlag, 1995
- [149] Ehrlenspiel, K.; Meerkamm, H.: Integration versus Spezialisierung – von der Notwendigkeit einer ganzheitlichen Konstruktionsforschung und -lehre an Universitäten und Hochschulen (Teil 1). Konstruktion o. B. (2016), Nr. 1-2, S. 71-74
- [150] Ehrlenspiel, K.; Meerkamm, H.: Integration versus Spezialisierung – von der Notwendigkeit einer ganzheitlichen Konstruktionsforschung und -lehre an Universitäten und Hochschulen (Teil 2). Konstruktion o. B. (2016), Nr. 3, S. 70-73
- [151] Ehrlenspiel, K.; Meerkamm, H.: Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit, München: Carl Hanser Verlag, 2017
- [152] Ehrlenspiel, K.; Kiewert, A.; Lindemann, U.; Mörtl, M.: Kostengünstig Entwickeln und Konstruieren. Kostenmanagement bei der integrierten Produktentwicklung, Berlin: Springer-Verlag, 2014
- [153] Eichinger, M.; Maurer, M.; Pulm, U.; Lindemann, U.: Extending design structure matrices and domain mapping matrices by multiple design structure matrices. Proceedings of 8th Biennial Conference on Engineering Systems Design and Analysis (ESDA 2006), ASME, 2006, S. 889-898
- [154] Eigner, M.: Einleitung – Modellbasierte Virtuelle Produktentwicklung. In: Eigner, M.; Roubanov, D.; Zafirov, R.: Modellbasierte Virtuelle Produktentwicklung. Berlin: Springer-Verlag, 2014, S. 1-13
- [155] Eigner, M.; Stelzer, R.: Product Lifecycle Management. Ein Leitfaden für Product Development und Life Cycle Management. Berlin: Springer-Verlag, 2009
- [156] Eigner, M.; Roubanov, D.; Zafirov, R. (Hrsg.): Modellbasierte virtuelle Produktentwicklung. Berlin: Springer-Verlag, 2014

- [157] Eigner, M.; Dickopf, T.; Apostolov, H.: The Evolution of the V-Model: From VDI 2206 to a System Engineering Based Approach for Developing Cybertronic Systems. In: Ríos, J.; Bernard, A.; Bouras A.; Foufou S.: 14th IFIP WG 5.1 International Conference (PLM 2017), Product Lifecycle Management and the Industry of the Future, IFIP Advances in Information and Communication Technology. Cham: Springer-Verlag, 2017, S. 382-393
- [158] Eisenbart, B.; Blessing, L.; Gericke, K.: Functional modelling perspectives across disciplines. A literature review. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 12th International Design Conference (DESIGN 2012). Glasgow: Design Society, 2012, S. 847-858
- [159] Embst, S.: Komplexitätsmanagement im front end of innovation. Graz: Abteilung für Innovationsmanagement und Unternehmensgründung, Technische Universität Graz, 2010.
- [160] Eppinger, S. D.: Patterns of product development interactions. In Culley, S. (Hrsg.): Proceeding of the 13th International Conference on Engineering Design (ICED2001), The Design Society, 2001, S. 283-290
- [161] Eppinger, S. D.; Browning, T. R.: Design structure matrix methods and applications. Cambridge: The MIT Press, 2012
- [162] Eppler, M. J.; Burkhard, R.: Knowledge Visualization. Towards a New Discipline and its Fields of Application. Lugano: Institute for Corporate Communication, Università della Svizzera italiana, 2004
- [163] Eppler, M. J.; Burkhard, R.: Knowledge Visualization. In: Schwartz, D. G. (Hrsg.): Encyclopedia of Knowledge Management. Hershey: IGI Global, 2006, S. 551-560
- [164] Eppler, M. J.; Seifried, P. M.; Röpnack, A.: Improving knowledge intensive processes through an enterprise knowledge medium. Proceedings of the 1999 ACM SIGCPR conference on Computer personnel research (SIGCPR1999), ACM Press, 1999, S. 222-230
- [165] Erden, M.; Komoto, H.; van Beek, T.; D'Amelio, V.; Echavarria, E.; Tomiyama, T.: A review of function modeling: Approaches and applications. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. (22)2008, Nr. 2, S. 147-169
- [166] Ertel, W.: Grundkurs Künstliche Intelligenz. Eine praxisorientierte Einführung. Wiesbaden: Springer Verlag, 2016

- [167] Estefan, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. In: International Council on Systems Engineering (INCOSE) (Hrsg.): Systems Engineering Body of Knowledge (SEBoK). San Diego: SEBoK/INCOSE, 2008
- [168] Ester, M.; Sander, J.: Knowledge discovery in databases. Techniken und Anwendungen. Berlin: Springer-Verlag, 2000
- [169] Eversheim, W.; Schuh, G.; Assmus, D.: Integrierte Produkt- und Prozessgestaltung. In: Eversheim, W.; Schuh, G. (Hrsg.): Integrierte Produkt- und Prozessgestaltung. Berlin: Springer-Verlag, 2005, S. 5-20
- [170] Falkenberg, E.; Hesse, W.; Lindgreen, P.; Nilsson, B.; Oei, J.; Rolland, C.; Stamper, R.; van Assche, F.; Verrijn-Stuart, A.; Voss, K.: A framework of information system concepts: the FRISCO report (Web edition): Laxenburg: International Federation for Information Processing (IFIP), 1998
- [171] Feess, E.: "Komplexität" auf Gabler Wirtschaftslexikon online. URL: <https://wirtschaftslexikon.gabler.de/definition/komplexitaet-39259/version-262672> (Abrufdatum: 30.05.2020)
- [172] Fei, G.; Gao, J.; Owodunni, O.; Tang, X.: A method for engineering design change analysis using system modelling and knowledge management techniques. International Journal of Computer Integrated Manufacturing, (24)2011, Nr. 6, S. 535-551
- [173] Feldhusen, J.: Gestaltung von Elementen und Systemen. In: Sauer, B. (Hrsg.): Konstruktionselemente des Maschinenbaus 1. Grundlagen der Berechnung und Gestaltung von Maschinenelementen. Berlin: Springer-Verlag, 2016, S. 151-187
- [174] Feldhusen, J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer-Verlag, 2013
- [175] Feldhusen, J.; Grote, K.-H.: Einleitung. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer-Verlag, 2013, S. 5-10
- [176] Feldhusen, J.; Grote, K.-H.; Göpfert, J.; Tretow, G.: Technische Systeme. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer-Verlag, 2013, S. 237-279

- [177] Fernandez-Lopez, M.; Gomez-Perez, A.; Juristo, N.: METHONTOLOGY: from Ontological Art towards Ontological Engineering. 6th Advancement of Artificial Intelligence Spring Symposium (AAAI 1997). Stanford: AAAI Press, 1997, S. 33-40
- [178] Fettke, P.: Kontaktstudium-Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. ZfbF – Schmalenbachs Zeitschrift für Betriebswirtschaftliche Forschung, (61)2009, Nr. 5, S. 550-580
- [179] Fixson, S. K.: A Roadmap for Product Architecture Costing. In: Simpson, T. W.; Siddique, Z.; Jiao, J. R. (Hrsg.): Product Platform and Product Family Design. Methods and Applications. New York: Springer, 2006, S. 305-334
- [180] Flood, R. L.; Carson, E. R.: Dealing with Complexity. An Introduction to the Theory and Application of Systems Science. New York: Springer, 1993
- [181] Frank, U.: Ein Vorschlag zur Konfiguration von Forschungsmethoden in der Wirtschaftsinformatik. In: Lehner, F.; Zelewski, S. (Hrsg.): Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik. Berlin: Gito-Verlag, 2007, S. 158-185
- [182] Frank, U.: Die Konstruktion möglicher Welten als Chance und Herausforderung der Wirtschaftsinformatik. In: Becker, J.; Krcmar, H.; Niehaves, B. (Hrsg.): Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Heidelberg: Physica-Verlag, 2009, S. 161-173
- [183] Frank, U.: Reflexionen zur sprachlichen Konstruktion von Informationssystemen. In: Heinemann, E. (Hrsg.): Anwendungsinformatik. Die Zukunft des Enterprise Engineering. Festschrift für Erich Ortner zum 60. Geburtstag. Baden-Baden: Nomos-Verlag, 2008, S. 37-50
- [184] Frank, U.: Outline of a method for designing domain-specific modelling languages. Essen: ICB-Research Report Nr. 42. Essen: Institut für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen, 2010

- [185] Frank, U.: Multi-Perspective Enterprise Modelling: Background and Terminological Foundation. ICB-Research Report Nr. 46. Essen: Institut für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen, 2011
- [186] Frank, U.: The MEMO meta modelling language (MML) and language architecture. ICB-research report, Nr. 43. Essen: Institut für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen, 2011
- [187] Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. *Software & Systems Modeling*, (13)2012, Nr. 3, S. 941-962
- [188] Frank, U.; van Laak, B.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Koblenz: Arbeitsberichte des Instituts für Wirtschaftsinformatik, Universität Koblenz Landau, 2003
- [189] Franke, W.; Platzer, B.: Rohrleitungen. Grundlagen – Planung – Montage. München: Carl Hanser Verlag, 2014
- [190] Frankenberger, E.: Arbeitsteilige Produktentwicklung: Empirische Untersuchung und Empfehlungen zur Gruppenarbeit in der Konstruktion. Düsseldorf: VDI-Verlag, 1997
- [191] Frese, E.; Graumann, M.; Talaulicar, T.; Theuvsen, L.: Dynamische Sicht der Organisationsgestaltung. In: Frese, E.; Graumann, M.; Talaulicar, T.; Theuvsen, L. (Hrsg.): Grundlagen der Organisation. Entscheidungsorientiertes Konzept der Organisationsgestaltung, Wiesbaden: Springer-Verlag, 2019, S. 485-495
- [192] Freund, J.; Rücker, B.: Praxishandbuch BPMN. Mit Einführung in DMN. München: Carl Hanser Verlag, 2019
- [193] Friedenthal, S.; Moore, A.; Steiner, R.: A Practical Guide to SysML. The Systems Modeling Language. Oxford: Elsevier-Verlag, 2014
- [194] Friedl, G.; Hofmann, C.; Pedell, B.: Kostenrechnung. Eine entscheidungsorientierte Einführung. München: Vahlen-Verlag, 2017
- [195] Friedrich, M. O.: Funktionsorientiertes Konzept zur Unterstützung früher Phasen der Produktentwicklung in der Informationstechnik. Garching/München: Lehrstuhl für Informationstechnik im Maschinenwesen, Technische Universität München (TUM), 2011

- [196] Fruchterman, T. M. J.; Reingold, E. M.: Graph Drawing by Force-directed Placement. *Software-Practice and Experience*, (21)1991, Nr. 11, 1129-1164
- [197] Fu, Y.; Li, M.; Chen, F.: Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. *International Journal of Project Management*, (30)2012, Nr. 3, S. 363-373
- [198] Functional Specs, Inc.: Functional Specs, Inc., 2020, URL: <https://axiomaticdesign.com> (Abrufdatum: 20.07.2020)
- [199] Furrer, F. J.: Eine kurze Geschichte der Ontologie. *Informatik-Spektrum*. (37)2014, Nr. 4, S. 308-317
- [200] Gaag, A.: Entwicklung einer Ontologie zur funktionsorientierten Lösungssuche in der Produktentwicklung. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2010
- [201] Gadatsch, A.: Grundkurs Geschäftsprozess-Management. Analyse, Modellierung, Optimierung und Controlling von Prozessen. Wiesbaden: Springer Fachmedien, 2020
- [202] Gan, G.; Ma, C.; Wu, J.: Data Clustering: Theory, Algorithms, and Applications. Alexandria: American Statistical Association, Philadelphia: Society for Industrial and Applied Mathematics, 2007
- [203] Gangemi, A.; Presutti, V.: Ontology Design Patterns. In: Staab, S.; Studer, R. (Hrsg.): *Handbook on Ontologies*. Berlin: Springer-Verlag, 2009, S. 221-243
- [204] Gangemi, A.; Presutti, V.; Blomqvist, E.: The Computational Ontology Perspective: Design Patterns for Web Ontologies. In: Sartor, G.; Casanovas, P.; Biasiotti, M.; Fernández-Barrera, M. (Hrsg.): *Approaches to Legal Ontologies*. Dordrecht: Springer-Verlag, 2010, S. 201-217
- [205] Gebhardt, M.: Kostenkalkulation im Kontext technischer Produktänderungen: Entwicklung und Evaluation eines kausal-analytischen Ansatzes zur Prognose indirekter Änderungskosten: Institut für Technologie- und Prozessmanagement. Ulm: Institut für Technologie- und Prozessmanagement, Universität Ulm, 2018

- [206] Gebhart, N.; Kruse, M.; Krause, D.: Gleichteile-, Modul- und Plattformstrategie. In: Lindemann, U. (Hrsg.): Handbuch Produktentwicklung. München: Carl Hanser Verlag, 2016, S. 111-149
- [207] Gehrke, B.; Claes, M.: Leadership and global understanding. In: Marques, J.; Dhiman, S. (Hrsg.): Leadership Today. Practices for Personal and Professional Performance. Cham: Springer International Publishing, 2017, S. 371-385
- [208] Geisler, R.; Klar, M.; Pons, C.: Dimensions and dichotomy in metamodeling. Berlin: Fachbereich 13, Informatik, Technische Universität Berlin, Bericht-Nr. 98-5, 1998
- [209] Gell-Mann, M.: What Is Complexity?. In: Curzio, A. Q.; Fortis, M. (Hrsg.): Complexity and Industrial Clusters. Dynamics and Models in Theory and Practice. Heidelberg: Physica-Verlag, 2002, S. 13-24
- [210] Gericke, K.; Blessing, L.: An analysis of design process models across disciplines. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 12th International Design Conference (DESIGN 2012). Glasgow: Design Society, 2012, S. 171-180
- [211] Gericke, K.; Meißner, M.; Paetzold, K.: Understanding the context of product development and the context of product development. In: Lindemann, U.; Srinivasan, V.; Kim, Y. S.; Lee, S. W.; Clarkson, J.; Cascini, G. (Hrsg.): 19th International Conference on Engineering Design (ICED 2013). Glasgow: Design Society, 2013, S. 191-200
- [212] Gero, J. S.; Kannengiesser, U.: Towards a situated Function-Behaviour-Structure framework as the basis of a theory of designing. In: Smithers, T. (Hrsg.): Workshop on Development and Application of Design Theories in AI in Design Research, 6th International Conference on Artificial Intelligence in Design. Dordrecht: Springer-Verlag, 2000, S. 1-5
- [213] Gero, J. S.; Kannengiesser, U.: The situated function-behaviour-structure framework. Design Studies, (25)2003, Nr.4, S. 373-391
- [214] Gero, J. S.; Kannengiesser, U.: A function-behaviour-structure ontology of processes. In: Gero, J. S. (Hrsg.): 2nd International Conference on Design Computing and Cognition (DCC 2006). Dordrecht: Springer-Verlag, 2006, S. 407-422

- [215] Gesellschaft für Systems Engineering (GfSE), Endler, D.; Geisreiter, M.; Kaffenberger, R. & Rambo, J. (Übersetzer): INCOSE Systems Engineering Handbuch. Ein Leitfaden für Systemzyklus-Prozesse und -Aktivitäten: In: International Council on Systems Engineering (INCOSE), Walden, D. D.; Roedler, G. J.; Forsberg, K.; Hamelin, R. D. & Shortell, T. M. (Hrsg. der Originalfassung): INCOSE Systems Engineering Handbook. München: GfSE, Gesellschaft für Systems Engineering e.V., 2017
- [216] Gießmann, M.: Komplexitätsmanagement in der Logistik: Kausal-analytische Untersuchung zum Einfluss der Beschaffungskomplexität auf den Logistikerfolg. Lohmar: Josef Eul Verlag, 2010
- [217] Giones, F.; Brem, A.; Berger, A.: Strategic decisions in turbulent times: Lessons from the energy industry. Business Horizons, Elsevier 62(2019), Nr. 2, S. 215-225
- [218] Goldkuhl, G.; Lind, M.; Seigerroth, U.: Method integration: the need for a learning perspective. IEE Proceedings – Software. (145)1998, Nr. 4, S. 113-118
- [219] Gómez-Pérez, A.; Fernández-López, M.; Corcho, O.: Ontological Engineering. With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. Berlin: Springer Verlag, 2004
- [220] Göpfert, J.: Modulare Produktentwicklung. Wiesbaden: Deutscher Universitätsverlag, 1998
- [221] Göpfert, J.; Tretow, G.: Produktarchitektur. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer-Verlag, 2013, S. 252-279
- [222] Götze, U.: Kostenrechnung und Kostenmanagement. Berlin: Springer-Verlag, 2010
- [223] Grabowski, H.; Lossack, R.-S.; El-Mejbri, E.-F.: Towards A Universal Design Theory. In: Integration of Process Knowledge into Design Support Systems. In: Kals, H.; van Houten, F. (Hrsg.): Integration of Process Knowledge into Design Support Systems. Proceedings of the 1999 CIRP International Design Seminar. Dordrecht: Springer-Verlag, 1999, S. 47-56

- [224] Gronau, N.: Grundlagen der Systemanalyse. In: Krallmann, H. (Hrsg.): Systemanalyse im Unternehmen. Geschäftsprozess-optimierung, partizipative Vorgehensmodelle, objektorientierte Analyse. München: Oldenbourg Verlag, 1994, S. 5-24
- [225] Gross, D.; Hauger, W.; Schröder, J.; Wall, W. A.: Technische Mechanik 2. Elastostatik. Berlin: Springer-Verlag, 2017
- [226] Gross, D.; Hauger, W.; Schröder, J.; Wall, W. A.: Technische Mechanik 1. Statik. Berlin: Springer-Verlag, 2019
- [227] Grote, K.-H.; Bender, B.; Göhlich, D. (Hrsg.): Dubbel. Taschenbuch für den Maschinenbau. Berlin: Springer-Verlag, 2018
- [228] Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition. (5)1993, Nr. 2, S. 199-220
- [229] Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies. (43)1995, Nr. 5-6, S. 907-928
- [230] Grüninger, M.; Fox, M. S.: Methodology for the Design and Evaluation of Ontologies. In: Skuce, D. (Hrsg.): Workshop on Basic Ontological Issues in Knowledge Sharing, in conjunction with 14th International Joint Conference on Artificial Intelligence (IJCAI-95). San Francisco: Morgan Kaufmann Publishers, 1995
- [231] Guarino, N.: Understanding, building and using ontologies. International Journal of Human-Computer Studies. (46)1997, Nr. 2-3, S. 293-310
- [232] Guarino, N.: Formal Ontology and Information Systems. In: Guarino, N. (Hrsg.): 1st Formal Ontology in Information Systems (FOIS 1998). Amsterdam: IOS Press, 1998, S. 3-15
- [233] Guarino, N.; Oberle, D.; Staab, S.: What Is an Ontology? In: Staab, S.; Studer, R. (Hrsg.): Handbook on Ontologies, Berlin: Springer-Verlag, 2009, S. 1-17
- [234] Gutierrez Fernandez, C. I.: Integration Analysis of Product Architecture to support effective team co-location. Massachusetts: Massachusetts Institute of Technology (MIT), 1998
- [235] Haase, P.: Semantic Technologies for Distributed Information Systems. Karlsruhe: Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Karlsruher Institut für Technologie (KIT), 2006

- [236] Hab, G.; Wagner, R.: Projektmanagement in der Automobil-industrie – Effizientes Management von Fahrzeugprojekten entlang der Wertschöpfungskette. Wiesbaden: Springer Gabler Verlag, 2013
- [237] Haberfellner, R.; de Weck, O.; Fricke, E.; Vössner, S.: Systems Engineering. Grundlagen und Anwendung. Zürich: Orell Füssli Verlag, 2012
- [238] Haberfellner, R.; de Weck, O.; Fricke, E.; Vössner, S.: Systems Engineering. Grundlagen und Anwendung. Zürich: Orell Füssli Verlag, 2015
- [239] Haberfellner, R.; de Weck, O.; Fricke, E.; Vössner, S.: Systems Engineering. Fundamentals and Applications. Cham: Springer-Nature Verlag, 2019
- [240] Hagberg, A.; Schult, D.; Swart, P.: NetworkX Reference. Release 2.5, 2020, URL: <https://networkx.github.io/documentation/stable/reference/index.html> (Abrufdatum: 30.08.2020)
- [241] Hahn, A.: Integration verteilter Produktmodelle durch Semantic-Web-Technologien. Wirtschaftsinformatik, (47)2005, Nr. 4, S. 278-284
- [242] Hammond, M.; Robinson, A.: Python Programming on Win32: Help for Windows Programmers. Sebastopol: O'Reilly Media Verlag, 2000
- [243] Hamraz, B.; Caldwell, N. H. M.; Clarkson, P. J.: A Holistic Categorization Framework for Literature on Engineering Change Management. Systems Engineering, (16)2013, Nr. 4, S. 473-505
- [244] Hamraz, B.; Caldwell, N. H. M.; Ridgman, T. W.; Clarkson, P. J.: FBS Linkage ontology and technique to support engineering change management. Research in Engineering Design, (26)2015, Nr. 1, S. 3-35
- [245] Hartmann, J.: Ontologiebasierte Gestaltung und Umsetzung von Wissensmanagementsystemen. Karlsruhe: Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Karlsruher Institut für Technologie (KIT), 2007
- [246] Hassannezhad, M.: Model-Based Support for Management of Engineering Design Processes. Department of Management and Production Engineering, Politecnico di Torino, Turin, 2015

- [247] Hassannezhad, M.; Cantamessa, M.; Montagna, F.: Actor-Based Signposting: A social perspective on modelling design processes. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 13th International Design Conference (DESIGN 2014). Glasgow: Design Society, 2014, S. 841-850
- [248] Hassannezhad, M.; Cantamessa, M.; Montagna, F.: Actor-based Signposting: A Modeling Tool to Improve the Socio-technical Design Processes. In: Weber, C.; Husung, S.; Cascini, G.; Cantamessa, M.; Marjanovic, D.; Rotini, F. (Hrsg.): 20th International Conference on Engineering Design (ICED 2015). Glasgow: Design Society, 2015, S. 1-10
- [249] Hassannezhad, M.; Cantamessa, M.; Montagna, F.; Clarkson, P. J.: Managing Sociotechnical Complexity in Engineering Design Projects. *Journal of Mechanical Design*, (141)2019, Nr. 8, S. 981-998
- [250] Hausmann, K.: *Perimeter – Performanzmessung in der Produktentwicklung auf Basis semantisch integrierter Produktmodelle*. Oldenburg: Abteilung Systemanalyse und -optimierung, Carl von Ossietzky Universität Oldenburg, 2009
- [251] Heisig, P.: Vernachlässigte Potentiale, neue Ansätze – eine Bestandaufnahme. *Wissensmanagement – Das Magazin für Führungskräfte*, 10(2008), Nr. 6, S. 40-41
- [252] Hellenbrand, D.: *Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse*. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2013
- [253] Heller, J. (Hrsg.): *Resilienz für die VUCA-Welt*. Wiesbaden: Springer-Nature Verlag, 2019
- [254] Helmer, R.; Yassine, A.; Meier, C.: Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design*, (21)2010, Nr. 6, S. 647-675
- [255] Herbst, P.: *Methode und Anwendung eines parametrischen Kostenmodells zur frühzeitigen Vorhersage der Produktentstehungskosten*. Paderborn: Heinz Nixdorf Institut, 2014
- [256] Heusel, J.; Bronnhuber, T.: Blechgerecht. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung*. Berlin: Springer-Verlag, 2013, S. 622-648

- [257] Heylighen, F.: The growth of structural and functional complexity during evolution. In: Heylighen, F.; Bollen, J.; Riegler, A. (Hrsg.): The evolution of complexity. The violet book of 'Einstein Meets Magritte', Dordrecht: Kluwer Academic Verlag, 1999, S. 17-44
- [258] Hirao, A.; Koga, T.; Aoyama, K.: Planning support of initial design process based on grouping and ordering of tasks – Design example of an integrated circuit. In: Wynn, D. C.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, J. (Hrsg.): 12th International Design Structure Matrix Conference (DSM 2010). München: Carl Hanser Verlag, 2010, S. 83-96
- [259] Hirao, A.; Oizumi, K.; Aoyama, K.: Finding Parameter Constraint Networks in a Product System. In: Onishi, M.; Maurer, M.; Kirner, K.; Lindemann, U. (Hrsg.): 14th International Dependency and Structure Modeling Conference (DSM 2012). München: Carl Hanser Verlag, 2012, S. 101-110
- [260] Hirtz, J.; Stone, R. B.; McAdams, D. A.; Szykman, S.; Wood, K. L.: A functional basis for engineering design: reconciling and evolving previous efforts. Research in Engineering Design, (13)2002, Nr. 2, S. 65-82
- [261] Hirzel, M.; Geiser, U.; Gaida, I. (Hrsg.): Prozessmanagement in der Praxis. Wertschöpfungsketten planen, optimieren und erfolgreich steuern. Wiesbaden: Gabler Verlag, 2013
- [262] Hitzler, P.; Krötzsch, M.; Parsia, B.; Patel-Schneider, P. F.; Rudolph, S.: OWL 2 Web Ontology Language Primer (Second Edition). W3C Recommendation 11 December 2012, 2012, URL: <http://www.w3.org/TR/2012/REC-owl2-primer-20121211> (Abrufdatum: 30.06.2020)
- [263] Holzer, B.; Stegbauer, C. (Hrsg.): Schlüsselwerke der Netzwerkforschung. Wiesbaden: Springer-Verlag, 2019
- [264] Homburg, C.; Daum, D.: Wege aus der Komplexitäts-kostenfalle. ZWF – Zeitschrift für wirtschaftlichen Fabrikbetrieb, (92)1997. Nr. 7-8, S. 333-337
- [265] Hruschka, P.: Business Analysis und Requirements Engineering. Produkte und Prozesse nachhaltig verbessern. München: Carl Hanser Verlag, 2014
- [266] Hub, H.: Ganzheitliches Denken im Management. Wiesbaden: Gabler Verlag, 1994

- [267] Hubka, V.: Principles of engineering design. London: Butterworth Scientific, 1982
- [268] Hubka, V.: Theorie Technischer Systeme. Grundlagen einer wissenschaftlichen Konstruktionslehre. Berlin: Springer-Verlag, 1984
- [269] Idicula, J.: Planning for Concurrent Engineering. Singapore: Singapore Institute of Manufacturing Technology (SIMTech), 1995
- [270] Ilchner, B.; Singer, R. F.: Werkstoffwissenschaften und Fertigungstechnik. Eigenschaften, Vorgänge, Technologien. Berlin: Springer-Verlag, 2016
- [271] INCOSE: Systems Engineering Vision 2020. Document No.: INCOSE-TP-2004-004-02, Version/Revision: 2.03. San Diego: Technical Operations International Council on Systems Engineering (INCOSE), 2007
- [272] INCOSE: A World in Motion – Systems Engineering Vision 2025. San Diego: Technical Operations International Council on Systems Engineering (INCOSE), 2014
- [273] INCOSE: INCOSE Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities. New York: John Wiley & Sons Verlag, 2015
- [274] INCOSE: Systems Engineering, 2020, URL: <https://www.incose.org/systems-engineering> (Abrufdatum: 25.06.2020)
- [275] Jacobson, A.; Prusak, L.: Wie Informationen wertvoll werden. Harvard Business Manager, 29(2007), Nr. 1, S. 6-7
- [276] Janke, T.: Ontology Engineering Based on Domain Specific Languages and the Application of Ontology Design Patterns. In: Salinesi, C.; Pastor, O.: 22th Conference on Advanced Information Systems Engineering Workshop (CAiSE 2011). Lecture Notes in Business Information Processing, Vol 83. Berlin: Springer-Verlag, 2011, S. 167-176
- [277] Jensen, S.: Erkenntnis – Konstruktivismus – Systemtheorie. Einführung in die Philosophie der konstruktivistischen Wissenschaft. Opladen/Wiesbaden: Westdeutscher Verlag, 1999

- [278] Jiang, P.; Wang, W.; Tan, R.: The use of Dependence Structure Matrix and Su-Field of TRIZ in simplify-ing the complex products. In: Eppinger, S. D.; Maurer, M.; Eben, K.; Lindemann, U. (Hrsg.): 13th International Design Structure Matrix Conference (DSM 2011). München: Carl Hanser Verlag, 2011, S. 337-345
- [279] Jiang, P.; Wang, W.; Tan, R. H.; Sun, J. G.: The use of Dependence Structure Matrix and Su-Field of TRIZ in simplifying the complex products. In: IEEE (Hrsg.): 6th International Conference on Management of Innovation & Technology (ICMIT 2012). Sanur Bali: IEEE, 2012, S. 818-823
- [280] Jockisch, M.; Rosendahl, J.: Klassifikation von Modellen. In: Bandow, G.; Holzmüller, H. (Hrsg.): „Das ist gar kein Modell!“ – Unterschiedliche Modelle und Modellierungen in Betriebswirtschaftslehre und Ingenieurwissenschaften. Wiesbaden: Gabler Springer Verlag, 2010, S. 23-52
- [281] Johansen, B.; Euchner, J.: Navigating the VUCA World. An Interview with Bob Johansen. *Research-Technology Management* 56(2013), Nr. 1, S. 10-15
- [282] Jones, B. B.; Brazzel, M. (Hrsg.): The NTL Handbook of Organization Development and Change. Principles, Practices, and Perspectives. San Francisco: John Wiley & Sons, 2014
- [283] Jörg, M.: Ein Beitrag zur ganzheitlichen Erfassung und Integration von Produktanforderungen mit Hilfe linguistischer Methoden. Aachen: Shaker-Verlag, 2005
- [284] Jünger, M.; Mutzel, P. (Hrsg.): Graph Drawing Software. Berlin: Springer-Verlag, 2004
- [285] Kaffenberger, R.; Schulze, S.-O.; Weber, H.: INCOSE Systems Engineering Handbuch. München: Hanser-Verlag, 2012
- [286] Kaiser, A.: Integriertes Variantenmanagement mit Hilfe der Prozesskostenrechnung. St. Gallen: Institut für Technologie-management, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften sowie Internationale Beziehungen, Universität St. Gallen (HSG), 1995
- [287] Kaiser, S.; Kozica, A.: Über Grenzverschiebungen in der neuen, vernetzten Arbeitswelt. In: Richter, A. (Hrsg.): Vernetzte Organisation. München: Oldenbourg Wissenschaftsverlag, 2014, S. 7-15

- [288] Kamada, T.; Kawai, S.: An algorithm for drawing general undirected graphs. *Information Processing Letters*, (31)1989, Nr. 1, S. 7-15
- [289] Kamps, T.; Mayer, J.: Finden 2.0 – oder: Von der Kunst der effektiven Datensuche. *wissensmanagement – Das Magazin für Führungskräfte* 10(2008) Nr. 7, S. 14-15
- [290] Karagiannis, D.; Höfferer, P.: Metamodels in action: An overview. In: Filipe, J.; Shishkov, B.; Helfert, M.: 1st International Conference on Software and Data Technologies (ICSOFT 2006). Setúbal: Institute for Systems and Technologies of Information, Control and Communication, 2008, S. 27-36
- [291] Karagiannis, D.; Kühn H.: Metamodelling Platforms. In: Bauknecht K.; Tjoa A. M.; Quirchmayr G. (Hrsg.): *E-Commerce and Web Technologies. 3rd International Conference (EC-Web 2002). Lecture Notes in Computer Science*, Vol. 2455. Berlin: Springer-Verlag, 2002, S. 182
- [292] Karagiannis, D.; Visic, N.: Next Generation of Modelling Platforms. In: Grabis J.; Kirikova M. (Hrsg.): 10th International Conference on Business Informatics Research (BIR 2001). *Lecture Notes in Business Information Processing*, Vol. 90. Berlin: Springer-Verlag, 2011, S. 19-28
- [293] Karagiannis, D.; Visic, N.: Platform-as-a-Service (PaaS): The ADOxx Metamodelling Platform. *Information Technologies – A Strategic Priority for the Knowledge Economy*, 2011
- [294] Karagiannis, D.; Buchmann, R. A.; Burzynski, P.; Reimer, U.; Walch, M.: Fundamental Conceptual Modeling Languages in OMiLAB. In: Karagiannis, D.; Mayr, H.; Mylopoulos, J. (Hrsg.): *Domain-Specific Conceptual Modeling. Concepts, Methods and Tools*. Cham: Springer-Verlag, 2016, S. 3-30
- [295] Kasperek, D.: *Structure-based System Dynamics Analysis of Engineering Design Processes*. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2016
- [296] Kastens, U.; Kleine Büning, H.: *Modellierung. Grundlagen und formale Methoden*. München: Carl Hanser Verlag, 2008
- [297] Kauf, F.: *Kostenmanagement*. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung*. Berlin: Springer-Verlag, 2013, S. 124-152

- [298] Kaufmann, M.; Wagner, D. (Hrsg.): Drawing Graphs – Methods and Models. Berlin: Springer-Verlag, 2001
- [299] Kickermann, H.: Rechnerunterstützte Verarbeitung von Anforderungen im methodischen Konstruktionsprozess. Braunschweig: Institut für Konstruktionslehre, Technische Universität Braunschweig, 1995
- [300] Kirchhof, R.: Ganzheitliches Komplexitätsmanagement. Grundlagen und Methodik des Umgangs mit Komplexität im Unternehmen. Wiesbaden: Springer-Verlag, 2003
- [301] Klabunde, S.: Wissensmanagement in der integrierten Produkt- und Prozessgestaltung. Best-Practice-Modelle zum Management von Meta-Wissen. Wiesbaden: Deutscher Universitäts-Verlag, 2003
- [302] Klaus, G.: Wörterbuch der Kybernetik: Band 1. Abbildtheorie – Messwertwandler. Frankfurt a. M.: Fischer-Bücherei, 1969
- [303] Kleiner, S.; Kramer, C.: Entwerfen und Entwickeln mit Systems Engineering auf Basis des RFLP-Ansatzes in V6. In: Stelzer, R.; Grote, K.-H.; Brökel, K.; Rieg, F.; Feldhusen, J. (Hrsg.): Entwerfen – Entwickeln – Erleben: Methoden und Werkzeuge in der Produktentwicklung, 10. Gemeinsames Kolloquium Konstruktions-technik (KT2012). Dresden: TUDpress, 2012, S. 459-474.
- [304] Kleiner, S.; Kramer, C.: Model Based Design with Systems Engineering Based on RFLP Using V6. In: Abramovici, M.; Stark, R. (Hrsg.): Smart Product Engineering. 23rd CIRP Design Conference (CIRP Design 2013). Lecture Notes in Production Engineering. Berlin: Springer-Verlag, 2013, S. 93-102
- [305] Kobler, M.: Qualität von Prozessmodellen. Kennzahlen zur analytischen Qualitätssicherung bei der Prozessmodellierung. Berlin: Logos-Verlag, 2010
- [306] Köbler, J.: Modell einer wandlungsfähigen Organisation produzierender Unternehmen. Heimsheim: Jost-Jetter-Verlag, 2006
- [307] Koch, S.: Einführung in das Management von Geschäftsprozessen – Six Sigma, Kaizen und TQM. Berlin. Springer-Verlag, 2015
- [308] Koh, E. C. Y.: A study on the Requirements to Support the Accurate Prediction of Engineering Change Propagation. Systems Engineering, (20)2017, Nr. 2, S. 147-157

- [309] Koh, E. C. Y.; Caldwell, N. H. M.; Clarkson, P. J.: A method to assess the effects of engineering change propagation. *Research in Engineering Design*, (23)2012, Nr. 4, S. 329-351
- [310] Kohler, J.: Wissenstransfer bei hoher Produkt- und Prozesskomplexität. Pilotierung, Rollout und Migration neuer Methoden am Beispiel der Automobilindustrie. Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler, 2008
- [311] Köhler, P.: CAD-Praktikum für den Maschinen- und Anlagenbau mit PTC Creo. Wiesbaden: Springer-Verlag, 2016
- [312] Köhler, C.; Conrad, J.; Wanke, S.; Weber, C.: A Matrix Representation of the CPM/PDD Approach As a Means for Change Impact Analysis. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 10th International Design Conference (DESIGN 2008). Glasgow: Design Society, 2008, S. 167 – 174
- [313] Kohn, A.: Entwicklung einer Wissensbasis für die Arbeit mit Produktmodellen. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2014
- [314] Kohn, A.; Lutter-Günther, M.; Hagg, M.; Maurer, M.: Handling product information – towards an improved use of product models in engineering design. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 12th International Design Conference (DESIGN 2012). Glasgow: Design Society, 2012, S. 1719-1730
- [315] Koller, R.: Konstruktionslehre für den Maschinenbau. Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen. Berlin: Springer-Verlag, 1998
- [316] König, C.; Kreimeyer, M.; Braun, T.: Multiple-Domain Matrix as a Framework for Systematic Process Analysis. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (Hrsg.): 10th International DSM Conference (DSM 2008). München: Carl Hanser-Verlag, 2008, S. 231-244
- [317] Koppenhagen, F.: Modulare Produktarchitekturen – Komplexitätsmanagement in der frühen Phase der Produktentwicklung. In: Schoeneberg, K.-P. (Hrsg.): Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern, Wiesbaden: Springer Gabler Verlag, 2014, S. 113-16

- [318] Koren, Y.: Drawing Graphs by Eigenvectors: Theory and Practice. *Computers & Mathematics with Applications*, (49)2005, Nr. 11, S. 1867-1888
- [319] Korff, A.; Schacher, M.: Modellierung von eingebetteten Systemen mit UML und SysML. Heidelberg: Springer Spektrum, 2008
- [320] Kößler, J.; Paetzold, K.: Integration of MBSE into existing development processes - Expectations and challenges: In: Maier, A.; Škec, S.; Kim, H.; Kokkolaras, M.; Oehmen, J.; Fadel, G.; Salustri, F.; Van der Loos, M. (Hrsg.): 21st International Conference on Engineering Design (ICED 2017). Glasgow: Design Society, 2017, S. 51-60
- [321] Krallmann, H.; Bobrik, A.; Levina, O. (Hrsg.): Systemanalyse im Unternehmen. Prozessorientierte Methoden der Wirtschaftsinformatik. München: Oldenbourg Verlag, 2013
- [322] Krause, F.-L.; Franke, H.-J.; Gausemeier, J.: Innovationspotenziale in der Produktentwicklung. München: Carl Hanser Verlag, 2007
- [323] Krehmer, H.: Vorgehensmodell zum Iterations- und Produktreife-gradmanagement in der eigenschaftsbasierten Produktentwicklung. Düsseldorf: VDI-Verlag, 2012
- [324] Krehmer, H.; Eckstein, R.; Lauer, W.; Roelofsen, J.; Stöber, C.; Troll, A.; Weber, N.; Zapf, J.: Coping with multidisciplinary product development – a process model approach. In: Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann, U. (Hrsg.): 17th International Conference on Engineering Design (ICED 2009). Glasgow: Design Society, 2009, S. 241-252
- [325] Krehmer, H.; Eckstein, R.; Lauer, W.; Roelofsen, J.; Stöber, C.; Troll, A.; Zapf, J.; Weber, N.; Meerkamm, H.; Henrich, A.; Lindemann, U.; Rieg, F.; Wartzack, S.: Das FORFLOW-Prozessmodell zur Unterstützung der multidisziplinären Produktentwicklung. *Konstruktion*, (10)2010, S. 59-68
- [326] Kreimeyer, M. F.: A Structural Measurement System for Engineering Design Processes. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2010
- [327] Kreimeyer, M.; Lindemann, U.: Complexity Metrics in Engineering Design. Managing the Structure of Design Processes. Berlin: Springer-Verlag, 2011

- [328] Kremer, D.; Leyh, J.: Anforderungen an Produktentwicklungsteams. In: Bertsche, B.; Bullinger, H.-J. (Hrsg.): Entwicklung und Erprobung innovativer Produkte. Rapid Prototyping. Berlin: Springer-Verlag, 2007, S. 34-39
- [329] Krischke, A.; Röpcke, H.: Graphen und Netzwerktheorie – Grundlagen – Methoden – Anwendungen. München: Carl Hanser Verlag, 2014
- [330] Kügler, P.; Marian, M.; Schleich, B.; Tremmel, S.; Wartzack, S.: tribAln - Towards an Explicit Specification of Shared Tribological Understanding. Applied Sciences, (10)2020, Nr. 13, S. 1-21
- [331] Kühn, H.; Murzek, M.: Interoperability issues in metamodeling platforms. In: Konstantas, D.; Bourrières, J.; Léonard, M.; Boudjlida, N. (Hrsg.): Interoperability of Enterprise Software and Applications. London: Springer-Verlag, 2006, S. 215-226
- [332] Kuhrmann, M.; Ternité, T.; Friedrich, J.: Das V-Modell® XT anpassen. Anpassung und Einführung kompakt für V-Modell® XT Prozessingenieure. Berlin: Springer-Verlag, 2011
- [333] Kumar, P.: Design process modelling: towards an ontology of engineering design activities. Clemson: Department of Mechanical Engineering, Clemson University, 2008
- [334] Kumar, P.; Mocko, G.: Modeling and Analysis of an Ontology of Engineering Design Activities Using the Design Structure Matrix. In: ASME (Hrsg.): ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE 2007), 9th International Conference on Design Theory and Methodology. Las Vegas: ASME, S. 559-571
- [335] Kunrath, K.; Cash, P.; Kleinsmann, M.: Social- and self-perception of designers' professional identity. Journal of Engineering Design, 31(2019), Nr. 2, S. 100-126
- [336] Kunrath, K.; Cash, P.; Kleinsmann, M.: Designers' professional identity: personal attributes and design skills. Journal of Engineering Design, 31(2020), Nr. 6, S. 1-34
- [337] Kurbel, K.: Entwicklung und Einsatz von Expertensystemen. Eine anwendungsorientierte Einführung in wissensbasierte Systeme. Berlin: Springer-Verlag, 1992

- [338] Kurtev, I.; Bézivin, J.; Jouault, F.; Valduriez, P.: Model-based DSL frameworks. In: Tarr, P.; Cook, W.: 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications (OOPSLA 2006). New York: Association for Computing Machinery, 2006, S. 602-616
- [339] Kusiak, A.: Interface Structure Matrix for analysis of products and processes. In: CIRP (Hrsg.): 15th CIRP International Conference on Life Cycle Engineering (LCE 2008). Sydney: CIRP, 2008, S. 444-448
- [340] Kusiak, A.; Larson, T. N.; Wang, J.: Reengineering of design and manufacturing processes. Computers & Industrial Engineering, (26)1994, Nr. 3, S. 521-536
- [341] Lashin, G.; Feldhusen, J.; Grote, K.-H.: Hilfsmittel für die Entwicklung und Konstruktion. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer-Verlag, 2013, S. 411-457
- [342] Lattix Inc.: Lattix, 2020, URL: <https://www.lattix.com> (Abrufdatum: 20.07.2020)
- [343] Lauer, W. M.: Integrative Dokumenten- und Prozessbeschreibung in dynamischen Produktentwicklungsprozessen. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2010
- [344] Laux, H.; Liermann, F.: Grundlagen der Organisation. Die Steuerung von Entscheidungen als Grundproblem der Betriebswirtschaftslehre. Berlin: Springer-Verlag, 2005
- [345] Lehner, F.: Wissensmanagement. Grundlagen, Methoden und technische Unterstützung. München: Carl Hanser Verlag, 2014
- [346] Lenz, U.: Coaching im Kontext der VUCA-Welt. Der Umbruch steht bevor. In: Heller, J. (Hrsg.): Resilienz für die VUCA-Welt. Wiesbaden: Springer Fachmedien, 2019, S. 49-68
- [347] Leopold, T. A.; Ratcheva, V.; Zahidi, S.: The Future of Jobs. Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution. Global Challenge Insight Report. Cologne: World Economic Forum, 2016
- [348] Leopold, T. A.; Ratcheva, V. & Zahidi, S.: The Future of Jobs Report – 2018. Insight Report. Cologne: World Economic Forum, 2018

- [349] Li, B.; Xie, S.: Module partition for 3D CAD assembly models: a hierarchical clustering method based on component dependencies. *International Journal of Production Research*, (53)2015, Nr. 17, S. 5224-5240
- [350] Lindemann, U.: *Methodische Entwicklung technischer Produkte. Methoden flexibel und situationsgerecht anwenden*. Berlin: Springer-Verlag, 2009
- [351] Lindemann, U.: *Handbuch Produktentwicklung*. München: Carl Hanser Verlag, 2016
- [352] Lindemann, U.; Kiewert, A.: Kostenmanagement im Entwicklungsprozess. Marktgerechte Kosten durch Target Costing. In: Schäppi, B.; Andreasen, M.; Kirchgeorg, M.; Radermacher, F.-J. (Hrsg.): *Handbuch Produktentwicklung*. München: Carl Hanser Verlag, 2005, S. 397-417
- [353] Lindemann, U.; Maurer, M.: Facing multi-domain complexity in product development. In: Krause, F. L. (Hrsg.): *The future of product development. 17th CIRP Design Conference (CIRP Design 2007)*. Berlin: Springer, 2007, S. 351-361
- [354] Lindemann, U.; Reichwald, R.; Zäh, M.: Einführung. In: Lindemann, U.; Reichwald, R.; Zäh, M. (Hrsg.): *Individualisierte Produkte. Komplexität beherrschen in Entwicklung und Produktion*. Berlin: Springer-Verlag, 2006, S. 1-6
- [355] Lindemann, U.; Maurer, M.; Braun, T.: *Structural Complexity Management. An Approach for the Field of Product Design*. Berlin: Springer-Verlag, 2009
- [356] Livingston, R. E.: The Future of organization development in a VUCA World. In: Jones, B. B.; Brazzel, M. (Hrsg.): *The NTL Handbook of Organization Development and Change. Principles, Practices, and Perspectives*. San Francisco: John Wiley & Sons, 2014, S. 659 – 672
- [357] Redpoint.Teseon AG: Loomeo, 2020, URL: <https://loomeo.com> (Abrufdatum: 20.07.2020)
- [358] Lossack, R.-S.: *Wissenschaftstheoretische Grundlagen für die rechnerunterstützte Konstruktion*. Berlin: Springer-Verlag, 2006

- [359] Lossack, R.; Grabowski, H.: The axiomatic approach in the universal design theory. In: Tate, D. (Hrsg.): 1st International Conference on Axiomatic Design (ICAD 2000). Woburn: Functional Specs, Inc., 2000, S. 203-210
- [360] Luczak, H.; Fricker, A.: Komplexitätsmanagement – ein Mittel der strategischen Unternehmensgestaltung. In: Schuh, G.; Wiendahl, H.-P. (Hrsg.): Komplexität und Agilität. Steckt die Produktion in der Sackgasse?. Berlin: Springer-Verlag, 1997, S. 309-323
- [361] Luhmann, N.: Komplexität. In: Grochla, E. (Hrsg.): Handwörterbuch der Organisation. Stuttgart: Poeschel-Verlag, 1980, Sp. 1064-1070
- [362] Mack, O.; Khare, A.: Perspectives on a VUCA World. In: Mack, O.; Khare, A.; Krämer, A.; Burgartz, T. (Hrsg.): Managing in a VUCA World. Cham: Springer International Publishing, 2016, S. 3-19
- [363] Mädche, A.; Staab, S.; Studer, R.: Ontologien. Wirtschaftsinformatik, (43)2001, Nr. 4, S. 393-395
- [364] Madni, A.; Purohit, S.: Economic Analysis of Model-Based Systems Engineering. Systems, (7)2019, Nr. 1, S. 1-18
- [365] Maier, J. F.; Wynn, D. C.; Biedermann, W.; Lindemann, U.; Clarkson, P. J.: Simulating progressive iteration, rework and change propagation to prioritise design tasks. Research in Engineering Design, (25)2014, Nr. 4, S. 283-307
- [366] Maier, I.; Kurz, L.; Bauer, W.: Verkürzung des Produktentwicklungsprozesses durch agile Entwicklungsmethoden. In: Binz, H.; Bertsche, B.; Bauer, W.; Spath, D.; Roth, D. (Hrsg.): 4. Stuttgarter Symposium für Produktentwicklung (SSP 2017). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), 2017, o. S.
- [367] Maisenbacher, S.: Integrated Value Engineering - Leitfaden zur integrierten Betrachtung von Produktwerten. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2018
- [368] Malik, F.: Strategie des Managements komplexer Systeme. Ein Beitrag zur Management-Kybernetik evolutionärer Systeme. Bern: Haupt Verlag, 2008

- [369] Martin, M. V.: Design for variety: a methodology for developing product platform architectures. Stanford: Department of Mechanical Engineering, Stanford University, 1999
- [370] Martin, M. V.; Ishii, K.: Design for variety: development of complexity indices and design charts. In: Proceedings of Design Engineering Technical Conference (DETC 1997). ASME Paper No. DETC97/DFM-4359, ASME, 1997, o. S.
- [371] Martin, M. V.; Ishii, K.: Design for variety: developing standardized and modularized product platform architectures. Research in Engineering Design. 13(2002), Nr. 4, S. 213-235
- [372] Seven years of product development in Industry – experiences and requirements for supporting engineering design with ‘Thinking Tools’. In: Culley, S. J.; Hicks, B. J.; McAloone, T. C.; Howard T. J.; Cantemessa, M. (Hrsg.): 18th International Conference on Engineering Design (ICED 2011). Glasgow: Design Society, 2011, S. 236-245
- [373] Matthiesen, S.; Albers, A.; Moeser, G.; Schmidt, S.; Bursac, N.: Neue Lehrkonzepte für die Aus- und Weiterbildung im Bereich MBSE. In: K. Kohlhammer, K. (Hrsg.): develop3 systems engineering – Interdisziplinäre Produktentwicklung in der Praxis. Leinfelden-Echterdingen: Konradin-Verlag, 2016, S. 26-30
- [374] Mattmann, I.: Modellintegrierte Produkt- und Prozessentwicklung. Wiesbaden: Springer Fachmedien, 2017
- [375] Maune, G.: Möglichkeiten des Komplexitätsmanagements für Automobilhersteller auf Basis IT-gestützter durchgängiger Systeme. Paderborn: Professur für Betriebswirtschaftslehre, insbesondere Produktionswirtschaft, Universität Paderborn, 2002
- [376] Maurer, M.: Structural awareness in complex product design. Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2007
- [377] Maurer, M.: Process optimization by DSM-based modelling of inputs and outputs. In: Culley, S. J.; Hicks, B. J.; McAloone, T. C.; Howard T. J.; Cantemessa, M. (Hrsg.): 18th International Conference on Engineering Design (ICED 2011). Glasgow: Design Society, 2011, S. 24-34
- [378] Maurer, M.: Complexity Management in Engineering Design – a Primer. Berlin: Springer-Verlag, 2017

- [379] Maximini, D.: Scrum – Einführung in der Unternehmenspraxis. Von starren Strukturen zu agilen Kulturen. Berlin: Springer-Verlag, 2018
- [380] Meerkamm, H.; Henrich, A.; Jablonksi, S.; Krcmar, H.; Lindemann, U.; Rieg, F. (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten – Navigation. Bayerischer Forschungsverbund FORFLOW für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung, Abschlussbericht. Aachen: Shaker-Verlag, 2009
- [381] Meffert, H.: Marketing. Grundlagen marktorientierter Unternehmensführung. Konzepte – Instrumente – Praxisbeispiele. Wiesbaden: Springer Gabler Verlag, 1998
- [382] Meffert, H.; Burmann, C.; Kirchgeorg, M.; Eisenbeiß, M.: Marketing. Grundlagen marktorientierter Unternehmensführung. Konzepte – Instrumente – Praxisbeispiele. Wiesbaden: Springer Gabler Verlag, 2019
- [383] Mehlstäubl, J.; Atzberger, A.; Paetzold, K.: General approach to support modelling of data and information flows in product development. In: Mortensen, N.; Hansen, C.; Deininger, M. (Hrsg.): 14th Norddesign Conference (NordDesign 2020). Glasgow: Design Society, 2020, o. S.
- [384] Meis, J.-F.: Produktionsseitiges Anforderungsmanagement. München: Lehrstuhl für Betriebswissenschaften und Montagetechnik, Technische Universität München (TUM), 2017
- [385] Mencken, H. L.: Prejudices: second series. New York: Alfred A. Knopf, 1920
- [386] Meran, R.; John, A.; Staudter, C.; Roenpage, O.: Six Sigma+Lean Toolset. Mindset zur erfolgreichen Umsetzung von Verbesserungsprojekten. Berlin: Springer-Verlag, 2014
- [387] Micouin, P.: Model based systems engineering: fundamentals and methods. London: ISTE Ltd Verlag, Hoboken: John Wiley & Sons Verlag, 2014
- [388] Milton, N. R.: Knowledge Acquisition in Practice. A Step-By-Step Guide. London: Springer-Verlag, 2007
- [389] Minonne, C.; Colicchio, C.; Litzke, M.; Keller, T.: Business-Process-Management 2011 – Status quo und Zukunft eine empirische Studie im deutschsprachigen Raum. Zürich: vdf Hochschulverlag, 2011

- [390] de Miroschedji, S. A.: Globale Unternehmens- und Wertschöpfungsnetzwerke. Grundlagen – Organisation – Gestaltung. Wiesbaden: Deutscher Universitätsverlag, 2002
- [391] Möller, R.; Haarslev, V.: Description Logic Systems. In: Baader, F.; Calvanese, D.; McGuinness, D.; Nardi D.; Patel-Schneider, P. (Hrsg.): The Description Logic Handbook: Theory, Implementation, and Applications. New York: Cambridge University Press, 2010, S. 304-328
- [392] Moody, D.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, (55)2005, Nr. 3, S. 243-276
- [393] Mortensen, N.: Design modelling in a designer's workbench: contribution to a design language. Lyngby/ Kopenhagen: Department of Control and Engineering Design, Technical University of Denmark (DTU), 1999
- [394] Mörtl, M.: Kostenrechnung in der Konstruktion. In: Rieg, F.; Steinhilper, R. (Hrsg.): Handbuch Konstruktion. München: Carl Hanser Verlag, 2018, S. 769-789
- [395] Mörtl, M.; Schmied, C.: Design for Cost – A Review of Methods, Tools and Research Directions. *Journal of the Indian Institute of Science*, (95)2015, Nr. 4, S. 379-404
- [396] Motta, E.: The knowledge modeling paradigm in knowledge engineering. In: Chang, S. K. (Hrsg.): Handbook of Software Engineering and Knowledge Engineering, River Edge: World Scientific Publishing Company, 2001, S. 589-613
- [397] Müller, S.; Kaiser, A.: Was kostet eine Produktvariante? Sonderdruck Technische Rundschau. o. V.(1995), o. N., S. 31-35
- [398] Nagarajah, A.: Methodisches Klären und Präzisieren der Aufgabenstellung. In: Feldhusen J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer Vieweg Verlag, 2013, S. 319-341
- [399] Nandram, S. S.; Bindlish, P. K.: Introduction to VUCA. In: Nandram, S. S.; Bindlish, P. K. (Hrsg.): Managing VUCA Through Integrative Self-Management. How to Cope with Volatility, Uncertainty, Complexity and Ambiguity in Organizational Behavior. Cham: Springer International Publishing, 2017, S. 3-14

- [400] Nardi, D.; Brachman, R. J.: An Introduction to Description Logics. In: Baader, F.; Calvanese, D.; McGuinness, D.; Nardi D.; Patel-Schneider, P. (Hrsg.): The Description Logic Handbook: Theory, Implementation, and Applications. New York: Cambridge University Press, 2010, S. 1-44
- [401] Negele, H.: Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung. München: Herbert Utz Verlag, 1998
- [402] Neto Souza, A.: Antares DSM: Visualização e Otimização de Dependências em Design Structures Matrices. Salvador: Department of Computer Science, Federal University of Bahia, 2008
- [403] Neto Souza, A.: Antares DSM: Visualization and Optimization Dependencies on Design Structures Matrices. Salvador: Department of Computer Science, Federal University of Bahia, 2008
- [404] Niazi, A.; Dai, J. S.; Balabani, S.; Seneviratne, L.: Product Cost Estimation: Technique Classification and Methodology Review. Journal of Manufacturing Science and Engineering, (128)2006, Nr. 2, S. 563-575
- [405] Nicola, A.; Missikoff, M.; Navigli, R.: A Proposal for a Unified Process for Ontology Building: UPON. In: Andersen, K.; Debenham, J.; Wagner, R. (Hrsg.): 13th International Conference on Database and Expert Systems Applications (DEX 2005). Lecture Notes in Computer Science, Vol. 3588. Berlin: Springer-Verlag, 2005, S. 655-664
- [406] Niemann, G.; Winter, H.; Höhn, B.-R.; Stahl, K.: Maschinenelemente 1. Konstruktion und Berechnung von Verbindungen, Lagern, Wellen. Berlin: Springer-Verlag, 2019
- [407] Nitzsche, M.: Graphen für Einsteiger. Rund um das Haus vom Nikolaus. Wiesbaden: Vieweg+Teubner Verlag, 2009
- [408] Nonaka, I.; Takeuchi, H.: The Knowledge-Creating Company. How Japanese Companies Create the Dynamics of Innovation. New York: Oxford University Press, 1995
- [409] North, K.: Wissensorientierte Unternehmensführung. Wissensmanagement gestalten. Wiesbaden: Springer Gabler Verlag, 2016

- [410] Noy, N.; McGuinness, D.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford: Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001
- [411] Oberle, D.: *Semantic Management of Middleware*. Boston: Springer-Verlag, 2006
- [412] Oxford: Oxford University Press: *Oxford English Dictionary (OED)*, 2020, URL: <https://www.oed.com> (Abrufdatum: 08.08.2020)
- [413] Ogden, C.; Richards, I.: *The meaning of meaning: a study of the influence of language upon thought and of the science of symbolism*. New York: Harcourt Brace & World Verlag, 1923
- [414] Oizumi, K.; Aoyama, K.: *Functional Structure Based Change Assessment in Product Design*. In: Onishi, M.; Maurer, M.; Kirner, K.; Lindemann, U. (Hrsg.): *14th International DSM Conference (DSM 2012)*. München: Carl Hanser Verlag, 2012, S. 167–180
- [415] Oizumi, K.; Aoyama, K.: *Inventive product design focusing on physical causal relationships causing trade-offs between functions*. In: Marjanović, D.; Štorga, M.; Škec, S.; Bojčetić, N.; Pavković, N. (Hrsg.): *15th International Design Conference (DESIGN 2018)*. Glasgow: Design Society, 2018, S. 1279–1288
- [416] Oizumi, K.; Kitajima, K.; Yoshie, N.; Koga, T.; Aoyama, K.: *Management of product development projects through integrated modeling of product and process information*. In: Culley, S. J.; Hicks, B. J.; McAloone, T. C.; Howard T. J.; Cantemessa, M. (Hrsg.): *18th International Conference on Engineering Design (ICED 2011)*. Glasgow: Design Society, 2011, S. 253–263
- [417] Oldham, K.; Kneebone, S.; Callot, M.; Murton, A.; Brimble, R.: *MOKA – A Methodology and tools Oriented to Knowledge-based engineering Applications*. In: Mårtensson, N.; Mackay, R.; Björgvinsson, S. (Hrsg.): *Changing the Ways We Work*. Amsterdam, IOS Press, 1998, S. 198–207
- [418] OMG: *OMG Systems Modeling Language (OMG SysML™)*. Version 1.6. Needham: *An OMGSysML Modeling Language™ Publication*, 2019

- [419] Österle, H. & Gutzwiller, T.: Konzepte angewandter Analyse- und Design-Methoden. Band 1: Ein Referenz-Metamodell für die Analyse und das System-Design. Hallbergmoos: Angewandte-Informationen-Technik-Verlag, 1992
- [420] Paech, B.: Plädoyer für ein einheitliches Grundgerüst bei der System- und Softwaremodellierung. In: Pohl, K.; Schürr, A. & Vossen, G. (Hrsg.): GI-Workshop Modellierung. Aachen: CEUR Workshop Proceedings, Vol. 9, 1998, S. 10-14
- [421] Paetzold, K.: Product and Systems Engineering/CA* Tool Chains. In: Biffel, S.; Lüder, A.; Gerhard, D. (Hrsg.): Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects. Cham: Springer International Publishing, 2017, S. 27-62
- [422] Pahl, G.; Beitz, W.: Konstruktionslehre. Handbuch für Studium und Praxis. Berlin: Springer-Verlag, 1977
- [423] Pandremenos, J.; Chryssolouris, G.: A neural network approach for the development of modular product architectures. International Journal of Computer Integrated Manufacturing, (24)2011, Nr. 10, S. 879-887
- [424] Pasqual, M. C.; de Weck, O. L.: Multilayer network model for analysis and management of change propagation. Research in Engineering Design, (23)2011, Nr. 4, S. 305-328
- [425] Patzak, G.: Systemtechnik. Planung komplexer innovativer Systeme. Berlin: Springer-Verlag, 1982
- [426] Pektaş, S.: Representing information flow in building design process using the parameter-based design structure matrix. Ankara: Institute of Economics and Social Sciences, Bilkent-Universität, 2003
- [427] Pektaş, Şu. T.; Pultar, M.: Modelling detailed information flows in building design with the parameter-based design structure matrix. Design Studies, (27)2006, Nr. 1, S. 99-122
- [428] Pepels, W. (Hrsg.): Organisationsgestaltung in marktorientierten Unternehmen. Heidelberg: Sauer-Verlag, 2001

- [429] Peters, W.: Zur Theorie der Modellierung von Natur und Umwelt: Ein Ansatz zur Rekonstruktion und Systematisierung der Grundperspektiven ökologischer Modellbildung für planungsbezogene Anwendungen. Berlin: Institut für Management in der Umweltplanung, Technischen Universität Berlin, 1998
- [430] Pfeffer, J.: Produkt-Entwicklung: Lean & Agile. München: Carl Hanser Verlag, 2019
- [431] Pfläging, N.; Hermann, S.: Komplexithoden: Clevere Wege zur (Wieder)Belebung von Unternehmen und Arbeit in Komplexität. München: Redline Verlag, 2015
- [432] Piller, F. T.; Waringer, D.: Modularisierung in der Automobilindustrie – neue Formen und Prinzipien. Modular Sourcing, Plattformkonzept und Fertigungssegmentierung als Mittel des Komplexitätsmanagements. Aachen: Shaker-Verlag, 1999
- [433] Pimmler, T. U.; Eppinger, S. D.: Integration analysis of product decompositions. In: AMSE (Hrsg.): ASME Design Theory and Methodology Conference. New York: ASME, 1994, S. 343-351
- [434] Pinto, H. S.; Staab, S.; Tempich, C.: DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvinG Engineering of oNTologies. In: de Mantaras, R.; Saitta, L. (Hrsg.): 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia: IOS Press, 2004, S. 393-397
- [435] Poggi, A.; Lembo, D.; Calvanese, D.; Giacomo, G.; Lenzerini, M.; Rosati, R.: Linking Data to Ontologies. In: Spaccapietra S. (Hrsg.): Journal on Data Semantics X. Lecture Notes in Computer Science, Vol. 4900. Berlin: Springer-Verlag, 2008, S: 133-173
- [436] Pohl, K.: Requirements Engineering. Grundlagen, Prinzipien, Techniken. Heidelberg: Dpunkt-Verlag, 2008
- [437] Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Systematisch von Anforderungen zu Konzepten und Gestaltlösungen. Berlin: Springer-Verlag, 2011
- [438] Pribilla, P.: Telekommunikation im Management. Strategien für den globalen Wettbewerb. Stuttgart: Schäffer-Poeschel-Verlag, 1996

- [439] Probst, G.; Raub, S.; Romhardt, K.: Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Wiesbaden: Springer Gabler, 2012
- [440] Parametric Technology Corporation: Creo®Parametric 2.0 VB API User's Guide. Needham: Parametric Technology Corporation, 2012
- [441] Pubanz, H.: Komplexitätscontrolling als Baustein eines effektiven Komplexitätsmanagements. Controlling – Zeitschrift für erfolgsorientierte Unternehmenssteuerung. 25(2013), Nr. 2, S. 68-74
- [442] Puhl, H.: Komplexitätsmanagement: Ein Konzept zur ganzheitlichen Erfassung, Planung und Regelung der Komplexität in Unternehmensprozessen. Kaiserslautern: Lehrstuhl für Fertigungstechnik und Betriebsorganisation, Technische Universität Kaiserslautern, 1999
- [443] Puhl, H.; Rauch, C.; Hiller, M.: Geschäftsprozessoptimierung – Komplexitätsbewusste Geschäftsprozessoptimierung. ZWF – Zeitschrift für wirtschaftlichen Fabrikbetrieb. 93 (1998), Nr. 7-8, S. 349-352
- [444] Pulm, U.: Eine systemtheoretische Betrachtung der Produktentwicklung. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2004
- [445] Puppe, T.: Spectral Graph Drawing. Konstanz: Fachbereich Informatik und Informationswissenschaft, Universität Konstanz, 2005
- [446] Purchase, H. C.; Carrington, D.; Alder, J.-A.: Empirical Evaluation of Aesthetics-based Graph Layout. Empirical Software Engineering, (7)2002, Nr. 3, S. 233-255
- [447] Rathnow, P.: Integriertes Variantenmanagement: Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt. Göttingen: Vandenhoeck & Ruprecht-Verlag, 1993
- [448] Reddy, E. J.; Sridhar, C. N. V.; Rangadu, P.: Knowledge Based Engineering: Notion, Approaches and Future Trends. American Journal of Intelligent Systems, (5)2015, Nr. 1, S. 1-17
- [449] Reichenberger, K.: Kompendium semantische Netze. Konzepte, Technologie, Modellierung. Berlin: Springer-Verlag, 2010

- [450] Reinhart, G.; Mößmer, H.: Wirtschaftliche Aspekte im Qualitätsmanagement. In: Reinhart, G.; Lindemann, U.; Heinzl, J. (Hrsg.): Qualitätsmanagement. Ein Kurs für Studium und Praxis. Berlin: Springer-Verlag, 1996, S. 259-280
- [451] Reiß, M.: Komplexitätsmanagement (I). WISU – Das Wirtschaftsstudium. (22)1993, Nr. 1, S. 54-60
- [452] Reiß, M.: Komplexitätsmanagement (II). WISU – Das Wirtschaftsstudium. 22(1993), Nr. 2, S. 132-137
- [453] Rinkevich, D. J.; Samson, F. P.: An improved powertrain attributes development process with the use of design structure matrix. Cambridge: System Design & Management Program, Massachusetts Institute of Technology (MIT), 2004
- [454] Ripperda, S.: Methodische Unterstützung zur kostenbasierten Auswahl modularer Produktstrukturen. Berlin: Springer-Verlag, 2019
- [455] Rodenacker, W.: Methodisches Konstruieren. Grundlagen, Methodik, praktische Beispiele. Berlin: Springer-Verlag, 1991
- [456] Rogers, J.: DeMAID/GA User's guide. Design Manager's Aid for Intelligent Decomposition with a genetic algorithm. Hampton: Hampton: Langley Research Center, National Aeronautics and Space Administration (NASA), 1996
- [457] Rohde, D.: Globales Komplexitätsmanagement – eine Methode zur integrierten Komplexitätsbewältigung globaler Wertschöpfungsarchitekturen. St. Gallen: Lehrstuhl für Produktionsmanagement, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften sowie Internationale Beziehungen, Universität St. Gallen (HSG), 2019
- [458] Ropohl, G.: Allgemeine Technologie. Eine Systemtheorie der Technik. Karlsruhe: Universitätsverlag Karlsruhe, 2009
- [459] von Rosing, M.; Kemp, N.; Hove, M.; Ross, J.: Process Tagging – A Process Classification and Categorization Concept. In: von Rosing, M.; von Scheel, H.; Scheer, A.-W. (Hrsg.): The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM, Vol. I, Oxford: Elsevier Verlag, 2015, S. 123-171
- [460] Roth, K.: Konstruieren mit Konstruktionskatalogen. Band 3: Verbindungen und Verschlüsse, Lösungsfindung. Berlin: Springer-Verlag, 1996

- [461] Roth, K.: Konstruieren mit Konstruktionskatalogen. Band 1: Konstruktionslehre. Berlin: Springer-Verlag, 2000
- [462] Roth, K.: Konstruieren mit Konstruktionskatalogen. Band 2: Kataloge. Berlin: Springer-Verlag, 2001
- [463] Roth, D.: Analyse und Bewertung von Wissen in der Produktentwicklung. Stuttgart: Institut für Konstruktionstechnik und Technisches Design, Universität Stuttgart, 2020
- [464] Rude, S.: Wissensbasiertes Konstruieren. Karlsruhe: Fakultät für Maschinenbau, Universität Fridericiana Karlsruhe, 1998
- [465] Rupp, C.; Queins, S.; die SOPHISTen: UML 2 glasklar: Praxiswissen für die UML-Modellierung. München: Carl Hanser Verlag, 2012
- [466] Russell-Walling, E.: Systemedenken. In: Russell-Walling, E. (Hrsg.): 50 Schlüsselideen Management. Heidelberg: Spektrum Akademischer Verlag, 2011, S. 172-175
- [467] Sandkuhl, K.; Wißotzki, M.; Stirna, J.: Unternehmensmodellierung. Grundlagen, Methode und Praktiken. Berlin: Springer Vieweg Verlag, 2013
- [468] Schatz, A.; Schöllhammer, O.; Jäger, J.: Ansatz zum Umgang mit Komplexität in Unternehmen. Controlling – Zeitschrift für erfolgsorientierte Unternehmenssteuerung. 26(2014), Nr. 12, S. 686-693
- [469] Scheer, A.-W.: ARIS - Vom Geschäftsprozess zum Anwendungssystem. Berlin: Springer-Verlag, 2002
- [470] Scheier, F.; Schramm, F.; Duschek, S.; Kannenberg, W.; Gärtner, C.: Identifikation von Kompetenzbedarfen in Spitzen-technologien. In: Bullinger-Hoffmann, A. C. (Hrsg.): Kompetenzmanagement in Organisationen. Berlin: Springer-Verlag, 2019, S. 27-44
- [471] Scheller, T.: Auf dem Weg zur agilen Organisation. Wie Sie Ihr Unternehmen dynamischer, flexibler und leistungsfähiger gestalten. München: Vahlen-Verlag, 2017
- [472] Scherer, H.: Modellbasierte Methoden zur Modellierung des Zielsystems und des Funktions-Gestalt-Zusammenhangs zur Unterstützung der Serienentwicklung von Baukästen am Beispiel von Hybrid-Triebstrangsystemen. Karlsruhe: Institut für Produktentwicklung (IPEK), Karlsruher Institut für Technologie (KIT), 2016

- [473] Schmelzer, H. J.; Sesselmann, W.: Geschäftsprozess-management in der Praxis. Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen. München: Carl Hanser Verlag, 2013
- [474] Schmidt, T. S.; Behrenbeck, J.; Burger, K.; Hostettler, R.; Paetzold, K.; Zimmermann, M.: Mapping the progress in agile product development: a multi-case study. In: Wartzack, S.; Schleich, B.; Gon (Hrsg.): 22nd International Conference on Engineering Design (ICED 2019). Cambridge: Cambridge University Press, 2019, S. 1215-1224
- [475] Schmied, C.; Reinbold, G.; Amekrane, R.; Igenbergs, E.; Mörtl, M.; Lindemann, U.: Extended Cost Analysis with Systems Engineering Considerations. In: Schulze, S.-O.; Muggeo, C. (Hrsg.): Tag des Systems Engineering (TdSE 2015). München: Carl Hanser-Verlag, 2015, S. 227-236
- [476] Schmied, C.; Gebhardt, M.; d'Albert, H.; Mörtl, M.: Erweiterung der Design for Cost Methodik für indirekte Änderungskosten. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 27. DfX-Symposium (DfX 2016). Hamburg: TuTech Verlag, 2016, S. 77-88
- [477] Schmied, C.; Gebhardt, M.; Mörtl, M.; Lindemann, U.: Expert based approach to analyse and influence indirect cost of engineering changes. In: Maier, A.; Škec, S.; Kim, H.; Kokkolaras, M.; Oehmen, J.; Fadel, G.; Salustri, F.; Van der Loos, M. (Hrsg.): 21st International Conference on Engineering Design (ICED 2017), Glasgow: Design Society, 2017, S. 663-672
- [478] Schoeneberg, K.-P.: Komplexität – Eine Einführung in die Komplexitätsforschung und Auswirkungen auf das Management komplexer Projekte. In: Looock, H.; Steppeler, H.: (Hrsg.): Marktorientierte Problemlösungen im Innovationsmarketing. Festschrift für Prof. Dr. Michael P. Zerres. Wiesbaden: Gabler Springer Verlag, 2010, S. 479-500
- [479] Schoeneberg, K.-P. (Hrsg.): Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Wiesbaden: Springer Gabler, 2014
- [480] Schoeneberg, K.-P.: Komplexität: Einführung in die Komplexitätsforschung und Herausforderungen für die Praxis. In: Schoeneberg, K.-P. (Hrsg.): Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Wiesbaden: Springer Gabler, 2014, S. 13-27

- [481] Schoeneberg, K.-P.; Pein, J.: Entscheidungsfindung mit Big Data – Einsatz fortschrittlicher Visualisierungsmöglichkeiten zur Komplexitätsbeherrschung betriebswirtschaftlicher Sachverhalte im Unternehmen. In: Schoeneberg, K.-P. (Hrsg.): Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Wiesbaden: Springer Gabler, 2014, S. 309-354
- [482] Scholes, R.: Semiotics and interpretation. New Haven: Yale University Press, 1982
- [483] Schreiber, G.: Knowledge engineering and management: the CommonKADS methodology. Cambridge: The MIT Press, 2000
- [484] Schreiber, D.: Management von Controllingwissen. Ein sach- und verhaltensorientierter Ansatz zur Verbesserung der Manager-Controller-Beziehung. Wiesbaden: Gabler-Verlag, 2010
- [485] Schrepfer, M.: BPMN einführen: Erfolgsfaktoren für die konsequente Prozessmodellierung. Wirtschaftsinformatik und Management, 2(2010), Nr. 85, S. 70-77
- [486] Schreyögg, G.; Geiger, D.: Organisation. Grundlagen moderner Organisationsgestaltung. Mit Fallstudien. Wiesbaden: Springer Gabler Verlag, 2016
- [487] Schröder, A. (Hrsg.): Agile Produktentwicklung. Schneller zur Innovation – erfolgreicher am Markt. München: Carl Hanser Verlag, 2017
- [488] Schuermann, T.: Vier Programme zur Graph-Visualisierung, 2013, URL: <https://www.linux-magazin.de/ausgaben/2013/07/bitparade> (Abrufdatum: 06.09.2020)
- [489] Schuh, G.: Strategisches Produktionsmanagement – Expansion durch Konzentration. In: NC-Gesellschaft (Hrsg.): NCG-Jahreskongress 1994: Umdenken – Wandeln – Bestehen: Führungs- und Fachkompetenz sichert Industrieproduktion. München, 1994
- [490] Schuh, G.: Produktionsmanagement heute. Expansion durch Konzentration. In: Thommen, J.-P. (Hrsg.): Management-Kompetenz. Die Gestaltungsansätze des NDU/Executive MBA der Hochschule St. Gallen. Zürich: Versus Verlag, 1995, S. 429-446
- [491] Schuh, G.: Produktkomplexität managen. Strategien – Methoden – Tools. München: Carl Hanser Verlag, 2005

- [492] Schuh, G.: Lean Innovation. Berlin: Springer-Verlag, 2013
- [493] Schuh, G.; Riesener, M.: Produktkomplexität managen. Strategien – Methoden – Tools. München: Carl Hanser Verlag, 2017
- [494] Schuh, G.; Schwenk, U.; Speth, C.: Komplexitätsmanagement im St. Galler Management-Konzept. IO-Management. 67(1998), Nr. 3, S. 78-85
- [495] Schuh, G.; Krumm, S.; Amann, W.: Chefsache Komplexität. Navigation für Führungskräfte. Wiesbaden: Springer Gabler Verlag, 2013
- [496] Schuh, G.; Krumm, S.; Amann, W.; Rudolf, S.; Schiffer, M.: Chefsache Komplexität – die 6 zentralen Handlungsfelder des Komplexitätsmanagements. In: Gleich, R. (Hrsg.): Komplexitätscontrolling. Komplexität verstehen, reduzieren und beherrschen. Freiburg: Haufe Lexware Verlag, 2013, S. 27-38
- [497] Schuh, G.; Riesener, M.; Breunig, S.: Design for Changeability: Incorporating Change Propagation Analysis in Modular Product Platform Design. In: Takata, S.; Umeda, Y.; Kondoh, S. (Hrsg.): 24th CIRP Conference on Life Cycle Engineering. Procedia CIRP, Vol. 61. Philadelphia: Elsevier Verlag, 2017, S. 63-68
- [498] Schulz, A. P.: Systemtechnische Gestaltung der Informationsarchitektur im Entwicklungsprozess. München: Herbert Utz Verlag, 2003
- [499] Schumann, H.; Müller, W.: Visualisierung. Grundlagen und allgemeine Methoden. Berlin: Springer-Verlag, 2000
- [500] Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations- und anpassungsorientierter Modelle. Wiesbaden: Springer Gabler Verlag, 1998
- [501] Schwarz, S.: Wissens- und Informationsmanagement – Change-Management-Perspektiven. In: Keuper, F.; Neumann, F. (Hrsg.): Wissens- und Informationsmanagement. Wiesbaden: Gabler Verlag, 2009, S. 119-145
- [502] Seidenschwarz, W.: Target Costing. In: Diller H.; Herrmann A. (Hrsg.): Handbuch Preispolitik. Wiesbaden: Gabler Verlag, 2003, S. 437-453
- [503] Seidenschwarz & Company GmbH: Target Costing – Auf dem Weg zum marktorientierten Unternehmen. Starnberg: Seidenschwarz & Company GmbH, 2005

- [504] Senge, P. M.: Die fünfte Disziplin: Kunst und Praxis der lernenden Organisation. Stuttgart: Schäffer-Poeschel Verlag, 2017
- [505] Shankar, P.; Morkos, B.; Summers, J. D.: Reasons for change propagation: a case study in an automotive OEM. *Research in Engineering Design*, (23)2012, Nr. 4, S. 291-303
- [506] Shapiro, D.; Clarkson, J.: A concept for modelling and analysing design process changes. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N.; Skec, S. (Hrsg.): 14th International Design Conference (DESIGN 2016). Glasgow: Design Society, 2016, S. 1573-1584
- [507] Sharman, D. M.; Yassine, A. A.: Characterizing complex product architectures. *Systems Engineering*, (7)2003, Nr. 1, S. 35-60
- [508] Sharon, A.; de Weck, O. L.; Dori, D.: Improving Project-Product Lifecycle Management with Model-Based Design Structure Matrix: A joint project management and systems engineering approach. *Systems Engineering*, (16)2013, Nr. 4, S. 413-426
- [509] Simon, H. A.: The Architecture of Complexity. *Proceedings of the American Philosophical Society*. 106(1962), Nr. 6, S. 467-482
- [510] Simon, C.; Hientzsch, B.: Prozesseigner. Wissen & Methoden für Manager von Unternehmensprozessen. Wiesbaden: Springer Vieweg, 2014
- [511] Sinz, E. J.: Konstruktionsforschung in der Wirtschaftsinformatik: Was sind die Erkenntnisziele gestaltungsorientierter Wirtschaftsinformatik-Forschung? In: Österle, H.; Winter, R.; Brenner, W. (Hrsg.): Gestaltungsorientierte Wirtschaftsinformatik. Ein Plädoyer für Rigor und Relevanz. Radebeul: infowerk, 2010, S. 27-35
- [512] Skirde, H.: Kostenorientierte Bewertung modularer Produktarchitekturen. Köln: Josef Eul Verlag, 2015
- [513] Smith, R.; Morrow, J.: Product development process modeling. *Design Studies*, (20)1999, Nr. 3, S. 237-261
- [514] Snowden, D. J.; Boone, M. E.: A Leader's Framework for Decision Making. *Harvard Business Review*, 85(2007) o. Nr., S. 68-76.
- [515] Soley GmbH: Soley, 2020, URL: <https://soley.io> (Abrufdatum: 20.07.2020)

- [516] Sowa, J.: Knowledge representation: logical, philosophical, and computational foundations. Pacific Grove: Brooks/Cole Verlag, 2000
- [517] Spath, D.: Grundlagen der Organisationsgestaltung. In: Spath, D.; Westkämper, E. (Hrsg.): Handbuch Unternehmensorganisation. Strategien, Planung, Umsetzung. Berlin: Springer-Verlag, 2017, S. 1-27
- [518] Spönemann, M.: Graph Layout Support for Model-Driven Engineering. Norderstedt: BoD – Books on Demand, 2015
- [519] Spur, G.; Krause, F.-L.: Das virtuelle Produkt: Management der CAD-Technik. München: Carl Hanser Verlag, 1997
- [520] Staab, S.: Wissensmanagement mit Ontologien und Metadaten. Informatik Spektrum, (25)2002, Nr. 3, S. 194-209
- [521] Staab, S.; Studer, R. (Hrsg.): Handbook on Ontologies. Berlin: Springer-Verlag, 2009
- [522] Staab, S.; Studer, R.; Schnurr, H.-P.; Sure, Y.: Knowledge processes and ontologies. IEEE Intelligent Systems, (16)2001, Nr. 1, S. 26-34
- [523] Staab, S.; Studer, R.; Sure, Y.: Knowledge Processes and Meta Processes in Ontology-Based Knowledge Management. In: Holsapple, C. (Hrsg.): Handbook on Knowledge Management. Berlin: Springer-Verlag, 2003, S. 47-67
- [524] Stachowiak, H.: Allgemeine Modelltheorie. Wien: Springer-Verlag, 1973
- [525] Stachowiak, H. (Hrsg.): Modelle – Konstruktion der Wirklichkeit. München: Wilhelm Fink Verlag, 1983
- [526] Stachowiak, H.: Modell. In: Seiffert, H.; Radnitzky, G. (Hrsg.): Handlexikon zur Wissenschaftstheorie. München: Ehrenwirth-Verlag, 1989, S. 220-222
- [527] Stechert, C.: Modellierung komplexer Anforderungen. Braunschweig: Institut für Konstruktionstechnik, Technische Universität Carolo-Wilhelmina zu Braunschweig, 2010

- [528] Stechert, C.; Franke, H.-J.: Modellierung von komplexen Anforderungen. In: Meerkamm, H. (Hrsg.): 19th Symposium Design for X (DfX 2008). Neukirchen/Erlangen: Lehrstuhl für Konstruktionstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2008, S. 139-148
- [529] Stelter, P.: Modulare mechatronische Produktentwicklung im Maschinen- und Anlagenbau mit Anwendungen zur Smart Factory. In: ten Hompel, M.; Vogel-Heuser, B.; Bauernhansl, T. (Hrsg.): Handbuch Industrie 4.0. Produktion, Automatisierung und Logistik. Springer Reference Technik. Berlin: Springer Vieweg, 2020, S. 1-40
- [530] Steward, D. V.: On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations. SIAM Review, (4)1962, Nr. 4, S. 321-342
- [531] Steward, D. V.: The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE Transactions on Engineering Management, (28)1981, Nr. 3, 1981, S. 71-74
- [532] Storga, M.; Andreasen, M.; Marjanović, D.: The design ontology: foundation for the design knowledge exchange and management. Journal of Engineering Design, (21)2010, Nr. 4, S. 427-454
- [533] Strahringer, S.: Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Aachen: Shaker-Verlag, 1996
- [534] Strahringer, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In: Pohl, K.; Schürr, A. & Vossen, G. (Hrsg.): GI-Workshop Modellierung. Aachen: CEUR Workshop Proceedings, Vol. 9, 1998, S. 15-20
- [535] Strahringer, S.: Probleme und Gefahren im Umgang mit „Meta“-Begriffen: ein Plädoyer für eine sorgfältige Begriffsbildung. In: International Knowledge Technology Forum (KnowTechForum 1999). Potsdam, 1999
- [536] Stuckenschmidt, H.: Ontologien. Konzepte, Technologien und Anwendungen. Berlin: Springer-Verlag, 2011
- [537] Sugiyama, K.: Graph Drawing and Applications for Software and Knowledge Engineers. New Jersey: World Scientific Verlag, 2002

- [538] Sugiyama, K.; Tagawa, S.; Toda, M.: Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Management, and Cybernetics*, (11)1981, Nr. 2, S. 109-125
- [539] Suh, N. P.: Design of systems. *CIRP Annals-Manufacturing Technology*, (46)1997, Nr. 1, S. 75-80
- [540] Suh, N. P.: Axiomatic design theory for systems. *Research in Engineering Design*, (10)1998, Nr. 4, S. 189-209
- [541] Suh, N. P.: *Axiomatic Design. Advances and Applications*. New York: Oxford University Press, 2001
- [542] Suh, E. S.; de Weck, O. L.; Chang, D.: Flexible product platforms: framework and case study. *Research in Engineering Design*, (18)2007, Nr. 2, S. 67-89
- [543] Sure, Y.; Akkermans, H.; Broekstra, J.; Davies, J.; Ding, Y.; Duke, A.; Engels, R.; Fensel, D.; Horrocks, I.; Iosif, V.; Kampman, A.; Kiryakov, A.; Klein, M.; Lau, T.; Ognyanov, D.; Reimer, U.; Simov, K.; Studer, R.; van der Meer, J.; van Harmelen, F.: On-To-Knowledge: Semantic Web-Enabled Knowledge Management. In: Zhong, N.; Liu, J.; Yao, Y. (Hrsg.): *Web Intelligence*. Berlin: Springer-Verlag, 2003, S. 277-300
- [544] Syldatke, T.; Lutz, M.; Chen, W.; Hess, C.: Managing Dependencies in the Product Development Process with Semantic Technologies. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (Hrsg.): *10th International DSM Conference (DSM 2008)*. München: Carl Hanser-Verlag, 2008, S. 105-114
- [545] Tamassia, R.: *Handbook of graph drawing and visualization*. London: Chapman and Hall Verlag, 2014
- [546] Tamassia, R.; Battista, G. D.; Batini, C.: Automatic Graph Drawing and Readability of Diagrams. *IEEE Transactions on Systems, Management, and Cybernetics*, (18)1988, Nr. 1, S. 61-79
- [547] Tang, D.; Zhu, R.; Tang, J.; Xu, R.; He, R.: Product design knowledge management based on design structure matrix. *Advanced Engineering Informatics*, (24)2010, Nr. 2, S. 159-166
- [548] Thebeau, R. E.: *Knowledge Management of System Interfaces and Interactions for Product Development Processes*. Cambridge: Massachusetts Institute of Technology (MIT), 2001

- [549] Thel, M.: Wissensstrukturierung und -repräsentation im Produktentwicklungsprozess. Aachen: Shaker-Verlag, 2007
- [550] Thiebes, F.; Plankert, N.: Umgang mit Komplexität in der Produktentwicklung. Komplexitätsbeherrschung durch Variantenmanagement. In: Schoeneberg, K.-P. (Hrsg.): Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Wiesbaden: Springer Gabler Verlag, 2014, S. 165-185
- [551] Thiele, H.: Zur Definition von Kompliziertheitsmaßnahmen für endliche Objekte. In: Klix, F. (Hrsg.): Organismische Informationsverarbeitung – Zeichenerklärung, Begriffsbildung, Problemlösen, Berlin: Akademie-Verlag, 1974, S. 147-164
- [552] Thomas, O.; Michael Fellmann, M.: Semantische Prozessmodellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen. Wirtschaftsinformatik, (51)2009, Nr. 6, S. 506-518
- [553] Tilstra, Andrew H.; Campbell, Matthew I.; Wood, Kristin L.; Seepersad, Carolyn C.: Comparing matrix-based and graph-based representations for product design. In: Wynn, D. C.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, J. (Hrsg.): 12th International Design Structure Matrix Conference (DSM 2010). München: Carl Hanser Verlag, 2010, S. 227-236
- [554] Tilstra, A. H.; Seepersad, C. C.; Wood, K. L.: A high-definition design structure matrix (HDDSM) for the quantitative assessment of product architecture. Journal of Engineering Design, (23)2012, Nr. 10-11, S. 767-789
- [555] Tittmann, P.: Graphentheorie: Eine anwendungsorientierte Einführung. München: Carl Hanser Verlag, 2019
- [556] Toepfer, F.; Naumann, T.: Towards cross-linked development of highly complex products. In: Maier, A.; Škec, S.; Kim, H.; Kokkolaras, M.; Oehmen, J.; Fadel, G.; Salustri, F.; Van der Loos, M. (Hrsg.): 21st International Conference on Engineering Design (ICED 2017), Glasgow: Design Society, 2017, S. 279-288
- [557] Tomiyama, T.; Gu, P.; Jin, Y.; Lutters, D.; Kind, C.; Kimura, F.: Design methodologies: Industrial and educational applications. CIRP Annals, (25)2009, Nr. 2, S. 543-565

- [558] Träger, T.: Organisation. Grundlagen der Organisationslehre mit Beispielen, Übungsaufgaben und Musterlösungen. München: Verlag Franz Vahlen, 2018
- [559] Trier, M.; Bobrik, A.; Neumann, N.; Wyssussek, B.: Systemtheorie und Modell. In: Krallmann, H.; Bobrik, A.; Levina, O. (Hrsg.): Systemanalyse im Unternehmen. Prozessorientierte Methodender Wirtschaftsinformatik. München: Oldenbourg Wissenschaftsverlag, 2013, S. 41-72
- [560] Technische Universität München: The Design Structure Matrix (DSM), 2020, URL: <https://dsmweb.org> (Abrufdatum: 20.07.2020)
- [561] Turau, V.: Algorithmische Graphentheorie. München: Oldenbourg Verlag, 2009
- [562] Turau, V.; Weyer, C.: Algorithmische Graphentheorie. Berlin: De Gruyter Verlag, 2015
- [563] Turner, J. U.; Subramaniam, S.; Gupta, S.: Constraint representation and reduction in assembly modeling and analysis. IEEE Transactions on Robotics and Automation, (8)1992, Nr. 6, S. 741-750
- [564] Tutte, W. T.: How to Draw a Graph. Proceedings of the London Mathematical Society, (13)1963, Nr. 1, S. 743-767
- [565] Ulrich, H.; Probst, G. J. B.: Anleitung zum ganzheitlichen Denken und Handeln. Ein Brevier für Führungskräfte. Bern: Paul Haupt Verlag, 1991
- [566] Uschold, M.; King, M.: Towards a Methodology for Building Ontologies. In: Skuce, D. (Hrsg.): Workshop on Basic Ontological Issues in Knowledge Sharing, in conjunction with 14th International Joint Conference on Artificial Intelligence (IJCAI-95). San Francisco: Morgan Kaufmann Publishers, 1995
- [567] Vahs, D.: Organisation. Ein Lehr- und Managementbuch. Stuttgart: Schäffer-Poeschel Verlag, 2019
- [568] Vajna, S.; Weber, C.; Zeman, K.; Hehenberger, P.; Gerhard, D.; Wartzack, S. (Hrsg.): CAx für Ingenieure. Eine praxisbezogene Einführung. Berlin: Springer-Verlag, 2018
- [569] VDI-Richtlinie 2235: Wirtschaftliche Entscheidungen beim Konstruieren – Methoden und Hilfen. Düsseldorf: VDI-Verlag, 1987

- [570] VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Berlin: Beuth Verlag, 1993
- [571] VDI-Richtlinie 2225 – Blatt 1: Konstruktionsmethodik. Technisch-wirtschaftliches Konstruieren. Vereinfachte Kostenermittlung. Berlin: Beuth Verlag, 1997
- [572] VDI-Richtlinie 2225 Blatt – 4: Konstruktionsmethodik. Technisch-wirtschaftliches Konstruieren. Bemessungslehre. Berlin: Beuth Verlag, 1997
- [573] VDI-Richtlinie 2225 – Blatt 2: Konstruktionsmethodik. Technisch-wirtschaftliches Konstruieren. Tabellenwerk. Berlin: Beuth Verlag, 1998
- [574] VDI-Richtlinie 2225 – Blatt 3: Konstruktionsmethodik. Technisch-wirtschaftliches Konstruieren. Technisch-wirtschaftliche Bewertung. Berlin: Beuth Verlag, 1998
- [575] VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme. Berlin: Beuth Verlag, 2004
- [576] VDI-Richtlinie 2209: 3-D-Produktmodellierung Technische und organisatorische Voraussetzungen Verfahren, Werkzeuge und Anwendungen Wirtschaftlicher Einsatz in der Praxis. Berlin: Beuth Verlag, 2009
- [577] VDI-Richtlinie 5610 – Blatt 1: Wissensmanagement im Ingenieurwesen. Grundlagen, Konzepte, Vorgehen. Berlin: Beuth Verlag, 2009
- [578] VDI-Richtlinie 3633 – Blatt 11: Simulation von Logistik-, Materialfluss- und Produktionssystemen. Simulation und Visualisierung. Berlin: Beuth Verlag, 2009
- [579] VDI-Richtlinie 5610 – Blatt 2: Wissensmanagement im Ingenieurwesen Wissensbasierte Konstruktion (KBE). Berlin: Beuth Verlag, 2009
- [580] VDI-Richtlinie 2221 – Blatt 1: Entwicklung technischer Produkte und Systeme. Modell der Produktentwicklung. Berlin: Beuth Verlag, 2019
- [581] VDI-Richtlinie 2221 – Blatt 2: Entwicklung technischer Produkte und Systeme. Gestaltung individueller Produktentwicklungsprozesse. Berlin: Beuth Verlag, 2019

- [582] VDMA: Kennzahlenkompass: Benchmarks und Informationen des Maschinen- und Anlagenbaus. Frankfurt am Main: VDMA Verlag, 2020
- [583] Vogel, H.: Einstieg in CAD. München: Carl Hanser Verlag, 2004
- [584] Voigt, S.; Seidel, H.: Herausforderung für Unternehmen. In: Mertins, K.; Seidel, H. (Hrsg.): Wissensmanagement im Mittelstand. Berlin: Springer Verlag, 2009, S. 9-13
- [585] Wahdan, H. G.; Kassem, S. S.; Abdelsalam, H. M.: Product Modularization Using Cuckoo Search Algorithm. In: Vitoriano, B.; Parlier, G. (Hrsg.): 5th International Conference on Operations Research and Enterprise Systems (ICORES 2016). Operations Research and Enterprise Systems. Cham: Springer-Verlag, 2017, S. 20-34
- [586] Walter, T.; Parreiras, F.; Staab, S.: An ontology-based framework for domain-specific modeling. *Software & Systems Modeling*, (13)2014, Nr. 1, S. 83-108
- [587] Walther, C.: Systemtechnisches Verfahren zur Bestimmung der Zusammenhänge zwischen Eigenschaften und Funktionsstruktur technischer Systeme. München: Lehrstuhl für Raumfahrttechnik, Technische Universität München (TUM), 1994
- [588] Wartzack, S.: Predictive Engineering: Assistenzsystem zur multikriteriellen Analyse alternativer Produktkonzepte. Düsseldorf: VDI-Verlag, 2001
- [589] Wartzack, S.: Auswahl- und Bewertungsmethoden. In: Feldhusen, J.; Grote, K.-H. (Hrsg.): Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin: Springer-Verlag, 2013, S. 380-409
- [590] Weber, C.: CPM/PDD – an extended theoretical approach to modelling products and product development processes. In: Bley, H.; Jansen, H.; Krause, F.-L.; Shpitalni, M. (Hrsg.): 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes. Stuttgart: Fraunhofer-IRB-Verlag, 2005, S. 159-179
- [591] Weber, C.: What is “complexity”? In: Samuel, A.; Lewis, W. (Hrsg.): 15th International Conference on Engineering Design (ICED 2005). Glasgow: Design Society, 2005, S. 292-293

- [592] Weber, C.: Produkte und Produktentwicklungsprozesse abbilden mit Hilfe von Merkmalen und Eigenschaften – eine kritische Zwischen-bilanz. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 23. Symposium Design For X (DfX 2012). Hamburg: Tu-Tech-Verlag, 2012, S. 25-62
- [593] Weber, C.; Werner, H.: Klassifizierung von CAx-Werkzeugen für die Produktentwicklung auf der Basis eines neuartigen Produkt- und Prozessmodells. In: Meerkamm, H. (Hrsg.): 11th Symposium Design for X (DfX 2000). Schnaittach/Erlangen: Lehrstuhl für Konstruktionstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2000, S. 126-143
- [594] Weber, C.; Werner, H.; Deubel, T.: A different view on PDM and its future potentials. In: Marjanovic D. (Hrsg.): 7th International Design Conference (DESIGN 2002), Glasgow: Design Society, 2002, S. 101-112
- [595] Weber, J.; Förster, D.; Kößler, J.; Paetzold, K.: Design of changeable production units within the automotive sector with axiomatic design. In: Thompson, M. K.; Giorgetti, A.; Citti, P.; Matt, D.; Suh, N. P. (Hrsg.): 9th International Conference on Axiomatic Design (ICAD 2015), Procedia CIRP, Vol. 34. Philadelphia: Elsevier Verlag, 2015, S. 93-97
- [596] Weilkiens, T.: Systems Engineering with SysML/ UML: Modeling, Analysis, Design. San Francisco: Morgan Kaufmann OMG Press, 2008
- [597] Weilkiens, T.: SYSMOD - The Systems Modeling Toolbox. Pragmatic MBSE with SysML. Fredesdorf: MBSE4U – Tim Weilkiens, 2016
- [598] Weiser, A.-K.: Methodik eines holistischen Variantenmanagements modularer Produktfamilien - Grundlagen, Systematik und beispielhafte Anwendung der VM_{ahead} Methodik. Karlsruhe: KIT Scientific Publishing, 2018
- [599] Weisshaupt, B.: Interdisziplinärer prozessorientierter Entwurf intelligenter technischer Systeme. München: Herbert Utz Verlag, 1998
- [600] Wenger, E.; Snyder, W. M.: Communities of Practice: The Organizational Frontier. Boston: Harvard Business School Press, 2020

- [601] Wenger, E.; McDermott, R.; Snyder, W. M.: *Cultivating Communities of Practice: A Guide to Managing Knowledge – Seven Principles for Cultivating Communities of Practice*. Boston: Harvard Business School Press, 2002
- [602] Wenzel, S.: *Organisation und Methodenauswahl in der Produktentwicklung*. München: Herbert Utz Verlag, 2003
- [603] Westkämper, E.; Hanselmann, J.: *Effektive Wissensnutzung in der Produktentwicklung*. *Industrie Management*. 16(2000), Nr. 1, S. 28-32
- [604] Westphal, C.: *Ein Beitrag zur semantischen Modellierung und Analyse von Informationsflüssen in der Produkteigenschaftsabsicherung*. Düsseldorf: VDI-Verlag, 2014
- [605] Weuster, A.: *Unternehmensorganisation. Organisationsprojekte – Aufbaustrukturen*. München: Rainer Hampp Verlag, 2008
- [606] Wied, M.; Oehmen, J.; Welo, T.: *Conceptualizing resilience in engineering systems: An analysis of the literature*. *Systems Engineering*. Wiley, 23(2019), Nr. 1, S. 3-13
- [607] Wiener, N.: *Cybernetics or control and communication in the animal and the machine*. Cambridge: The M.I.T. Press, 1948
- [608] Wissenschaftliche Gesellschaft für Produktentwicklung (WiGeP): *Herausforderungen und Ziele*, 2020, URL: <http://www.wigep.de/index.php?id=78&L=38> (Abrufdatum: 26.07.2020)
- [609] Wilde, T.; Hess, T.: *Methodenspektrum der Wirtschaftsinformatik. Eine empirische Untersuchung*. *Wirtschaftsinformatik*, (49)2007, Nr. 4, S. 280-287
- [610] Wildemann, H.: *Komplexität: Vermeiden oder beherrschen lernen – Für ein überlegtes Komplexitätsmanagement gibt es genug wirksame Mittel und Wege*. *Harvard Business Manager* 21(1999), Nr. 6, S. 31-45
- [611] Wildemann, H.: *Komplexitätsmanagement – in Vertrieb, Beschaffung, Produkt, Entwicklung und Produktion*. München: TCW-Verlag, 2011
- [612] Wildemann, H.: *Standardisierung und Individualisierung des Produktprogramms durch das Controlling*. *Controlling - Zeitschrift für erfolgsorientierte Unternehmenssteuerung*. 23(2011), Nr. 4-5, S. 212-216

- [613] Wildemann, H.: Änderungsmanagement: Leitfaden zur Einführung eines effizienten Managements technischer Änderungen. München: TCW Transfer-Centrum, 2015
- [614] Wildemann, H.; Voigt, K.-I.: Komplexitätsindex-Tool: Entscheidungsgrundlagen für die Produktprogrammgestaltung bei KMU. München und Nürnberg: Forschungsinstitut für Unternehmensführung, Logistik und Produktion, Technische Universität München (TUM); Lehrstuhl für Industriebetriebslehre, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2011
- [615] Wildemann, H.; Ann, C.; Broy, M.; Günthner, W. A.; Lindemann, U.: Plagiatschutz. Handlungsspielräume der produzierenden Industrie gegen Produktpiraterie. München: TCW-Verlag, 2007
- [616] Wilking, F.; Schleich, B.; Wartzack, S.: MBSE along the Value Chain – An Approach for the Compensation of additional Effort. In: IEEE (Hrsg.): 15th International Conference of System of Systems Engineering (SoSE). Budapest: IEEE, 2020, S. 61-66
- [617] Wittel, H.; Jannasch, D.; Voßiek, J.; Spura, C.: Roloff/Matek Maschinenelemente. Normung, Berechnung, Gestaltung. Wiesbaden: Springer-Verlag, 2019
- [618] Woll, R.; Hayka, H.; Stark, R.: Ontologiebasierte Datenintegration für das Modellbasierte Systems Engineering. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering (TdSE 2014). München: Hanser-Verlag, 2014, S. 33-42
- [619] Würtenberger, J.: Ein Beitrag zur Identifikation und Beherrschung von Unsicherheit bei der Modellierung technischer Systeme. Darmstadt: Fachgebiet Produktentwicklung und Maschinenelemente, Technischen Universität Darmstadt (TUD), 2018
- [620] Wymore, A.: Model-Based Systems Engineering: an introduction to the mathematical theory of discrete systems and to the tricotyledon theory of system design. Boca Raton: Taylor & Francis Inc, 1993
- [621] Wynn, D. C.: Model-Based Approaches to Support Process Improvement in Complex Product Development. Cambridge: Homerton College, University of Cambridge, 2007

- [622] Wynn, D. C.; Nair, S. MT.; Clarkson, P. J.: The P₃ Platform: An approach and software system for developing diagrammatic model-based methods in design research. In: Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann, U. (Hrsg.): 17th International Conference on Engineering Design (ICED 2009). Glasgow: Design Society, 2009, S. 559-570
- [623] Wynn, D. C.; Wyatt, D. F.; Nair, S. M.; Clarkson, P. J.: An Introduction to the Cambridge Advanced Modeller. In: Heisig, P.; Clarkson, P. J.; Vajna, S. (Hrsg.): 1st International Conference on Modelling and Management of Engineering Processes (MMEP 2010). Modelling and Management of Engineering Processes. London: Springer-Verlag, 2010, S. 89-100
- [624] Xie, S. Q.; Yang, W. Z.; Tu, Y. L.: Towards a generic product modelling framework. International Journal of Production Research, (46)2008, Nr. 8, S. 2229-2254
- [625] Yang, Q.; Yao, T.; Lu, T.; Zhang, B.: An Overlapping-Based Design Structure Matrix for Measuring Interaction Strength and Clustering Analysis in Product Development Project. IEEE Transactions on Engineering Management, Institute of Electrical & Electronics Engineers (IEEE), 2014, 61, 159-170
- [626] Yassine, A.: An introduction to modeling and analyzing complex product development processes using the design structure matrix (DSM) method. Quaderni di Manage (Italian Management Review), 2004, Nr. 9, S. 71-88
- [627] Yu, T.-L.; Yassine, A. A.; Goldberg, D. E.: A Genetic Algorithm for Developing Modular Product Architectures. Volume 3b: 15th International Conference on Design Theory and Methodology, New York: ASME, 2003, 515-524
- [628] Yu, T.-L.; Yassine, A. A.; Goldberg, D. E.: An information theoretic method for developing modular architectures using genetic algorithms. Research in Engineering Design, Springer Science and Business Media LLC, 2007, 18, 91-109
- [629] Zakarian, A.: A New Nonbinary Matrix Clustering Algorithm for Development of System Architectures. IEEE Transactions on Systems, Management, and Cybernetics, (38)2008, Nr. 1, S. 135-141

- [630] Zingel, J. C.: Basisdefinition einer gemeinsamen Sprache der Produktentwicklung im Kontext der Modellbildung technischer Systeme und einer Modellierungstechnik für Zielsystem und Objektsystem technischer Systeme in SysML auf Grundlage des Zho-Prinzips. Karlsruhe: Institut für Produktentwicklung, Karlsruher Institut für Technologie (KIT), 2013
- [631] Zirkler, S. C.: Transdisziplinäres Zielkostenmanagement komplexer mechatronischer Produkte. München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2010

Verzeichnis promotionsbezogener, eigener Publikationen

- [P1] Bausewein, N.; Luft, T.; Thater, G.; Wartzack, S.: A Holistic Decision Making Process for Awarding Development Projects to External Service Providers. In: Brökel, K.; Feldhusen, J.; Grote, K.; Frank, R.; Stelzer, R. (Hrsg.): 11. Gemeinsames Kolloquium Konstruktions-technik (KT 2013). Aachen: Shaker, 2013, S. 73-82
- [P2] Crostack, A.; Luft, T.; Binz, H.; Wartzack, S.; Roth, D.: Konzept eines erweiterten Produktbeschreibungsansatzes basierend auf dem Vergleich des Characteristics-Properties-Modelling und der matrixbasierten Produktbeschreibung. In: Binz, H.; Bertsche, B.; Bauer, W.; Roth, D. (Hrsg.): 3. Stuttgarter Symposium für Produktentwicklung (SSP 2015). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), 2015, o. S.
- [P3] Fechter, M.; Kett, S.; Luft, T.; Wartzack, S.: Impact of collaborative space on team work in engineering design education. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N.; Skec, S. (Hrsg.): 14th International Design Conference (DESIGN 2016). Glasgow: Design Society, 2016, S. 975-984
- [P4] Kamin, S.; Luft, T.; Miehl, J.; Williger, B.; Lang, F. R.; Wartzack, S.: Subjektive Adaptionsfähigkeit im Kontext der alternssensiblen Produktentwicklung. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 25. DfX-Symposium (DfX 2014). Hamburg: TuTech, 2014, S. 99-110
- [P5] Kestel, P.; Luft, T.; Schon, C.; Kügler, P.; Bayer, T.; Schleich, B.; Staab, S.; Wartzack, S.: Konzept zur zielgerichteten, ontologie-basierten Wiederverwendung von Produktmodellen. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 28. DfX-Symposium (DfX 2017). Hamburg: TuTech, 2017, S. 241-252
- [P6] Luft, T.; Westphal, C.; Wartzack, S.: *Anforderungsanalyse zur kontextbezogenen Verwaltung und Bereitstellung von Prozess- und Produktwissen*. Diplomarbeit (2011), Erlangen

- [P7] Luft, T.; Breitsprecher, T.; Roth, D.; Lindow, K.; Binz, H.; Wartzack, S.: Die Rolle des Wissensingenieurs im Unternehmen – Ergebnisse einer Umfrage und Darstellung in der VDI-Richtlinie „Wissensbasiertes Konstruieren“. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 23. DfX-Symposium (DfX 2012). Hamburg: TuTech, 2012, S. 63-78
- [P8] Luft, T.; Wartzack, S.: Requirement analysis for contextual management and supply of process- and design knowledge – a case study. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 12th International Design Conference (DESIGN 2012). Glasgow: Design Society, 2012, S. 1515-1524
- [P9] Luft, T.; Wartzack, S.: Retrieving knowledge and information by using a systematic search interface – an industrial case study. In: Kyvsgård Hansen, P.; Rasmussen, J.; Jørgensen, K. A.; Tollestrup, C. (Hrsg.): 9th Norddesign Conference (NordDesign 2012). Glasgow: Design Society, 2012, o. S.
- [P10] Luft, T.; Krehmer, H.; Wartzack, S.: An advanced procedure model for property-based product development. In: Lindemann, U.; Srinivasan, V.; Kim, Y. S.; Lee, S. W.; Clarkson, J.; Cascini, G. (Hrsg.): 19th International Conference on Engineering Design (ICED 2013). Glasgow: Design Society, 2013, S. 187-196
- [P11] Luft, T.; Smoll, S.; Wartzack, S.: An Indicator-Based Process Monitoring Cockpit for Controlling and Enhancing Product Development Processes – An Industrial Case Study. In: Abramovici, M.; Stark, R. (Hrsg.): Smart Product Engineering. 23rd CIRP Design Conference (CIRP Design 2013). Lecture Notes in Production Engineering. Berlin: Springer, 2013, S. 905-914
- [P12] Luft, T.; Kamin, S.; Miehl, J.; Lang, F. R.; Wartzack, S.: Die eigenschaftsbasierte Produktentwicklung unter Berücksichtigung der differenziellen Anforderungen älterer Nutzer. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): 24. DfX-Symposium (DfX 2013). Hamburg: TuTech, S. 197-208
- [P13] Luft, T.; Wartzack, S.: Die matrixbasierte Produktbeschreibung als Bestandteil des Vorgehensmodells in der eigenschaftsbasierten Produktentwicklung. In: Spath, D.; Binz, H.; Bertsche, B. (Hrsg.): 2. Stuttgarter Symposium für Produktentwicklung (SSP 2013). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), 2013, o. S. – *ausgezeichnet mit Best Paper Award*

- [P14] Luft, T.; Bochmann, J.; Wartzack, S.: Enhancing the flow of information in the PLM by using numerical DSMs – an industrial case study. In: Bernard, A.; Rivest, L.; Dutta, D. (Hrsg.): 10th IFIP International Conference on Product Lifecycle Management (PLM 2013). Product Lifecycle Management for Society. IFIP Advances in Information and Communication Technology, Vol 409. Berlin: Springer, S. 90-99 – *ausgezeichnet mit Best Paper Award*
- [P15] Luft, T.; Schleich, B.; Wartzack, S.: Concept development for innovative products – a challenge for engineering design education. In: Lindemann, U.; Srinivasan, V.; Kim, Y. S.; Lee, S. W.; Clarkson, J.; Cascini, G. (Hrsg.): 19th International Conference on Engineering Design (ICED 2013). Glasgow: Design Society, 2013, S. 105-114
- [P16] Luft, T.; Ewringmann, N.; Wartzack, S.: Application and validation of the matrix-based product description in a case study by using the software Loomeo. In: Moroni, G.; Tolio, T. (Hrsg.): 24th CIRP Design Conference (CIRP Design 2014). Procedia CIRP, Vol. 21. Philadelphia: Elsevier Verlag, 2014, S. 479-484
- [P17] Luft, T.; Klein, F.; Wartzack, S.: Weight and cost reduction of the thermal management for traction batteries by using the DSM methodology. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 13th International Design Conference (DESIGN 2014). Glasgow: Design Society, 2014, S. 293-302
- [P18] Luft, T.; Wartzack, S.: Klassifikation und Handhabung von Unsicherheiten zur entwicklungsbegleitenden Erfassung des Produktreifegrades. In: Stelzer, R. (Hrsg.): Entwerfen – Entwickeln – Erleben 2014: Beiträge zur virtuellen Produktentwicklung und Konstruktionstechnik (EEE 2014). Dresden: TUDpress, S. 535-549
- [P19] Luft, T.; Reitmeier, J.; Paetzold, K.; Wartzack, S.: Uncertainty considerations by context factors for the monitoring of the degree of product maturity. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (Hrsg.): 13th International Design Conference (DESIGN 2014). Glasgow: Design Society, 2014, S. 893-904
- [P20] Luft, T.; Wartzack, S.: Verschiedene Rollen im Wissensmanagement – eine bibliografische Studie. Wissensmanagement – Das Magazin für Führungskräfte 16(2014), Nr. 8, S. 24-27
- [P21] Luft, T.; Frey, D.; Wartzack, S.: Welches Wissensmanagement-System passt zum Unternehmen?. Wissensmanagement – Das Magazin für Führungskräfte 16(2014), Nr. 1, S. 48-51

- [P22] Luft, T.; Schneider, S.; Wartzack, S.: A methodical approach to model and map interconnected decision making situations and their consequences. In: Weber, C.; Husung, S.; Cascini, G.; Cantamessa, M.; Marjanovic, D.; Rotini, F. (Hrsg.): 20th International Conference on Engineering Design (ICED 2015). Glasgow: Design Society, 2015, S. 329-340
- [P23] Luft, T.; Wartzack, S.: Multi-criteria evaluation and decision making on the basis of the matrix-based modeling of the product development. In: Dumitrescu, R.; Lindemann, U. (Hrsg.): 3rd International Spring School on Systems Engineering (IS3E 2015). Paderborn. Fraunhofer, 145-154
- [P24] Luft, T.; Wartzack, S.: Was macht ein Wissensingenieur?. Wissensmanagement – Das Magazin für Führungskräfte 17(2015), Nr. 1, S. 8-11
- [P25] Luft, T.; Stal-Le Cardinal, J.; Wartzack, S.: Methoden der Entscheidungsfindung. In: Lindemann, U. (Hrsg.): Handbuch der Produktentwicklung, München: Carl Hanser Verlag, 2016, S. 759-803
- [P26] Luft, T.; Lamé, G.; Ponn, J.; Stal-Le Cardinal, J.; Wartzack, S.: A business model canvas for idecide - how to design a new decision making app. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N.; Skec, S. (Hrsg.): 14th International Design Conference (DESIGN 2016). Glasgow: Design Society, 2016, S. 1523-1532
- [P27] Luft, T.; Wartzack, S.: Application, evaluation and future research potential of the matrix and graph based product model. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N.; Skec, S. (Hrsg.): 14th International Design Conference (DESIGN 2016). Glasgow: Design Society, 2016, S. 351-360
- [P28] Luft, T.; Roth, D.; Wartzack, S.: Knowledge Based Engineering: Konstruktionsrelevantes Wissen erfassen & nutzbar machen. Wissensmanagement – Das Magazin für Führungskräfte 18(2016), Nr. 8, S. 33-35
- [P29] Luft, T.; Rupprecht, S.; Wartzack, S.: A concept and prototype for a new app to support collaborative and multi- criteria decision making in product development. In: Maier, A.; Škec, S.; Kim, H.; Kokkolaras, M.; Oehmen, J.; Fadel, G.; Salustri, F.; Van der Loos, M. (Hrsg.): 21st International Conference on Engineering Design (ICED 2017). Glasgow: Design Society, 2017, S. 389-398

- [P30] Luft, T.; Roth, D.; Binz, H.; Wartzack, S.: A new "knowledge-based engineering" guideline. In: Maier, A.; Škec, S.; Kim, H.; Kokkolaras, M.; Oehmen, J.; Fadel, G.; Salustri, F.; Van der Loos, M. (Hrsg.): 21st International Conference on Engineering Design (ICED 2017), Glasgow: Design Society, 2017, S. 207-216
- [P31] Luft, T.; Roth, D.; Binz, H.; Wartzack, S.: Die neue VDI-Richtlinie 5610 Blatt 2 „Wissensbasierte Konstruktion“. In: Binz, H.; Bertsche, B.; Bauer, W.; Spath, D.; Roth, D. (Hrsg.): 4. Stuttgarter Symposium für Produktentwicklung (SSP 2017), Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), 2017, o. S.
- [P32] Luft, T.; Schuh, P.; Wartzack, S.: A Software Tool for the Graph-based Visualization of Complex Relations between Product Properties and Characteristics. In: Chakrabarti, A.; Chakrabarti, D. (Hrsg.): 6th International Conference on Research into Design (ICoRD 2017). Research into Design for Communities, Vol. 1, Singapore: Springer, 2017, S. 713-724
- [P33] Reitmeier, J.; Luft, T.; Paetzold, K.; Wartzack, S.: Consideration of Uncertainties in the Product Development Process. In: Schabacker, M.; Gericke, K.; Szélig, N.; Vajna, S. (Hrsg.): Modelling and Management of Engineering Processes. 3rd International Conference 2013 (MMEP 2013). Berlin: Springer, 2015, S. 137-149
- [P34] Schweigert, S.; Luft, T.; Wartzack, S.; Lindemann, U.: Combination of matrix-based and graph-based modeling for product and organizational structures. In: Hölttä-Otto, K.; Browning, T. R.; Eppinger, S. T.; Becerril, L. (Hrsg.): 19th International Dependency and structure modeling conference (DSM 2017). München: Lehrstuhl für Produktentwicklung, Technische Universität München (TUM), 2017, S. 227-236
- [P35] Tüchsen, J.; Obendorfer, M.; Luft, T.; Schleich, B.; Wartzack, S.: Methode zur Einbindung von heuristischem Wissen in die erweiterte matrixbasierte Produktbeschreibung. In: Rieg, F.; Brökel, K.; Scharr, G.; Grote K.; Müller, N.; Lohrengel, A.; Nagarajah, A.; Corves, B. (Hrsg.): 16. Gemeinsames Kolloquium Konstruktionstechnik (KT 2018). Bayreuth: Lehrstuhl für Konstruktionslehre und CAD, Universität Bayreuth, 2018, S. 337-348
- [P36] Wartzack, S.; Luft, T.: Praxisorientierte Lehre: Konzeptentwicklung innovativer Produkte (KIP). In: Newsletter Wissenschaftliche Gesellschaft für Produktentwicklung, WiGeP, 2(2015), S. 3-4.

Verzeichnis promotionsbezogener, studentischer Arbeiten

- [S1] Bausewein, N.: *Ganzheitlicher Entscheidungsprozess zur Vergabe von Entwicklungsprojekten an externe Dienstleister*. Bachelorarbeit (2012), Erlangen
- [S2] Bayer, T.: *Semi-formale Erstellung und Bewertung technischer Produktmodelle zur Implementierung von Vergessensmechanismen in der Produktentwicklung*. Masterarbeit (2017), Erlangen
- [S3] Beck, M.: *Entwicklung und Implementierung eines Konzepts zur computerunterstützten Verknüpfung der Prozessmodellierung mit der matrixbasierten Produktbeschreibung und deren Visualisierung*. Projektarbeit (2015), Erlangen
- [S4] Beck, M.: *Entwicklung und prototypenhafte Implementierung eines Konzepts zur matrixbasierten Integration von Anforderungs-, Produkt- und Prozessmodellen*. Masterarbeit (2016), Erlangen
- [S5] Bertl, D.: *Die matrixbasierte Produktbeschreibung als Bestandteil der eigenschaftsbasierten Produktentwicklung – Validierung am Demonstrator Fahrrad*. Masterarbeit (2013), Erlangen
- [S6] Bochmann, J.: *An improved information flow in the product development process between the affected departments – an industrial case study*. Masterarbeit (2012), Erlangen – Ausgezeichnet mit dem Studienpreis 2013 der SEW-EURODRIVE-Stiftung im Bereich Maschinenbau [Dotierung: 2.500 €]
- [S7] Breyer, M.: *Ein Ansatz für die automatisierte Akquise von geometrischen Produktmerkmalen aus CATIA zur matrixbasierten Produktmodellierung*. Masterarbeit (2016), Erlangen
- [S8] Burkhardt, T.: *Anwendung und Bewertung von IT-Werkzeugen zur matrix- und graphbasierten Visualisierung komplexer Produktstrukturen*. Projektarbeit (2017), Erlangen

- [S9] Dieckrueger, M.: *Entwicklung und prototypenhafte Implementierung eines Konzepts zur computerunterstützten holistischen Analyse von Produkt- und Prozessmodellen*. Masterarbeit (2016), Erlangen
- [S10] Erfurt, A.: *Erarbeitung und Auswertung einer Umfrage zur Rolle des Wissensingenieurs in Unternehmen*. Projektarbeit (2012), Erlangen
- [S11] Ewringmann, N.: *Anwendung und Erweiterung der matrixbasierten Systembeschreibung in der eigenschaftsbasierten Produktentwicklung am Demonstrator Fahrwerk*. Diplomarbeit (2013), Erlangen
- [S12] Feldmann, D.: *Ein Ansatz zur Kombination der CAD-Modellierung mit der matrixbasierten Produktbeschreibung - Anwendung am Demonstrator Fahrrad*. Bachelorarbeit (2014), Erlangen
- [S13] Feldmann, D.: *Integration des Kostenmanagements in die matrixbasierte Modellierung der Produkt- und Wissensdomäne zur zielgerichteten Anforderungserfüllung*. Masterarbeit (2017), Erlangen
- [S14] Frank, S.: *Complexity Management in Product Development and Engineering Processes: Application and Comparison of Lean Development, Simultaneous Engineering and Matrix-based Approaches*. Masterarbeit (2014), Erlangen
- [S15] Frey, D.: *Evaluation bestehender Wissensmanagementlösungen*. Masterarbeit (2012), Erlangen
- [S16] Gietl, C.: *Computerunterstützte, graphbasierte Visualisierung von integrierten Produkt-, Prozess- und Wissensmodellen*. Projektarbeit (2017), Erlangen
- [S17] Griesbeck, F.: *Konzeption und industrielle Validierung eines IT-gestützten Vorgehensmodells für die Wert-analyse an Rohrleitungssystemen*. Masterarbeit (2015), Erlangen

- [S18] Hartmann, A.: *Kostenorientiertes Variantenmanagement im Rahmen der matrixbasierten Produktentwicklung*. Projektarbeit (2014), Erlangen
- [S19] Heid, F.: *TRIZ-szenariobasierte Technologiebewertung für eine zukunftsorientierte Produktentwicklung*. Masterarbeit (2015), Erlangen – Ausgezeichnet mit dem TRIZ Studentenpreis der International TRIZ Association (MA TRIZ) [Dotierung: Buchpreise und Übernahme von Reise- und Übernachtungskosten für eine TRIZ-Konferenz]
- [S20] Heublein, M.: *Entwicklung einer (teil-)automatisierten, matrixbasierten Modellierung der Organisations- und Prozessdomäne für das integrierte Kostenmanagement*. Masterarbeit (2017), Erlangen
- [S21] Hittinger, M.: *Entwicklung und Implementierung einer IT-Schnittstelle zur Nutzung von KBE-Anwendungen zur matrixbasierten Produktmodellierung*. Projektarbeit (2016), Erlangen
- [S22] Hittinger, M.: *Mathematische Modellierung von Relationen im matrixbasierten Produktmodell und computerunterstützte Modellierung von domänenübergreifenden Relationen*. Masterarbeit (2017), Erlangen
- [S23] Hofmann, M.: *Entwicklung eines IT-Werkzeuges zur Visualisierung der matrixbasierten Produktmodellierung*. Projektarbeit (2016), Erlangen
- [S24] Hofmann, M.: *Computerunterstützte Modellierung, Manipulation und Visualisierung von Anforderungen, Produkt- und Prozessmodellen*. Masterarbeit (2017), Erlangen
- [S25] Jakobs, B.: *Ein IT-Werkzeug zur graphenbasierten Visualisierung von matrixbasierten Produktmodellen*. Masterarbeit (2016), Erlangen

- [S26] Jin, Y.: *Ein Beitrag zur automatisierten, matrixbasierten Modellierung der integrierten Produkt- und Prozessdomäne in der Produktentwicklung*. Projektarbeit (2016), Erlangen
- [S27] Kahl, M.: *Konzeption und Umsetzung eines Datenbanksystems mit MS Access zur effizienten Verwaltung und bedarfsgerechten Bereitstellung von Decision Making Literatur*. Bachelorarbeit (2013), Erlangen
- [S28] Kahl, M.: *Entwicklung eines Softwareprototyps zur mathematischen Modellierung der Relationen zwischen Produktelementen*. Projektarbeit (2017), Erlangen.
- [S29] Kahl, M.: *Computerunterstützte Visualisierung und Bewertung von Modifikationen in der Produktarchitektur*. Masterarbeit (2017), Erlangen
- [S30] Knörnschild, A.: *Erarbeitung und Auswertung einer Umfrage zur Wissens- und Informationssuche und -bereitstellung für den Produktentwickler im Produktentwicklungsprozess*. Projektarbeit (2012), Erlangen
- [S31] Kohl, M.: *Erarbeitung einer Methodik zur Ableitung von Anforderungen aus dem Produktlebenszyklus*. Bachelorarbeit (2014), Erlangen
- [S32] Kohl, M.: *Konzept zur matrixbasierten Integration der Produkt-, Prozess-, Wissens- und Informations- sowie Organisationsmodellierung*. Projektarbeit (2016), Erlangen
- [S33] Konopik, T.: *Bewertung von Systemmodellierungsansätzen und Entwicklung eines Konzeptes zur computergestützten Integration ausgewählter Systemmodellierungsansätze in matrixbasierter Modellierung*. Bachelorarbeit (2017), Erlangen
- [S34] Konrad, S.: *Erarbeitung eines Ansatzes für das Produktreifegradmanagement in der eigenschaftsbasierten Produktentwicklung*. Masterarbeit (2014), Erlangen

- [S35] Kosog, C.: *Analyse kultureller Einflüsse und deren Integration in Methoden und Werkzeuge für die Entscheidungsfindung in der Produktentwicklung*. Bachelorarbeit (2017), Erlangen
- [S36] Krenzer, P.: *Konzeptentwicklung und prototypenhafte Implementierung eines Decision Support Systems zur Unterstützung von Entscheidungsprozessen in den frühen Phasen des Innovationsprozesses*. Masterarbeit (2014), Erlangen
- [S37] Krieglmeier, D.: *Entwicklung eines matrix- und graphbasierten Modellierungskonzepts zur Verknüpfung der Produkt- und Prozessmodellierung*. Projektarbeit (2016), Erlangen
- [S38] Kuhn, N.: *Anwendung und Evaluierung verschiedener Vorgehensweisen zur Erstellung von matrix- und CAD-basierten Produktmodellen am Beispiel eines Teilsystems einer Vorderradaufhängung*. Projektarbeit (2015), Erlangen
- [S39] Morische, J.: *Prozessmodellierung in der Produktentwicklung - Analyse und Auswahl einer Modellierungssprache für Entscheidungsprozesse*. Projektarbeit (2015), Erlangen
- [S40] Mott, S.: *Entwicklung eines Methodenbaukastens zur Kostenreduzierung in der Produktentwicklung*. Masterarbeit (2013), Erlangen
- [S41] Müller, J.: *Validation of Different Scientific Approaches in the Field of Decision Making*. Bachelorarbeit (2013), Erlangen
- [S42] Müller, P.: *Identifizierung, Analyse und Bewertung des Einsatzpotenzials von verschiedenen Decision Support Systems in der Produktentwicklung*. Projektarbeit (2015), Erlangen
- [S43] Noppenberger, T.: *Anwendung, Evaluation und Weiterentwicklung eines Assistenzsystems zur Komplexitätsbeherrschung am Beispiel eines Formula Student Rennwagens*. Projektarbeit (2017), Erlangen
- [S44] Peijan, A.: *Systematische Literaturrecherche und -analyse zu den Grundlagen der Entscheidungstheorie*. Projektarbeit (2015), Erlangen

- [S45] Pöllot, P.: *Entwicklung und Anwendung einer methodischen Vorgehensweise zur prozessorientierten Verknüpfung von Domänen der Produktentwicklung*, Masterarbeit (2014), Erlangen
- [S46] Rahmig, R.: *Modellierung des Freigabeentwicklungsprozesses und Entwicklung eines Werkzeugs für die Prozessprognose und -steuerung*. Masterarbeit (2015), Erlangen
- [S47] Rösch, M.: *Decision Making in der Produktentwicklung - Evaluation und Kategorisierung des aktuellen Forschungsstands und Umsetzung in einem rechnerunterstützten Tool*. Masterarbeit (2013), Erlangen
- [S48] Rupp, A.: *Ein Ansatz zur matrixbasierten Modellierung, Analyse und Visualisierung der Organisationsdomäne*. Bachelorarbeit (2017), Erlangen
- [S49] Rupprecht, S.: *Parametrisierbare Wertfunktionen zur Bewertung von Produkteigenschaften in der Produktentwicklung*. Projektarbeit (2015), Erlangen
- [S50] Rupprecht, S.: *Developing a concept for an app to support collaborative decision making in product development*. Masterarbeit (2016), Erlangen
- [S51] Schlenz, S.: *Entwicklung eines Tools zur automatisierten Transformation eines CAD-Modells aus PTC Creo in ein matrixbasiertes Produktmodell*. Bachelorarbeit (2017), Erlangen
- [S52] Schneider, S.: *Methodology to model and map interconnected decision making situations in product development processes*. Projektarbeit (2014), Erlangen
- [S53] Schuh, P.: *Entwicklung eines Softwaretools mit MATLAB zur graphenbasierten Visualisierung von komplexen Relationen zwischen Produktmerkmalen und -eigenschaften*. Bachelorarbeit (2015), Erlangen

- [S54] Schwarz, A.: *Entwicklung und prototypenhafte Implementierung weiterer matrixbasierter Analysemethoden für das Assistenzsystem zum Komplexitätsmanagement*. Bachelorarbeit (2017), Erlangen
- [S55] Seifert, J.: *Implementierung einer IT-Schnittstelle zur Überführung nicht nutzerparametrisierter Geometriemodelle in matrixbasierte Produktmodelle*. Projektarbeit (2016), Erlangen
- [S56] Singer, R.: *Entwicklung einer Methodik zur Aufwands- und Kostenschätzung in der Produktentwicklung und Umsetzung in einem rechnerunterstützten Tool*. Diplomarbeit (2013), Erlangen
- [S57] Smoll, S.: *Entwicklung einer kennzahlenbasierten Bewertungsmethodik zur Überwachung und Optimierung von Produktentstehungsprozessen – eine industrielle Fallstudie*. Diplomarbeit (2012), Erlangen
- [S58] Ströbel, P.: *Die matrixbasierte Produktbeschreibung am Demonstrator Fahrrad – Evaluierung geeigneter Software zur Rechnerunterstützung*. Diplomarbeit (2014), Erlangen
- [S59] Wachsmuth, P.: *Entwicklung einer IT-Schnittstelle zur automatisierten Überführung von im CAD-Modell modellierten Maschinenelementen in matrixbasierte Produktmodelle*. Projektarbeit (2016), Erlangen
- [S60] Weisgerber, F.: *Bewertung von PLM-Systemen zur Entwicklung eines Konzepts für die teilautomatisierte Transformation von Prozess- und Wissenselementen aus PLM-System in matrixbasierte Modelle*. Projektarbeit (2017), Erlangen
- [S61] Zha, H.: *Konzeptionierung und Implementierung einer IT-Schnittstelle zwischen ADONIS und dem Prozessmodell im Assistenzsystem sowie einer computerunterstützten Modellierung von Relationen*. Projektarbeit (2017), Erlangen

- [S62] Zhang, P.: *Die matrixbasierte Produktentwicklungsbeschreibung als ein Ansatz zur Modellierung von Relationen – Anwendung und Evaluierung am Demonstrator Fahrrad*. Projektarbeit (2014), Erlangen
- [S63] Zhu, J.: *Integration der FMEA-Methode in den Prototypen einer App zur Entscheidungsunterstützung in der Produktentwicklung*. Projektarbeit (2016), Erlangen
- [S64] Zucker, K.: *Einsatz CAD-integrierte KBE-Funktionen zur matrixbasierten Produktbeschreibung – eine konzeptionelle Betrachtung*. Projektarbeit (2016), Erlangen
- [S65] Zucker, K.: *Entwicklung eines Reifegradmodells für mechanische und mechatronische Produkt in der Produktentstehung*. Masterarbeit (2017), Erlangen

Reihenübersicht

Koordination der Reihe (Stand 2022):
Geschäftsstelle Maschinenbau, Dr.-Ing. Oliver Kreis, www.mb.fau.de/diss/

Im Rahmen der Reihe sind bisher die nachfolgenden Bände erschienen.

Band 1 – 52
Fertigungstechnik – Erlangen
ISSN 1431-6226
Carl Hanser Verlag, München

Band 53 – 307
Fertigungstechnik – Erlangen
ISSN 1431-6226
Meisenbach Verlag, Bamberg

ab Band 308
FAU Studien aus dem Maschinenbau
ISSN 2625-9974
FAU University Press, Erlangen

Die Zugehörigkeit zu den jeweiligen Lehrstühlen ist wie folgt gekennzeichnet:

Lehrstühle:

FAPS	Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik
FMT	Lehrstuhl für Fertigungsmesstechnik
KTmfk	Lehrstuhl für Konstruktionstechnik
LFT	Lehrstuhl für Fertigungstechnologie
LPT	Lehrstuhl für Photonische Technologien
REP	Lehrstuhl für Ressourcen- und Energieeffiziente Produktionsmaschinen

Band 1: Andreas Hemberger
Innovationspotentiale in der
rechnerintegrierten Produktion durch
wissensbasierte Systeme
FAPS, 208 Seiten, 107 Bilder. 1988.
ISBN 3-446-15234-2.

Band 2: Detlef Classe
Beitrag zur Steigerung der Flexibilität
automatisierter Montagesysteme
durch Sensorintegration und erweiterte
Steuerungskonzepte
FAPS, 194 Seiten, 70 Bilder. 1988.
ISBN 3-446-15529-5.

Band 3: Friedrich-Wilhelm Nolting
Projektierung von Montagesystemen
FAPS, 201 Seiten, 107 Bilder, 1 Tab. 1989.
ISBN 3-446-15541-4.

Band 4: Karsten Schlüter
Nutzungsgradsteigerung von
Montagesystemen durch den Einsatz
der Simulationstechnik
FAPS, 177 Seiten, 97 Bilder. 1989.
ISBN 3-446-15542-2.

Band 5: Shir-Kuan Lin
Aufbau von Modellen zur Lageregelung
von Industrierobotern
FAPS, 168 Seiten, 46 Bilder. 1989.
ISBN 3-446-15546-5.

Band 6: Rudolf Nuss
Untersuchungen zur Bearbeitungsqualität
im Fertigungssystem Laserstrahlschneiden
LFT, 206 Seiten, 115 Bilder, 6 Tab. 1989.
ISBN 3-446-15783-2.

Band 7: Wolfgang Scholz
Modell zur datenbankgestützten Planung
automatisierter Montageanlagen
FAPS, 194 Seiten, 89 Bilder. 1989.
ISBN 3-446-15825-1.

Band 8: Hans-Jürgen Wißmeier
Beitrag zur Beurteilung des Bruchverhal-
tens von Hartmetall-Fließpreßmatrizen
LFT, 179 Seiten, 99 Bilder, 9 Tab. 1989.
ISBN 3-446-15921-5.

Band 9: Rainer Eisele
Konzeption und Wirtschaftlichkeit von
Planungssystemen in der Produktion
FAPS, 183 Seiten, 86 Bilder. 1990.
ISBN 3-446-16107-4.

Band 10: Rolf Pfeiffer
Technologisch orientierte
Montageplanung am Beispiel der Schraub-
technik
FAPS, 216 Seiten, 102 Bilder, 16 Tab. 1990.
ISBN 3-446-16161-9.

Band 11: Herbert Fischer
Verteilte Planungssysteme zur
Flexibilitätssteigerung der
rechnerintegrierten Teilefertigung
FAPS, 201 Seiten, 82 Bilder. 1990.
ISBN 3-446-16105-8.

Band 12: Gerhard Kleineidam
CAD/CAP: Rechnergestützte Montage-
feinplanung
FAPS, 203 Seiten, 107 Bilder. 1990.
ISBN 3-446-16112-0.

Band 13: Frank Vollertsen
Pulvermetallurgische Verarbeitung eines
übereutektoiden verschleißfesten Stahls
LFT, XIII u. 217 Seiten, 67 Bilder, 34 Tab.
1990. ISBN 3-446-16133-3.

Band 14: Stephan Biermann
Untersuchungen zur Anlagen- und
Prozeßdiagnostik für das Schneiden
mit CO₂-Hochleistungslasern
LFT, VIII u. 170 Seiten, 93 Bilder, 4 Tab.
1991. ISBN 3-446-16269-0.

Band 15: Uwe Geißler
Material- und Datenfluß in einer flexiblen
Blechbearbeitungszelle
LFT, 124 Seiten, 41 Bilder, 7 Tab. 1991.
ISBN 3-446-16358-1.

Band 16: Frank Oswald Hake
Entwicklung eines rechnergestützten
Diagnosesystems für automatisierte
Montagezellen
FAPS, XIV u. 166 Seiten, 77 Bilder. 1991.
ISBN 3-446-16428-6.

Band 17: Herbert Reichel
Optimierung der Werkzeugbereitstellung
durch rechnergestützte
Arbeitsfolgenbestimmung
FAPS, 198 Seiten, 73 Bilder, 2 Tab. 1991.
ISBN 3-446-16453-7.

Band 18: Josef Scheller
Modellierung und Einsatz von
Softwaresystemen für rechnergeführte
Montagezellen
FAPS, 198 Seiten, 65 Bilder. 1991.
ISBN 3-446-16454-5.

Band 19: Arnold vom Ende
Untersuchungen zum Biegeumforme mit
elastischer Matrize
LFT, 166 Seiten, 55 Bilder, 13 Tab. 1991.
ISBN 3-446-16493-6.

Band 20: Joachim Schmid
Beitrag zum automatisierten Bearbeiten
von Keramikguß mit Industrierobotern
FAPS, XIV u. 176 Seiten, 111 Bilder, 6 Tab.
1991. ISBN 3-446-16560-6.

Band 21: Egon Sommer
Multiprozessorsteuerung für
kooperierende Industrieroboter in
Montagezellen
FAPS, 188 Seiten, 102 Bilder. 1991.
ISBN 3-446-17062-6.

Band 22: Georg Geyer
Entwicklung problemspezifischer
Verfahrensketten in der Montage
FAPS, 192 Seiten, 112 Bilder. 1991.
ISBN 3-446-16552-5.

Band 23: Rainer Flohr
Beitrag zur optimalen
Verbindungstechnik in der
Oberflächenmontage (SMT)
FAPS, 186 Seiten, 79 Bilder. 1991.
ISBN 3-446-16568-1.

Band 24: Alfons Rief
Untersuchungen zur Verfahrensfolge
Laserstrahlschneiden und -schweißen
in der Rohkarosseriefertigung
LFT, VI u. 145 Seiten, 58 Bilder, 5 Tab.
1991. ISBN 3-446-16593-2.

Band 25: Christoph Thim
Rechnerunterstützte Optimierung
von Materialflußstrukturen in der
Elektronikmontage durch Simulation
FAPS, 188 Seiten, 74 Bilder. 1992.
ISBN 3-446-17118-5.

Band 26: Roland Müller
CO₂-Laserstrahlschneiden von
kurzglasverstärkten Verbundwerkstoffen
LFT, 141 Seiten, 107 Bilder, 4 Tab. 1992.
ISBN 3-446-17104-5.

Band 27: Günther Schäfer
Integrierte Informationsverarbeitung
bei der Montageplanung
FAPS, 195 Seiten, 76 Bilder. 1992.
ISBN 3-446-17117-7.

Band 28: Martin Hoffmann
Entwicklung einer
CAD/CAM-Prozeßkette für die
Herstellung von Blechbiegeteilen
LFT, 149 Seiten, 89 Bilder. 1992.
ISBN 3-446-17154-1.

Band 29: Peter Hoffmann
Verfahrensfolge Laserstrahlschneiden
und -schweißen: Prozeßführung und
Systemtechnik in der 3D-Laserstrahlbear-
beitung von Blechformteilen
LFT, 186 Seiten, 92 Bilder, 10 Tab. 1992.
ISBN 3-446-17153-3.

Band 30: Olaf Schrödel
Flexible Werkstattsteuerung mit
objektorientierten Softwarestrukturen
FAPS, 180 Seiten, 84 Bilder. 1992.
ISBN 3-446-17242-4.

Band 31: Hubert Reinisch
Planungs- und Steuerungswerkzeuge zur
impliziten Geräteprogrammierung
in Roboterzellen
FAPS, XI u. 212 Seiten, 112 Bilder. 1992.
ISBN 3-446-17380-3.

Band 32: Brigitte Bärnreuther
Ein Beitrag zur Bewertung des Kommuni-
kationsverhaltens von Automatisierungs-
geräten in flexiblen Produktionszellen
FAPS, XI u. 179 Seiten, 71 Bilder. 1992.
ISBN 3-446-17451-6.

Band 33: Joachim Hutfless
Laserstrahlregelung und Optikdiagnostik
in der Strahlführung einer
CO₂-Hochleistungslaseranlage
LFT, 175 Seiten, 70 Bilder, 17 Tab. 1993.
ISBN 3-446-17532-6.

Band 34: Uwe Günzel
Entwicklung und Einsatz eines Simula-
tionsverfahrens für operative und
strategische Probleme der
Produktionsplanung und -steuerung
FAPS, XIV u. 170 Seiten, 66 Bilder, 5 Tab.
1993. ISBN 3-446-17604-7.

Band 35: Bertram Ehmann
Operatives Fertigungscontrolling durch
Optimierung auftragsbezogener Bearbei-
tungsabläufe in der Elektronikfertigung
FAPS, XV u. 167 Seiten, 114 Bilder. 1993.
ISBN 3-446-17658-6.

Band 36: Harald Kolléra
Entwicklung eines benutzerorientierten
Werkstattprogrammiersystems für das
Laserstrahlschneiden
LFT, 129 Seiten, 66 Bilder, 1 Tab. 1993.
ISBN 3-446-17719-1.

Band 37: Stephanie Abels
Modellierung und Optimierung von
Montageanlagen in einem integrierten
Simulationssystem
FAPS, 188 Seiten, 88 Bilder. 1993.
ISBN 3-446-17731-0.

Band 38: Robert Schmidt-Hebbel
Laserstrahlbohren durchflußbestimmen-
der Durchgangslöcher
LFT, 145 Seiten, 63 Bilder, 11 Tab. 1993.
ISBN 3-446-17778-7.

Band 39: Norbert Lutz
Oberflächenfeinbearbeitung
keramischer Werkstoffe mit
XeCl-Excimerlaserstrahlung
LFT, 187 Seiten, 98 Bilder, 29 Tab. 1994.
ISBN 3-446-17970-4.

Band 40: Konrad Grampp
Rechnerunterstützung bei Test und
Schulung an Steuerungssoftware von
SMD-Bestücklinien
FAPS, 178 Seiten, 88 Bilder. 1995.
ISBN 3-446-18173-3.

Band 41: Martin Koch
Wissensbasierte Unterstützung der
Angebotsbearbeitung in der
Investitionsgüterindustrie
FAPS, 169 Seiten, 68 Bilder. 1995.
ISBN 3-446-18174-1.

Band 42: Armin Gropp
Anlagen- und Prozeßdiagnostik beim
Schneiden mit einem gepulsten
Nd:YAG-Laser
LFT, 160 Seiten, 88 Bilder, 7 Tab. 1995.
ISBN 3-446-18241-1.

Band 43: Werner Heckel
Optische 3D-Konturerfassung und
on-line Biegewinkelmessung mit
dem Lichtschnittverfahren
LFT, 149 Seiten, 43 Bilder, 11 Tab. 1995.
ISBN 3-446-18243-8.

Band 44: Armin Rothhaupt
Modulares Planungssystem zur
Optimierung der Elektronikfertigung
FAPS, 180 Seiten, 101 Bilder. 1995.
ISBN 3-446-18307-8.

Band 45: Bernd Zöllner
Adaptive Diagnose in der
Elektronikproduktion
FAPS, 195 Seiten, 74 Bilder, 3 Tab. 1995.
ISBN 3-446-18308-6.

Band 46: Bodo Vormann
Beitrag zur automatisierten
Handhabungsplanung komplexer
Blechbiegeteile
LFT, 126 Seiten, 89 Bilder, 3 Tab. 1995.
ISBN 3-446-18345-0.

Band 47: Peter Schnepf
Zielkostenorientierte Montageplanung
FAPS, 144 Seiten, 75 Bilder. 1995.
ISBN 3-446-18397-3.

Band 48: Rainer Klotzbücher
Konzept zur rechnerintegrierten
Materialversorgung in flexiblen
Fertigungssystemen
FAPS, 156 Seiten, 62 Bilder. 1995.
ISBN 3-446-18412-0.

Band 49: Wolfgang Greska
Wissensbasierte Analyse und
Klassifizierung von Blechteilen
LFT, 144 Seiten, 96 Bilder. 1995.
ISBN 3-446-18462-7.

Band 50: Jörg Franke
Integrierte Entwicklung neuer
Produkt- und Produktionstechnologien
für räumliche spritzgegossene
Schaltungsträger (3-D MID)
FAPS, 196 Seiten, 86 Bilder, 4 Tab. 1995.
ISBN 3-446-18448-1.

Band 51: Franz-Josef Zeller
Sensorplanung und schnelle
Sensorregelung für Industrieroboter
FAPS, 190 Seiten, 102 Bilder, 9 Tab. 1995.
ISBN 3-446-18601-8.

Band 52: Michael Solvie
Zeitbehandlung und
Multimedia-Unterstützung in
Feldkommunikationssystemen
FAPS, 200 Seiten, 87 Bilder, 35 Tab. 1996.
ISBN 3-446-18607-7.

Band 53: Robert Hopperdietzel
Reengineering in der Elektro- und
Elektronikindustrie
FAPS, 180 Seiten, 109 Bilder, 1 Tab. 1996.
ISBN 3-87525-070-2.

Band 54: Thomas Rebhahn
Beitrag zur Mikromaterialbearbeitung
mit Excimerlasern - Systemkomponenten
und Verfahrensoptimierungen
LFT, 148 Seiten, 61 Bilder, 10 Tab. 1996.
ISBN 3-87525-075-3.

Band 55: Henning Hanebuth
Laserstrahlhartlöten mit
Zweistrahlschweißtechnik
LFT, 157 Seiten, 58 Bilder, 11 Tab. 1996.
ISBN 3-87525-074-5.

Band 56: Uwe Schönherr
Steuerung und Sensordatenintegration
für flexible Fertigungszellen mit
kooperierenden Robotern
FAPS, 188 Seiten, 116 Bilder, 3 Tab. 1996.
ISBN 3-87525-076-1.

Band 57: Stefan Holzer
Berührungslose Formgebung mit
Laserstrahlung
LFT, 162 Seiten, 69 Bilder, 11 Tab. 1996.
ISBN 3-87525-079-6.

Band 58: Markus Schultz
Fertigungsqualität beim
3D-Laserstrahlschweißen von
Blechformteilen
LFT, 165 Seiten, 88 Bilder, 9 Tab. 1997.
ISBN 3-87525-080-X.

Band 59: Thomas Krebs
Integration elektromechanischer
CA-Anwendungen über einem
STEP-Produktmodell
FAPS, 198 Seiten, 58 Bilder, 8 Tab. 1997.
ISBN 3-87525-081-8.

Band 60: Jürgen Sturm
Prozeßintegrierte Qualitätssicherung
in der Elektronikproduktion
FAPS, 167 Seiten, 112 Bilder, 5 Tab. 1997.
ISBN 3-87525-082-6.

Band 61: Andreas Brand
Prozesse und Systeme zur Bestückung
räumlicher elektronischer Baugruppen
(3D-MID)
FAPS, 182 Seiten, 100 Bilder. 1997.
ISBN 3-87525-087-7.

Band 62: Michael Kauf
Regelung der Laserstrahlleistung und
der Fokusparameter einer
CO₂-Hochleistungslaseranlage
LFT, 140 Seiten, 70 Bilder, 5 Tab. 1997.
ISBN 3-87525-083-4.

Band 63: Peter Steinwasser
Modulares Informationsmanagement
in der integrierten Produkt- und
Prozeßplanung
FAPS, 190 Seiten, 87 Bilder. 1997.
ISBN 3-87525-084-2.

Band 64: Georg Liedl
Integriertes Automatisierungskonzept
für den flexiblen Materialfluß in der
Elektronikproduktion
FAPS, 196 Seiten, 96 Bilder, 3 Tab. 1997.
ISBN 3-87525-086-9.

Band 65: Andreas Otto
Transiente Prozesse beim
Laserstrahlschweißen
LFT, 132 Seiten, 62 Bilder, 1 Tab. 1997.
ISBN 3-87525-089-3.

Band 66: Wolfgang Blöchl
Erweiterte Informationsbereitstellung
an offenen CNC-Steuerungen zur
Prozeß- und Programmoptimierung
FAPS, 168 Seiten, 96 Bilder. 1997.
ISBN 3-87525-091-5.

Band 67: Klaus-Uwe Wolf
Verbesserte Prozeßführung und
Prozeßplanung zur Leistungs- und
Qualitätssteigerung beim
Spulenwickeln
FAPS, 186 Seiten, 125 Bilder. 1997.
ISBN 3-87525-092-3.

Band 68: Frank Backes
Technologieorientierte Bahnplanung
für die 3D-Laserstrahlbearbeitung
LFT, 138 Seiten, 71 Bilder, 2 Tab. 1997.
ISBN 3-87525-093-1.

Band 69: Jürgen Kraus
Laserstrahlumformen von Profilen
LFT, 137 Seiten, 72 Bilder, 8 Tab. 1997.
ISBN 3-87525-094-X.

Band 70: Norbert Neubauer
Adaptive Strahlführungen für
CO₂-Laseranlagen
LFT, 120 Seiten, 50 Bilder, 3 Tab. 1997.
ISBN 3-87525-095-8.

Band 71: Michael Steber
Prozeßoptimierter Betrieb flexibler
Schraubstationen in der
automatisierten Montage
FAPS, 168 Seiten, 78 Bilder, 3 Tab. 1997.
ISBN 3-87525-096-6.

Band 72: Markus Pfestorf
Funktionale 3D-Oberflächenkenngrößen
in der Umformtechnik
LFT, 162 Seiten, 84 Bilder, 15 Tab. 1997.
ISBN 3-87525-097-4.

Band 73: Volker Franke
Integrierte Planung und Konstruktion
von Werkzeugen für die Biegebearbeitung
LFT, 143 Seiten, 81 Bilder. 1998.
ISBN 3-87525-098-2.

Band 74: Herbert Scheller
Automatisierte Demontagesysteme
und recyclinggerechte Produktgestaltung
elektronischer Baugruppen
FAPS, 184 Seiten, 104 Bilder, 17 Tab. 1998.
ISBN 3-87525-099-0.

Band 75: Arthur Meßner
Kaltmassivumformung metallischer
Kleinstteile – Werkstoffverhalten,
Wirkflächenreibung, Prozeßauslegung
LFT, 164 Seiten, 92 Bilder, 14 Tab. 1998.
ISBN 3-87525-100-8.

Band 76: Mathias Glasmacher
Prozeß- und Systemtechnik zum
Laserstrahl-Mikroschweißen
LFT, 184 Seiten, 104 Bilder, 12 Tab. 1998.
ISBN 3-87525-101-6.

Band 77: Michael Schwind
Zerstörungsfreie Ermittlung mechanischer
Eigenschaften von Feinblechen mit dem
Wirbelstromverfahren
LFT, 124 Seiten, 68 Bilder, 8 Tab. 1998.
ISBN 3-87525-102-4.

Band 78: Manfred Gerhard
Qualitätssteigerung in der
Elektronikproduktion durch
Optimierung der Prozeßführung
beim Löten komplexer Baugruppen
FAPS, 179 Seiten, 113 Bilder, 7 Tab. 1998.
ISBN 3-87525-103-2.

Band 79: Elke Rauh
Methodische Einbindung der Simulation
in die betrieblichen Planungs- und
Entscheidungsabläufe
FAPS, 192 Seiten, 114 Bilder, 4 Tab. 1998.
ISBN 3-87525-104-0.

Band 80: Sorin Niederkorn
Meßeinrichtung zur Untersuchung
der Wirkflächenreibung bei umformtech-
nischen Prozessen
LFT, 99 Seiten, 46 Bilder, 6 Tab. 1998.
ISBN 3-87525-105-9.

Band 81: Stefan Schubert
Regelung der Fokusslage beim Schweißen
mit CO₂-Hochleistungslasern unter
Einsatz von adaptiven Optiken
LFT, 140 Seiten, 64 Bilder, 3 Tab. 1998.
ISBN 3-87525-106-7.

Band 82: Armando Walter Colombo
Development and Implementation of
Hierarchical Control Structures of
Flexible Production Systems Using High
Level Petri Nets
FAPS, 216 Seiten, 86 Bilder. 1998.
ISBN 3-87525-109-1.

Band 83: Otto Meedt
Effizienzsteigerung bei Demontage
und Recycling durch flexible
Demontagetechnologien und optimierte
Produktgestaltung
FAPS, 186 Seiten, 103 Bilder. 1998.
ISBN 3-87525-108-3.

Band 84: Knuth Götz
Modelle und effiziente Modellbildung
zur Qualitätssicherung in der
Elektronikproduktion
FAPS, 212 Seiten, 129 Bilder, 24 Tab. 1998.
ISBN 3-87525-112-1.

Band 85: Ralf Luchs

Einsatzmöglichkeiten leitender Klebstoffe zur zuverlässigen Kontaktierung elektronischer Bauelemente in der SMT
FAPS, 176 Seiten, 126 Bilder, 30 Tab. 1998.
ISBN 3-87525-113-7.

Band 86: Frank Pöhlau

Entscheidungsgrundlagen zur Einführung räumlicher spritzgegossener Schaltungsträger (3-D MID)
FAPS, 144 Seiten, 99 Bilder. 1999.
ISBN 3-87525-114-8.

Band 87: Roland T. A. Kals

Fundamentals on the miniaturization of sheet metal working processes
LFT, 128 Seiten, 58 Bilder, 11 Tab. 1999.
ISBN 3-87525-115-6.

Band 88: Gerhard Luhn

Implizites Wissen und technisches Handeln am Beispiel der Elektronikproduktion
FAPS, 252 Seiten, 61 Bilder, 1 Tab. 1999.
ISBN 3-87525-116-4.

Band 89: Axel Sprenger

Adaptives Streckbiegen von Aluminium-Strangpreßprofilen
LFT, 114 Seiten, 63 Bilder, 4 Tab. 1999.
ISBN 3-87525-117-2.

Band 90: Hans-Jörg Pucher

Untersuchungen zur Prozeßfolge Umformen, Bestücken und Laserstrahllöten von Mikrokontakten
LFT, 158 Seiten, 69 Bilder, 9 Tab. 1999.
ISBN 3-87525-119-9.

Band 91: Horst Arnet

Profilbiegen mit kinematischer Gestalterzeugung
LFT, 128 Seiten, 67 Bilder, 7 Tab. 1999.
ISBN 3-87525-120-2.

Band 92: Doris Schubart

Prozeßmodellierung und Technologieentwicklung beim Abtragen mit CO₂-Laserstrahlung
LFT, 133 Seiten, 57 Bilder, 13 Tab. 1999.
ISBN 3-87525-122-9.

Band 93: Adrianus L. P. Coremans

Laserstrahlsintern von Metallpulver - Prozeßmodellierung, Systemtechnik, Eigenschaften laserstrahlgesinterter Metallkörper
LFT, 184 Seiten, 108 Bilder, 12 Tab. 1999.
ISBN 3-87525-124-5.

Band 94: Hans-Martin Biehler

Optimierungskonzepte für Qualitätsdatenverarbeitung und Informationsbereitstellung in der Elektronikfertigung
FAPS, 194 Seiten, 105 Bilder. 1999.
ISBN 3-87525-126-1.

Band 95: Wolfgang Becker

Oberflächenausbildung und tribologische Eigenschaften excimerlaserstrahlbearbeiteter Hochleistungskeramiken
LFT, 175 Seiten, 71 Bilder, 3 Tab. 1999.
ISBN 3-87525-127-X.

Band 96: Philipp Hein

Innenhochdruck-Umformen von Blechpaaren: Modellierung, Prozeßauslegung und Prozeßführung
LFT, 129 Seiten, 57 Bilder, 7 Tab. 1999.
ISBN 3-87525-128-8.

Band 97: Gunter Beitinger
Herstellungs- und Prüfverfahren für
thermoplastische Schaltungsträger
FAPS, 169 Seiten, 92 Bilder, 20 Tab. 1999.
ISBN 3-87525-129-6.

Band 98: Jürgen Knoblach
Beitrag zur rechnerunterstützten
verursachungsgerechten
Angebotskalkulation von Blechteilen
mit Hilfe wissensbasierter Methoden
LFT, 155 Seiten, 53 Bilder, 26 Tab. 1999.
ISBN 3-87525-130-X.

Band 99: Frank Breitenbach
Bildverarbeitungssystem zur Erfassung der
Anschlußgeometrie elektronischer SMT-
Bauelemente
LFT, 147 Seiten, 92 Bilder, 12 Tab. 2000.
ISBN 3-87525-131-8.

Band 100: Bernd Falk
Simulationsbasierte
Lebensdauervorhersage für Werkzeuge
der Kaltmassivumformung
LFT, 134 Seiten, 44 Bilder, 15 Tab. 2000.
ISBN 3-87525-136-9.

Band 101: Wolfgang Schlögl
Integriertes Simulationsdaten-Manage-
ment für Maschinenentwicklung und
Anlagenplanung
FAPS, 169 Seiten, 101 Bilder, 20 Tab. 2000.
ISBN 3-87525-137-7.

Band 102: Christian Hinsel
Ermüdungsbruchversagen
hartstoffbeschichteter Werkzeugstähle
in der Kaltmassivumformung
LFT, 130 Seiten, 80 Bilder, 14 Tab. 2000.
ISBN 3-87525-138-5.

Band 103: Stefan Bobbert
Simulationsgestützte Prozessauslegung
für das Innenhochdruck-Umformen
von Blechpaaren
LFT, 123 Seiten, 77 Bilder. 2000.
ISBN 3-87525-145-8.

Band 104: Harald Rottbauer
Modulares Planungswerkzeug zum
Produktionsmanagement in der
Elektronikproduktion
FAPS, 166 Seiten, 106 Bilder. 2001.
ISBN 3-87525-139-3.

Band 105: Thomas Hennige
Flexible Formgebung von Blechen
durch Laserstrahlumformen
LFT, 119 Seiten, 50 Bilder. 2001.
ISBN 3-87525-140-7.

Band 106: Thomas Menzel
Wissensbasierte Methoden für die
rechnergestützte Charakterisierung
und Bewertung innovativer
Fertigungsprozesse
LFT, 152 Seiten, 71 Bilder. 2001.
ISBN 3-87525-142-3.

Band 107: Thomas Stöckel
Kommunikationstechnische Integration
der Prozeßebene in Produktionssysteme
durch Middleware-Frameworks
FAPS, 147 Seiten, 65 Bilder, 5 Tab. 2001.
ISBN 3-87525-143-1.

Band 108: Frank Pitter
Verfügbarkeitssteigerung von
Werkzeugmaschinen durch Einsatz
mechatronischer Sensorlösungen
FAPS, 158 Seiten, 131 Bilder, 8 Tab. 2001.
ISBN 3-87525-144-X.

Band 109: Markus Korneli
Integration lokaler CAP-Systeme in
einen globalen Fertigungsdatenverbund
FAPS, 121 Seiten, 53 Bilder, 11 Tab. 2001.
ISBN 3-87525-146-6.

Band 110: Burkhard Müller
Laserstrahljustieren mit Excimer-Lasern -
Prozeßparameter und Modelle zur
Aktorkonstruktion
LFT, 128 Seiten, 36 Bilder, 9 Tab. 2001.
ISBN 3-87525-159-8.

Band 111: Jürgen Göhringer
Integrierte Telediagnose via Internet
zum effizienten Service von
Produktionssystemen
FAPS, 178 Seiten, 98 Bilder, 5 Tab. 2001.
ISBN 3-87525-147-4.

Band 112: Robert Feuerstein
Qualitäts- und kosteneffiziente Integra-
tion neuer Bauelementetechnologien in
die Flachbaugruppenfertigung
FAPS, 161 Seiten, 99 Bilder, 10 Tab. 2001.
ISBN 3-87525-151-2.

Band 113: Marcus Reichenberger
Eigenschaften und Einsatzmöglichkeiten
alternativer Elektroniklote in der
Oberflächenmontage (SMT)
FAPS, 165 Seiten, 97 Bilder, 18 Tab. 2001.
ISBN 3-87525-152-0.

Band 114: Alexander Huber
Justieren vormontierter Systeme mit dem
Nd:YAG-Laser unter Einsatz von Aktoren
LFT, 122 Seiten, 58 Bilder, 5 Tab. 2001.
ISBN 3-87525-153-9.

Band 115: Sami Krimi
Analyse und Optimierung von Montage-
systemen in der Elektronikproduktion
FAPS, 155 Seiten, 88 Bilder, 3 Tab. 2001.
ISBN 3-87525-157-1.

Band 116: Marion Merklein
Laserstrahlumformen von
Aluminiumwerkstoffen - Beeinflussung
der Mikrostruktur und der mechanischen
Eigenschaften
LFT, 122 Seiten, 65 Bilder, 15 Tab. 2001.
ISBN 3-87525-156-3.

Band 117: Thomas Collisi
Ein informationslogistisches
Architekturkonzept zur Akquisition
simulationsrelevanter Daten
FAPS, 181 Seiten, 105 Bilder, 7 Tab. 2002.
ISBN 3-87525-164-4.

Band 118: Markus Koch
Rationalisierung und ergonomische
Optimierung im Innenausbau durch
den Einsatz moderner
Automatisierungstechnik
FAPS, 176 Seiten, 98 Bilder, 9 Tab. 2002.
ISBN 3-87525-165-2.

Band 119: Michael Schmidt
Prozeßregelung für das Laserstrahl-Punkt-
schweißen in der Elektronikproduktion
LFT, 152 Seiten, 71 Bilder, 3 Tab. 2002.
ISBN 3-87525-166-0.

Band 120: Nicolas Tiesler
Grundlegende Untersuchungen zum
Fließpressen metallischer Kleinstteile
LFT, 126 Seiten, 78 Bilder, 12 Tab. 2002.
ISBN 3-87525-175-X.

Band 121: Lars Pursche
Methoden zur technologieorientierten
Programmierung für die
3D-Lasermikrobearbeitung
LFT, 111 Seiten, 39 Bilder, 0 Tab. 2002.
ISBN 3-87525-183-0.

Band 122: Jan-Oliver Brassel
Prozeßkontrolle beim
Laserstrahl-Mikroschweißen
LFT, 148 Seiten, 72 Bilder, 12 Tab. 2002.
ISBN 3-87525-181-4.

Band 123: Mark Geisel
Prozeßkontrolle und -steuerung beim
Laserstrahlschweißen mit den Methoden
der nichtlinearen Dynamik
LFT, 135 Seiten, 46 Bilder, 2 Tab. 2002.
ISBN 3-87525-180-6.

Band 124: Gerd Eßer
Laserstrahlunterstützte Erzeugung
metallischer Leiterstrukturen auf
Thermoplastsubstraten für die
MID-Technik
LFT, 148 Seiten, 60 Bilder, 6 Tab. 2002.
ISBN 3-87525-171-7.

Band 125: Marc Fleckenstein
Qualität laserstrahl-gefüger
Mikroverbindungen elektronischer
Kontakte
LFT, 159 Seiten, 77 Bilder, 7 Tab. 2002.
ISBN 3-87525-170-9.

Band 126: Stefan Kaufmann
Grundlegende Untersuchungen zum
Nd:YAG- Laserstrahlfügen von Silizium
für Komponenten der Optoelektronik
LFT, 159 Seiten, 100 Bilder, 6 Tab. 2002.
ISBN 3-87525-172-5.

Band 127: Thomas Fröhlich
Simultanes Löten von Anschlußkontakten
elektronischer Bauelemente mit
Diodenlaserstrahlung
LFT, 143 Seiten, 75 Bilder, 6 Tab. 2002.
ISBN 3-87525-186-5.

Band 128: Achim Hofmann
Erweiterung der Formgebungsgrenzen
beim Umformen von
Aluminiumwerkstoffen durch den Einsatz
prozessangepasster Platinen
LFT, 113 Seiten, 58 Bilder, 4 Tab. 2002.
ISBN 3-87525-182-2.

Band 129: Ingo Kriebitzsch
3 - D MID Technologie in der
Automobilelektronik
FAPS, 129 Seiten, 102 Bilder, 10 Tab. 2002.
ISBN 3-87525-169-5.

Band 130: Thomas Pohl
Fertigungsqualität und Umformbarkeit
laserstrahlgeschweißter Formplatinen
aus Aluminiumlegierungen
LFT, 133 Seiten, 93 Bilder, 12 Tab. 2002.
ISBN 3-87525-173-3.

Band 131: Matthias Wenk
Entwicklung eines konfigurierbaren
Steuerungssystems für die flexible
Sensorführung von Industrierobotern
FAPS, 167 Seiten, 85 Bilder, 1 Tab. 2002.
ISBN 3-87525-174-1.

Band 132: Matthias Negendanck
Neue Sensorik und Aktorik für
Bearbeitungsköpfe zum
Laserstrahlschweißen
LFT, 116 Seiten, 60 Bilder, 14 Tab. 2002.
ISBN 3-87525-184-9.

Band 133: Oliver Kreis

Integrierte Fertigung - Verfahrensintegration durch Innenhochdruck-Umformen, Trennen und Laserstrahlschweißen in einem Werkzeug sowie ihre tele- und multimediale Präsentation

LFT, 167 Seiten, 90 Bilder, 43 Tab. 2002.

ISBN 3-87525-176-8.

Band 134: Stefan Trautner

Technische Umsetzung produktbezogener Instrumente der Umweltpolitik bei Elektro- und Elektronikgeräten

FAPS, 179 Seiten, 92 Bilder, 11 Tab. 2002.

ISBN 3-87525-177-6.

Band 135: Roland Meier

Strategien für einen produktorientierten Einsatz räumlicher spritzgegossener Schaltungsträger (3-D MID)

FAPS, 155 Seiten, 88 Bilder, 14 Tab. 2002.

ISBN 3-87525-178-4.

Band 136: Jürgen Wunderlich

Kostensimulation - Simulationsbasierte Wirtschaftlichkeitsregelung komplexer Produktionssysteme

FAPS, 202 Seiten, 119 Bilder, 17 Tab. 2002.

ISBN 3-87525-179-2.

Band 137: Stefan Novotny

Innenhochdruck-Umformen von Blechen aus Aluminium- und Magnesiumlegierungen bei erhöhter Temperatur

LFT, 132 Seiten, 82 Bilder, 6 Tab. 2002.

ISBN 3-87525-185-7.

Band 138: Andreas Licha

Flexible Montageautomatisierung zur Komplettmontage flächenhafter Produktstrukturen durch kooperierende Industrieroboter

FAPS, 158 Seiten, 87 Bilder, 8 Tab. 2003.

ISBN 3-87525-189-X.

Band 139: Michael Eisenbarth

Beitrag zur Optimierung der Aufbau- und Verbindungstechnik für mechatronische Baugruppen

FAPS, 207 Seiten, 141 Bilder, 9 Tab. 2003.

ISBN 3-87525-190-3.

Band 140: Frank Christoph

Durchgängige simulationsgestützte Planung von Fertigungseinrichtungen der Elektronikproduktion

FAPS, 187 Seiten, 107 Bilder, 9 Tab. 2003.

ISBN 3-87525-191-1.

Band 141: Hinnerk Hagenah

Simulationsbasierte Bestimmung der zu erwartenden Maßhaltigkeit für das Blechbiegen

LFT, 131 Seiten, 36 Bilder, 26 Tab. 2003.

ISBN 3-87525-192-X.

Band 142: Ralf Eckstein

Scherschneiden und Biegen metallischer Kleinstteile - Materialeinfluss und Materialverhalten

LFT, 148 Seiten, 71 Bilder, 19 Tab. 2003.

ISBN 3-87525-193-8.

Band 143: Frank H. Meyer-Pittroff

Excimerlaserstrahlbiegen dünner metallischer Folien mit homogener Lichtlinie

LFT, 138 Seiten, 60 Bilder, 16 Tab. 2003.

ISBN 3-87525-196-2.

Band 144: Andreas Kach

Rechnergestützte Anpassung von Laserstrahlschneidbahnen an Bauteilabweichungen

LFT, 139 Seiten, 69 Bilder, 11 Tab. 2004.

ISBN 3-87525-197-0.

Band 145: Stefan Hierl
System- und Prozeßtechnik für das
simultane Löten mit Diodenlaserstrahlung
von elektronischen Bauelementen
LFT, 124 Seiten, 66 Bilder, 4 Tab. 2004.
ISBN 3-87525-198-9.

Band 146: Thomas Neudecker
Tribologische Eigenschaften keramischer
Blechumformwerkzeuge- Einfluss einer
Oberflächenendbearbeitung mittels
Excimerlaserstrahlung
LFT, 166 Seiten, 75 Bilder, 26 Tab. 2004.
ISBN 3-87525-200-4.

Band 147: Ulrich Wenger
Prozessoptimierung in der Wickeltechnik
durch innovative maschinenbauliche und
regelungstechnische Ansätze
FAPS, 132 Seiten, 88 Bilder, 0 Tab. 2004.
ISBN 3-87525-203-9.

Band 148: Stefan Slama
Effizienzsteigerung in der Montage durch
marktorientierte Montagestrukturen und
erweiterte Mitarbeiterkompetenz
FAPS, 188 Seiten, 125 Bilder, 0 Tab. 2004.
ISBN 3-87525-204-7.

Band 149: Thomas Wurm
Laserstrahljustieren mittels Aktoren-Ent-
wicklung von Konzepten und Methoden
für die rechnerunterstützte Modellierung
und Optimierung von komplexen
Aktorsystemen in der Mikrotechnik
LFT, 122 Seiten, 51 Bilder, 9 Tab. 2004.
ISBN 3-87525-206-3.

Band 150: Martino Celeghini
Wirkmedienbasierte Blechumformung:
Grundlagenuntersuchungen zum Einfluss
von Werkstoff und Bauteilgeometrie
LFT, 146 Seiten, 77 Bilder, 6 Tab. 2004.
ISBN 3-87525-207-1.

Band 151: Ralph Hohenstein
Entwurf hochdynamischer Sensor- und
Regelsysteme für die adaptive
Laserbearbeitung
LFT, 282 Seiten, 63 Bilder, 16 Tab. 2004.
ISBN 3-87525-210-1.

Band 152: Angelika Hutterer
Entwicklung prozessüberwachender
Regelkreise für flexible
Formgebungsprozesse
LFT, 149 Seiten, 57 Bilder, 2 Tab. 2005.
ISBN 3-87525-212-8.

Band 153: Emil Egerer
Massivumformen metallischer Kleinstteile
bei erhöhter Prozesstemperatur
LFT, 158 Seiten, 87 Bilder, 10 Tab. 2005.
ISBN 3-87525-213-6.

Band 154: Rüdiger Holzmann
Strategien zur nachhaltigen Optimierung
von Qualität und Zuverlässigkeit in
der Fertigung hochintegrierter
Flachbaugruppen
FAPS, 186 Seiten, 99 Bilder, 19 Tab. 2005.
ISBN 3-87525-217-9.

Band 155: Marco Nock
Biegeumformen mit
Elastomerwerkzeugen Modellierung,
Prozessauslegung und Abgrenzung des
Verfahrens am Beispiel des Rohrbiegens
LFT, 164 Seiten, 85 Bilder, 13 Tab. 2005.
ISBN 3-87525-218-7.

Band 156: Frank Niebling
Qualifizierung einer Prozesskette zum
Laserstrahlsintern metallischer Bauteile
LFT, 148 Seiten, 89 Bilder, 3 Tab. 2005.
ISBN 3-87525-219-5.

Band 157: Markus Meiler
Großserientauglichkeit trockenschmierstoffbeschichteter Aluminiumbleche im Presswerk Grundlegende Untersuchungen zur Tribologie, zum Umformverhalten und Bauteilversuche
LFT, 104 Seiten, 57 Bilder, 21 Tab. 2005.
ISBN 3-87525-221-7.

Band 158: Agus Sutanto
Solution Approaches for Planning of Assembly Systems in Three-Dimensional Virtual Environments
FAPS, 169 Seiten, 98 Bilder, 3 Tab. 2005.
ISBN 3-87525-220-9.

Band 159: Matthias Boiger
Hochleistungssysteme für die Fertigung elektronischer Baugruppen auf der Basis flexibler Schaltungsträger
FAPS, 175 Seiten, 111 Bilder, 8 Tab. 2005.
ISBN 3-87525-222-5.

Band 160: Matthias Pitz
Laserunterstütztes Biegen höchstfester Mehrphasenstähle
LFT, 120 Seiten, 73 Bilder, 11 Tab. 2005.
ISBN 3-87525-223-3.

Band 161: Meik Vahl
Beitrag zur gezielten Beeinflussung des Werkstoffflusses beim Innenhochdruck-Umformen von Blechen
LFT, 165 Seiten, 94 Bilder, 15 Tab. 2005.
ISBN 3-87525-224-1.

Band 162: Peter K. Kraus
Plattformstrategien - Realisierung einer varianz- und kostenoptimierten Wertschöpfung
FAPS, 181 Seiten, 95 Bilder, 0 Tab. 2005.
ISBN 3-87525-226-8.

Band 163: Adrienn Cser
Laserstrahlschmelzabtrag - Prozessanalyse und -modellierung
LFT, 146 Seiten, 79 Bilder, 3 Tab. 2005.
ISBN 3-87525-227-6.

Band 164: Markus C. Hahn
Grundlegende Untersuchungen zur Herstellung von Leichtbauverbundstrukturen mit Aluminiumschaumkern
LFT, 143 Seiten, 60 Bilder, 16 Tab. 2005.
ISBN 3-87525-228-4.

Band 165: Gordana Michos
Mechatronische Ansätze zur Optimierung von Vorschubachsen
FAPS, 146 Seiten, 87 Bilder, 17 Tab. 2005.
ISBN 3-87525-230-6.

Band 166: Markus Stark
Auslegung und Fertigung hochpräziser Faser-Kollimator-Arrays
LFT, 158 Seiten, 115 Bilder, 11 Tab. 2005.
ISBN 3-87525-231-4.

Band 167: Yurong Zhou
Kollaboratives Engineering Management in der integrierten virtuellen Entwicklung der Anlagen für die Elektronikproduktion
FAPS, 156 Seiten, 84 Bilder, 6 Tab. 2005.
ISBN 3-87525-232-2.

Band 168: Werner Enser
Neue Formen permanenter und lösbarer elektrischer Kontaktierungen für mechatronische Baugruppen
FAPS, 190 Seiten, 112 Bilder, 5 Tab. 2005.
ISBN 3-87525-233-0.

Band 169: Katrin Melzer
Integrierte Produktpolitik bei elektrischen
und elektronischen Geräten zur
Optimierung des Product-Life-Cycle
FAPS, 155 Seiten, 91 Bilder, 17 Tab. 2005.
ISBN 3-87525-234-9.

Band 170: Alexander Putz
Grundlegende Untersuchungen zur
Erfassung der realen Vorspannung von
armierten Kaltfließpresswerkzeugen
mittels Ultraschall
LFT, 137 Seiten, 71 Bilder, 15 Tab. 2006.
ISBN 3-87525-237-3.

Band 171: Martin Prechtel
Automatisiertes Schichtverfahren für
metallische Folien - System- und
Prozesstechnik
LFT, 154 Seiten, 45 Bilder, 7 Tab. 2006.
ISBN 3-87525-238-1.

Band 172: Markus Meidert
Beitrag zur deterministischen
Lebensdauerabschätzung von
Werkzeugen der Kaltmassivumformung
LFT, 131 Seiten, 78 Bilder, 9 Tab. 2006.
ISBN 3-87525-239-X.

Band 173: Bernd Müller
Robuste, automatisierte Montagesysteme
durch adaptive Prozessführung und
montageübergreifende Fehlerprävention
am Beispiel flächiger Leichtbauteile
FAPS, 147 Seiten, 77 Bilder, 0 Tab. 2006.
ISBN 3-87525-240-3.

Band 174: Alexander Hofmann
Hybrides Laserdurchstrahlsschweißen
von Kunststoffen
LFT, 136 Seiten, 72 Bilder, 4 Tab. 2006.
ISBN 978-3-87525-243-9.

Band 175: Peter Wölflick
Innovative Substrate und Prozesse
mit feinsten Strukturen für bleifreie
Mechatronik-Anwendungen
FAPS, 177 Seiten, 148 Bilder, 24 Tab. 2006.
ISBN 978-3-87525-246-0.

Band 176: Attila Komlodi
Detection and Prevention of Hot Cracks
during Laser Welding of Aluminium Al-
loys Using Advanced Simulation Methods
LFT, 155 Seiten, 89 Bilder, 14 Tab. 2006.
ISBN 978-3-87525-248-4.

Band 177: Uwe Popp
Grundlegende Untersuchungen zum
Laserstrahlstrukturieren von Kaltmassiv-
umformwerkzeugen
LFT, 140 Seiten, 67 Bilder, 16 Tab. 2006.
ISBN 978-3-87525-249-1.

Band 178: Veit Rückel
Rechnergestützte Ablaufplanung und
Bahngenerierung Für kooperierende
Industrieroboter
FAPS, 148 Seiten, 75 Bilder, 7 Tab. 2006.
ISBN 978-3-87525-250-7.

Band 179: Manfred Dirscherl
Nicht-thermische Mikrojustiertechnik
mittels ultrakurzer Laserpulse
LFT, 154 Seiten, 69 Bilder, 10 Tab. 2007.
ISBN 978-3-87525-251-4.

Band 180: Yong Zhuo
Entwurf eines rechnergestützten
integrierten Systems für Konstruktion
und Fertigungsplanung räumlicher spritz-
gegossener Schaltungsträger
(3D-MID)
FAPS, 181 Seiten, 95 Bilder, 5 Tab. 2007.
ISBN 978-3-87525-253-8.

Band 181: Stefan Lang
Durchgängige Mitarbeiterinformation
zur Steigerung von Effizienz und
Prozesssicherheit in der Produktion
FAPS, 172 Seiten, 93 Bilder. 2007.
ISBN 978-3-87525-257-6.

Band 182: Hans-Joachim Krauß
Laserstrahlinduzierte Pyrolyse
präkeramischer Polymere
LFT, 171 Seiten, 100 Bilder. 2007.
ISBN 978-3-87525-258-3.

Band 183: Stefan Junker
Technologien und Systemlösungen für
die flexibel automatisierte Bestückung
permanent erregter Läufer mit
oberflächenmontierten Dauermagneten
FAPS, 173 Seiten, 75 Bilder. 2007.
ISBN 978-3-87525-259-0.

Band 184: Rainer Kohlbauer
Wissensbasierte Methoden für die
simulationsgestützte Auslegung wirk-
medienbasierter Blechumformprozesse
LFT, 135 Seiten, 50 Bilder. 2007.
ISBN 978-3-87525-260-6.

Band 185: Klaus Lamprecht
Wirkmedienbasierte Umformung
tiefgezogener Vorformen unter
besonderer Berücksichtigung
maßgeschneiderter Halbzeuge
LFT, 137 Seiten, 81 Bilder. 2007.
ISBN 978-3-87525-265-1.

Band 186: Bernd Zolleiß
Optimierte Prozesse und Systeme
für die Bestückung mechatronischer
Baugruppen
FAPS, 180 Seiten, 117 Bilder. 2007.
ISBN 978-3-87525-266-8.

Band 187: Michael Kerausch
Simulationsgestützte Prozessauslegung
für das Umformen lokal
wärmebehandelter Aluminiumplatten
LFT, 146 Seiten, 76 Bilder, 7 Tab. 2007.
ISBN 978-3-87525-267-5.

Band 188: Matthias Weber
Unterstützung der Wandlungsfähigkeit
von Produktionsanlagen durch innovative
Softwaresysteme
FAPS, 183 Seiten, 122 Bilder, 3 Tab. 2007.
ISBN 978-3-87525-269-9.

Band 189: Thomas Frick
Untersuchung der prozessbestimmenden
Strahl-Stoff-Wechselwirkungen beim
Laserstrahlschweißen von Kunststoffen
LFT, 104 Seiten, 62 Bilder, 8 Tab. 2007.
ISBN 978-3-87525-268-2.

Band 190: Joachim Hecht
Werkstoffcharakterisierung und
Prozessauslegung für die wirkmedienba-
sierte Doppelblech-Umformung von
Magnesiumlegierungen
LFT, 107 Seiten, 91 Bilder, 2 Tab. 2007.
ISBN 978-3-87525-270-5.

Band 191: Ralf Völkl
Stochastische Simulation zur Werkzeugle-
bensdaueroptimierung und Präzisionsfer-
tigung in der Kaltmassivumformung
LFT, 178 Seiten, 75 Bilder, 12 Tab. 2008.
ISBN 978-3-87525-272-9.

Band 192: Massimo Tolazzi
Innenhochdruck-Umformen verstärkter
Blech-Rahmenstrukturen
LFT, 164 Seiten, 85 Bilder, 7 Tab. 2008.
ISBN 978-3-87525-273-6.

Band 193: Cornelia Hoff
Untersuchung der Prozesseinflussgrößen
beim Presshärten des höchstfesten
Vergütungsstahls 22MnB5
LFT, 133 Seiten, 92 Bilder, 5 Tab. 2008.
ISBN 978-3-87525-275-0.

Band 194: Christian Alvarez
Simulationsgestützte Methoden zur
effizienten Gestaltung von Lötprozessen
in der Elektronikproduktion
FAPS, 149 Seiten, 86 Bilder, 8 Tab. 2008.
ISBN 978-3-87525-277-4.

Band 195: Andreas Kunze
Automatisierte Montage von makrome-
chatronischen Modulen zur flexiblen
Integration in hybride
Pkw-Bordnetzsysteme
FAPS, 160 Seiten, 90 Bilder, 14 Tab. 2008.
ISBN 978-3-87525-278-1.

Band 196: Wolfgang Hußnätter
Grundlegende Untersuchungen zur
experimentellen Ermittlung und zur
Modellierung von Fließortkurven bei
erhöhten Temperaturen
LFT, 152 Seiten, 73 Bilder, 21 Tab. 2008.
ISBN 978-3-87525-279-8.

Band 197: Thomas Bigl
Entwicklung, angepasste Herstellungsver-
fahren und erweiterte Qualitätssicherung
von einsatzgerechten elektronischen Bau-
gruppen
FAPS, 175 Seiten, 107 Bilder, 14 Tab. 2008.
ISBN 978-3-87525-280-4.

Band 198: Stephan Roth
Grundlegende Untersuchungen zum
Excimerlaserstrahl-Abtragen unter
Flüssigkeitsfilmen
LFT, 113 Seiten, 47 Bilder, 14 Tab. 2008.
ISBN 978-3-87525-281-1.

Band 199: Artur Giera
Prozesstechnische Untersuchungen
zum Rührreibschweißen metallischer
Werkstoffe
LFT, 179 Seiten, 104 Bilder, 36 Tab. 2008.
ISBN 978-3-87525-282-8.

Band 200: Jürgen Lechler
Beschreibung und Modellierung
des Werkstoffverhaltens von
presshärtbaren Bor-Manganstählen
LFT, 154 Seiten, 75 Bilder, 12 Tab. 2009.
ISBN 978-3-87525-286-6.

Band 201: Andreas Blankl
Untersuchungen zur Erhöhung der
Prozessrobustheit bei der Innenhoch-
druck-Umformung von flächigen Halb-
zeugen mit vor- bzw. nachgeschalteten
Laserstrahlfügeoperationen
LFT, 120 Seiten, 68 Bilder, 9 Tab. 2009.
ISBN 978-3-87525-287-3.

Band 202: Andreas Schaller
Modellierung eines nachfrageorientierten
Produktionskonzeptes für mobile
Telekommunikationsgeräte
FAPS, 120 Seiten, 79 Bilder, 0 Tab. 2009.
ISBN 978-3-87525-289-7.

Band 203: Claudius Schimpf
Optimierung von Zuverlässigkeitsuntersu-
chungen, Prüfabläufen und Nacharbeits-
prozessen in der Elektronikproduktion
FAPS, 162 Seiten, 90 Bilder, 14 Tab. 2009.
ISBN 978-3-87525-290-3.

Band 204: Simon Dietrich
Sensoriken zur Schwerpunktslagebestim-
mung der optischen Prozessemissionen
beim Laserstrahl-tiefschweißen
LFT, 138 Seiten, 70 Bilder, 5 Tab. 2009.
ISBN 978-3-87525-292-7.

Band 205: Wolfgang Wolf
Entwicklung eines agentenbasierten
Steuerungssystems zur
Materialflussorganisation im
wandelbaren Produktionsumfeld
FAPS, 167 Seiten, 98 Bilder. 2009.
ISBN 978-3-87525-293-4.

Band 206: Steffen Polster
Laserdurchstrahlsschweißen
transparenter Polymerbauteile
LFT, 160 Seiten, 92 Bilder, 13 Tab. 2009.
ISBN 978-3-87525-294-1.

Band 207: Stephan Manuel Dörfler
Rührreibschweißen von walzplattiertem
Halbzeug und Aluminiumblech zur
Herstellung flächiger Aluminiumschaum-
Sandwich-Verbundstrukturen
LFT, 190 Seiten, 98 Bilder, 5 Tab. 2009.
ISBN 978-3-87525-295-8.

Band 208: Uwe Vogt
Seriennahe Auslegung von Aluminium
Tailored Heat Treated Blanks
LFT, 151 Seiten, 68 Bilder, 26 Tab. 2009.
ISBN 978-3-87525-296-5.

Band 209: Till Laumann
Qualitative und quantitative Bewertung
der Crashtauglichkeit von höchstfesten
Stählen
LFT, 117 Seiten, 69 Bilder, 7 Tab. 2009.
ISBN 978-3-87525-299-6.

Band 210: Alexander Diehl
Größeneffekte bei Biegeprozessen-
Entwicklung einer Methodik zur
Identifikation und Quantifizierung
LFT, 180 Seiten, 92 Bilder, 12 Tab. 2010.
ISBN 978-3-87525-302-3.

Band 211: Detlev Staud
Effiziente Prozesskettenauslegung für das
Umformen lokal wärmebehandelter und
geschweißter Aluminiumbleche
LFT, 164 Seiten, 72 Bilder, 12 Tab. 2010.
ISBN 978-3-87525-303-0.

Band 212: Jens Ackermann
Prozesssicherung beim Laserdurchstrahl-
schweißen thermoplastischer Kunststoffe
LPT, 129 Seiten, 74 Bilder, 13 Tab. 2010.
ISBN 978-3-87525-305-4.

Band 213: Stephan Weidel
Grundlegende Untersuchungen zum
Kontaktzustand zwischen Werkstück
und Werkzeug bei umformtechnischen
Prozessen unter tribologischen
Gesichtspunkten
LFT, 144 Seiten, 67 Bilder, 11 Tab. 2010.
ISBN 978-3-87525-307-8.

Band 214: Stefan Geißdörfer
Entwicklung eines mesoskopischen
Modells zur Abbildung von Größeneffek-
ten in der Kaltmassivumformung mit
Methoden der FE-Simulation
LFT, 133 Seiten, 83 Bilder, 11 Tab. 2010.
ISBN 978-3-87525-308-5.

Band 215: Christian Matzner
Konzeption produktspezifischer Lösungen
zur Robustheitssteigerung elektronischer
Systeme gegen die Einwirkung von
Betauung im Automobil
FAPS, 165 Seiten, 93 Bilder, 14 Tab. 2010.
ISBN 978-3-87525-309-2.

Band 216: Florian Schüssler
Verbindungs- und Systemtechnik für
thermisch hochbeanspruchte und
miniaturisierte elektronische Baugruppen
FAPS, 184 Seiten, 93 Bilder, 18 Tab. 2010.
ISBN 978-3-87525-310-8.

Band 217: Massimo Cojutti
Strategien zur Erweiterung der Prozess-
grenzen bei der Innhochdruck-Umfor-
mung von Rohren und Blechpaaren
LFT, 125 Seiten, 56 Bilder, 9 Tab. 2010.
ISBN 978-3-87525-312-2.

Band 218: Raoul Plettke
Mehrkriterielle Optimierung komplexer
Aktorsysteme für das Laserstrahljustieren
LFT, 152 Seiten, 25 Bilder, 3 Tab. 2010.
ISBN 978-3-87525-315-3.

Band 219: Andreas Dobroschke
Flexible Automatisierungslösungen für
die Fertigung wickeltechnischer Produkte
FAPS, 184 Seiten, 109 Bilder, 18 Tab. 2011.
ISBN 978-3-87525-317-7.

Band 220: Azhar Zam
Optical Tissue Differentiation for
Sensor-Controlled Tissue-Specific
Laser Surgery
LPT, 99 Seiten, 45 Bilder, 8 Tab. 2011.
ISBN 978-3-87525-318-4.

Band 221: Michael Rösch
Potenziale und Strategien zur Optimie-
rung des Schablonendruckprozesses in der
Elektronikproduktion
FAPS, 192 Seiten, 127 Bilder, 19 Tab. 2011.
ISBN 978-3-87525-319-1.

Band 222: Thomas Rechtenwald
Quasi-isothermes Laserstrahlsintern von
Hochtemperatur-Thermoplasten - Eine
Betrachtung werkstoff-prozessspezifischer
Aspekte am Beispiel PEEK
LPT, 150 Seiten, 62 Bilder, 8 Tab. 2011.
ISBN 978-3-87525-320-7.

Band 223: Daniel Craiovan
Prozesse und Systemlösungen für die
SMT-Montage optischer Bauelemente auf
Substrate mit integrierten Lichtwellenlei-
tern
FAPS, 165 Seiten, 85 Bilder, 8 Tab. 2011.
ISBN 978-3-87525-324-5.

Band 224: Kay Wagner
Beanspruchungsangepasste
Kaltmassivumformwerkzeuge durch
lokal optimierte Werkzeuoberflächen
LFT, 147 Seiten, 103 Bilder, 17 Tab. 2011.
ISBN 978-3-87525-325-2.

Band 225: Martin Brandhuber
Verbesserung der Prognosegüte des Versa-
gens von Punktschweißverbindungen bei
höchstfesten Stahlgüten
LFT, 155 Seiten, 91 Bilder, 19 Tab. 2011.
ISBN 978-3-87525-327-6.

Band 226: Peter Sebastian Feuser
Ein Ansatz zur Herstellung von
pressgehärteten Karosseriekomponenten
mit maßgeschneiderten mechanischen
Eigenschaften: Temperierte Umformwerk-
zeuge. Prozessfenster, Prozesssimulation
und funktionale Untersuchung
LFT, 195 Seiten, 97 Bilder, 60 Tab. 2012.
ISBN 978-3-87525-328-3.

Band 227: Murat Arbak
Material Adapted Design of Cold Forging
Tools Exemplified by Powder
Metallurgical Tool Steels and Ceramics
LFT, 109 Seiten, 56 Bilder, 8 Tab. 2012.
ISBN 978-3-87525-330-6.

Band 228: Indra Pitz
Beschleunigte Simulation des
Laserstrahlumformens von
Aluminiumblechen
LPT, 137 Seiten, 45 Bilder, 27 Tab. 2012.
ISBN 978-3-87525-333-7.

Band 229: Alexander Grimm
Prozessanalyse und -überwachung des
Laserstrahlhartlötens mittels optischer
Sensorik
LPT, 125 Seiten, 61 Bilder, 5 Tab. 2012.
ISBN 978-3-87525-334-4.

Band 230: Markus Kaupper
Biegen von höhenfesten Stahlblechwerk-
stoffen - Umformverhalten und Grenzen
der Biegebarkeit
LFT, 160 Seiten, 57 Bilder, 10 Tab. 2012.
ISBN 978-3-87525-339-9.

Band 231: Thomas Kroiß
Modellbasierte Prozessauslegung für
die Kaltmassivumformung unter
Brücksichtigung der Werkzeug- und
Pressenauffederung
LFT, 169 Seiten, 50 Bilder, 19 Tab. 2012.
ISBN 978-3-87525-341-2.

Band 232: Christian Goth
Analyse und Optimierung der Entwick-
lung und Zuverlässigkeit räumlicher
Schaltungsträger (3D-MID)
FAPS, 176 Seiten, 102 Bilder, 22 Tab. 2012.
ISBN 978-3-87525-340-5.

Band 233: Christian Ziegler
Ganzheitliche Automatisierung
mechatronischer Systeme in der Medizin
am Beispiel Strahlentherapie
FAPS, 170 Seiten, 71 Bilder, 19 Tab. 2012.
ISBN 978-3-87525-342-9.

Band 234: Florian Albert
Automatisiertes Laserstrahllöten
und -reparatlöten elektronischer
Baugruppen
LPT, 127 Seiten, 78 Bilder, 11 Tab. 2012.
ISBN 978-3-87525-344-3.

Band 235: Thomas Stöhr
Analyse und Beschreibung des
mechanischen Werkstoffverhaltens
von presshärtbaren Bor-Manganstählen
LFT, 118 Seiten, 74 Bilder, 18 Tab. 2013.
ISBN 978-3-87525-346-7.

Band 236: Christian Kägeler
Prozessdynamik beim
Laserstrahlschweißen verzinkter
Stahlbleche im Überlappstoß
LPT, 145 Seiten, 80 Bilder, 3 Tab. 2013.
ISBN 978-3-87525-347-4.

Band 237: Andreas Sulzberger
Seriennahe Auslegung der Prozesskette
zur wärmeunterstützten Umformung
von Aluminiumblechwerkstoffen
LFT, 153 Seiten, 87 Bilder, 17 Tab. 2013.
ISBN 978-3-87525-349-8.

Band 238: Simon Opel
Herstellung prozessangepasster
Halbzeuge mit variabler Blechdicke
durch die Anwendung von Verfahren
der Blechmassivumformung
LFT, 165 Seiten, 108 Bilder, 27 Tab. 2013.
ISBN 978-3-87525-350-4.

Band 239: Rajesh Kanawade
In-vivo Monitoring of Epithelium
Vessel and Capillary Density for the
Application of Detection of Clinical Shock
and Early Signs of Cancer Development
LPT, 124 Seiten, 58 Bilder, 15 Tab. 2013.
ISBN 978-3-87525-351-1.

Band 240: Stephan Busse
Entwicklung und Qualifizierung eines
Schneidclinchverfahrens
LFT, 119 Seiten, 86 Bilder, 20 Tab. 2013.
ISBN 978-3-87525-352-8.

Band 241: Karl-Heinz Leitz
Mikro- und Nanostrukturierung mit kurz
und ultrakurz gepulster Laserstrahlung
LPT, 154 Seiten, 71 Bilder, 9 Tab. 2013.
ISBN 978-3-87525-355-9.

Band 242: Markus Michl
Webbasierte Ansätze zur ganzheitlichen
technischen Diagnose
FAPS, 182 Seiten, 62 Bilder, 20 Tab. 2013.
ISBN 978-3-87525-356-6.

Band 243: Vera Sturm
Einfluss von Chargenschwankungen
auf die Verarbeitungsgrenzen von
Stahlwerkstoffen
LFT, 113 Seiten, 58 Bilder, 9 Tab. 2013.
ISBN 978-3-87525-357-3.

Band 244: Christian Neudel
Mikrostrukturelle und mechanisch-
technologische Eigenschaften
widerstandspunktgeschweißter
Aluminium-Stahl-Verbindungen für
den Fahrzeugbau
LFT, 178 Seiten, 171 Bilder, 31 Tab. 2014.
ISBN 978-3-87525-358-0.

Band 245: Anja Neumann
Konzept zur Beherrschung der
Prozessschwankungen im Presswerk
LFT, 162 Seiten, 68 Bilder, 15 Tab. 2014.
ISBN 978-3-87525-360-3.

Band 246: Ulf-Hermann Quentin
Laserbasierte Nanostrukturierung mit
optisch positionierten Mikrolinsen
LPT, 137 Seiten, 89 Bilder, 6 Tab. 2014.
ISBN 978-3-87525-361-0.

Band 247: Erik Lamprecht
Der Einfluss der Fertigungsverfahren
auf die Wirbelstromverluste von
Stator-Einzelzahnblechpaketen für
den Einsatz in Hybrid- und Elektrofahr-
zeugen
FAPS, 148 Seiten, 138 Bilder, 4 Tab. 2014.
ISBN 978-3-87525-362-7.

Band 248: Sebastian Rösler
Wirkmedienbasierte Umformung von
Blechhalbzeugen unter Anwendung
magnetorheologischer Flüssigkeiten als
kombiniertes Wirk- und Dichtmedium
LFT, 148 Seiten, 61 Bilder, 12 Tab. 2014.
ISBN 978-3-87525-363-4.

Band 249: Paul Hippchen

Simulative Prognose der Geometrie
indirekt pressgehärteter Karosseriebau-
teile für die industrielle Anwendung

LFT, 163 Seiten, 89 Bilder, 12 Tab. 2014.

ISBN 978-3-87525-364-1.

Band 250: Martin Zubeil

Versagensprognose bei der Prozess
simulation von Biegeumform- und Falz-
verfahren

LFT, 171 Seiten, 90 Bilder, 5 Tab. 2014.

ISBN 978-3-87525-365-8.

Band 251: Alexander Kühl

Flexible Automatisierung der
Statorenmontage mit Hilfe einer
universellen ambidexteren Kinematik

FAPS, 142 Seiten, 60 Bilder, 26 Tab. 2014.

ISBN 978-3-87525-367-2.

Band 252: Thomas Albrecht

Optimierte Fertigungstechnologien
für Rotoren getriebeintegrierter
PM-Synchronmotoren von
Hybridfahrzeugen

FAPS, 198 Seiten, 130 Bilder, 38 Tab. 2014.

ISBN 978-3-87525-368-9.

Band 253: Florian Risch

Planning and Production Concepts for
Contactless Power Transfer Systems for
Electric Vehicles

FAPS, 185 Seiten, 125 Bilder, 13 Tab. 2014.

ISBN 978-3-87525-369-6.

Band 254: Markus Weigl

Laserstrahlschweißen von Mischverbin-
dungen aus austenitischen und
ferritischen korrosionsbeständigen
Stahlwerkstoffen

LPT, 184 Seiten, 110 Bilder, 6 Tab. 2014.

ISBN 978-3-87525-370-2.

Band 255: Johannes Noneder

Beanspruchungserfassung für die Validie-
rung von FE-Modellen zur Auslegung von
Massivumformwerkzeugen

LFT, 161 Seiten, 65 Bilder, 14 Tab. 2014.

ISBN 978-3-87525-371-9.

Band 256: Andreas Reinhardt

Ressourceneffiziente Prozess- und
Produktionstechnologie für flexible
Schaltungsträger

FAPS, 123 Seiten, 69 Bilder, 19 Tab. 2014.

ISBN 978-3-87525-373-3.

Band 257: Tobias Schmuck

Ein Beitrag zur effizienten Gestaltung
globaler Produktions- und
Logistiknetzwerke mittels Simulation

FAPS, 151 Seiten, 74 Bilder. 2014.

ISBN 978-3-87525-374-0.

Band 258: Bernd Eichenhüller

Untersuchungen der Effekte und
Wechselwirkungen charakteristischer
Einflussgrößen auf das Umformverhalten
bei Mikroumformprozessen

LFT, 127 Seiten, 29 Bilder, 9 Tab. 2014.

ISBN 978-3-87525-375-7.

Band 259: Felix Lütteke

Vielseitiges autonomes Transportsystem
basierend auf Weltmodellerstellung
mittels Datenfusion von Deckenkameras
und Fahrzeugsensoren

FAPS, 152 Seiten, 54 Bilder, 20 Tab. 2014.

ISBN 978-3-87525-376-4.

Band 260: Martin Grüner

Hochdruck-Blechumformung mit
formlos festen Stoffen als Wirkmedium

LFT, 144 Seiten, 66 Bilder, 29 Tab. 2014.

ISBN 978-3-87525-379-5.

Band 261: Christian Brock
Analyse und Regelung des
Laserstrahltiefschweißprozesses durch
Detektion der Metaldampffackelposition
LPT, 126 Seiten, 65 Bilder, 3 Tab. 2015.
ISBN 978-3-87525-380-1.

Band 262: Peter Vatter
Sensitivitätsanalyse des
3-Rollen-Schubbiegens auf Basis der
Finite Elemente Methode
LFT, 145 Seiten, 57 Bilder, 26 Tab. 2015.
ISBN 978-3-87525-381-8.

Band 263: Florian Klämpfl
Planung von Laserbestrahlungen durch
simulationsbasierte Optimierung
LPT, 169 Seiten, 78 Bilder, 32 Tab. 2015.
ISBN 978-3-87525-384-9.

Band 264: Matthias Domke
Transiente physikalische Mechanismen
bei der Laserablation von dünnen
Metallschichten
LPT, 133 Seiten, 43 Bilder, 3 Tab. 2015.
ISBN 978-3-87525-385-6.

Band 265: Johannes Götz
Community-basierte Optimierung des An-
lagenengineerings
FAPS, 177 Seiten, 80 Bilder, 30 Tab. 2015.
ISBN 978-3-87525-386-3.

Band 266: Hung Nguyen
Qualifizierung des Potentials von
Verfestigungseffekten zur Erweiterung
des Umformvermögens aushärtbarer
Aluminiumlegierungen
LFT, 137 Seiten, 57 Bilder, 16 Tab. 2015.
ISBN 978-3-87525-387-0.

Band 267: Andreas Kuppert
Erweiterung und Verbesserung von Ver-
suchs- und Auswertetechniken für die Be-
stimmung von Grenzformänderungskur-
ven
LFT, 138 Seiten, 82 Bilder, 2 Tab. 2015.
ISBN 978-3-87525-388-7.

Band 268: Kathleen Klaus
Erstellung eines Werkstofforientierten
Fertigungsprozessfensters zur Steigerung
des Formgebungsvermögens von Alumi-
niumlegierungen unter Anwendung einer
zwischen geschalteten Wärmebehandlung
LFT, 154 Seiten, 70 Bilder, 8 Tab. 2015.
ISBN 978-3-87525-391-7.

Band 269: Thomas Svec
Untersuchungen zur Herstellung von
funktionsoptimierten Bauteilen im
partiellen Presshärtprozess mittels lokal
unterschiedlich temperierter Werkzeuge
LFT, 166 Seiten, 87 Bilder, 15 Tab. 2015.
ISBN 978-3-87525-392-4.

Band 270: Tobias Schrader
Grundlegende Untersuchungen zur
Verschleißcharakterisierung beschichteter
Kaltmassivumformwerkzeuge
LFT, 164 Seiten, 55 Bilder, 11 Tab. 2015.
ISBN 978-3-87525-393-1.

Band 271: Matthäus Brela

Untersuchung von Magnetfeld-Messmethoden zur ganzheitlichen Wertschöpfungsoptimierung und Fehlerdetektion an magnetischen Aktoren

FAPS, 170 Seiten, 97 Bilder, 4 Tab. 2015.

ISBN 978-3-87525-394-8.

Band 272: Michael Wieland

Entwicklung einer Methode zur Prognose adhäsiven Verschleißes an Werkzeugen für das direkte Presshärten

LFT, 156 Seiten, 84 Bilder, 9 Tab. 2015.

ISBN 978-3-87525-395-5.

Band 273: René Schramm

Strukturierte additive Metallisierung durch kaltaktives Atmosphärendruckplasma

FAPS, 136 Seiten, 62 Bilder, 15 Tab. 2015.

ISBN 978-3-87525-396-2.

Band 274: Michael Lechner

Herstellung beanspruchungsangepasster Aluminiumblechhalbzeuge durch eine maßgeschneiderte Variation der Abkühlgeschwindigkeit nach Lösungsglühen

LFT, 136 Seiten, 62 Bilder, 15 Tab. 2015.

ISBN 978-3-87525-397-9.

Band 275: Kolja Andreas

Einfluss der Oberflächenbeschaffenheit auf das Werkzeugeinsatzverhalten beim Kaltfließpressen

LFT, 169 Seiten, 76 Bilder, 4 Tab. 2015.

ISBN 978-3-87525-398-6.

Band 276: Marcus Baum

Laser Consolidation of ITO Nanoparticles for the Generation of Thin Conductive Layers on Transparent Substrates

LPT, 158 Seiten, 75 Bilder, 3 Tab. 2015.

ISBN 978-3-87525-399-3.

Band 277: Thomas Schneider

Umformtechnische Herstellung dünnwandiger Funktionsbauteile aus Feinblech durch Verfahren der Blechmassivumformung

LFT, 188 Seiten, 95 Bilder, 7 Tab. 2015.

ISBN 978-3-87525-401-3.

Band 278: Jochen Merhof

Sematische Modellierung automatisierter Produktionssysteme zur Verbesserung der IT-Integration zwischen Anlagen-Engineering und Steuerungsebene

FAPS, 157 Seiten, 88 Bilder, 8 Tab. 2015.

ISBN 978-3-87525-402-0.

Band 279: Fabian Zöller

Erarbeitung von Grundlagen zur Abbildung des tribologischen Systems in der Umformsimulation

LFT, 126 Seiten, 51 Bilder, 3 Tab. 2016.

ISBN 978-3-87525-403-7.

Band 280: Christian Hezler

Einsatz technologischer Versuche zur Erweiterung der Versagensvorhersage bei Karosseriebauteilen aus höchstfesten Stählen

LFT, 147 Seiten, 63 Bilder, 44 Tab. 2016.

ISBN 978-3-87525-404-4.

Band 281: Jochen Bönig
Integration des Systemverhaltens von
Automobil-Hochvoltleitungen in die
virtuelle Absicherung durch
strukturmechanische Simulation
FAPS, 177 Seiten, 107 Bilder, 17 Tab. 2016.
ISBN 978-3-87525-405-1.

Band 282: Johannes Kohl
Automatisierte Datenerfassung für
diskret ereignisorientierte Simulationen in
der energieflexiblen Fabrik
FAPS, 160 Seiten, 80 Bilder, 27 Tab. 2016.
ISBN 978-3-87525-406-8.

Band 283: Peter Bechtold
Mikroschockwellenumformung mittels
ultrakurzer Laserpulse
LPT, 155 Seiten, 59 Bilder, 10 Tab. 2016.
ISBN 978-3-87525-407-5.

Band 284: Stefan Berger
Laserstrahlschweißen thermoplastischer
Kohlenstofffaserverbundwerkstoffe mit
spezifischem Zusatzdraht
LPT, 118 Seiten, 68 Bilder, 9 Tab. 2016.
ISBN 978-3-87525-408-2.

Band 285: Martin Bornschlegl
Methods-Energy Measurement - Eine
Methode zur Energieplanung für
Fügeverfahren im Karosseriebau
FAPS, 136 Seiten, 72 Bilder, 46 Tab. 2016.
ISBN 978-3-87525-409-9.

Band 286: Tobias Rackow
Erweiterung des Unternehmenscontrol-
lings um die Dimension Energie
FAPS, 164 Seiten, 82 Bilder, 29 Tab. 2016.
ISBN 978-3-87525-410-5.

Band 287: Johannes Koch
Grundlegende Untersuchungen zur
Herstellung zyklisch-symmetrischer
Bauteile mit Nebenformelementen durch
Blechmassivumformung
LFT, 125 Seiten, 49 Bilder, 17 Tab. 2016.
ISBN 978-3-87525-411-2.

Band 288: Hans Ulrich Vierzigmann
Beitrag zur Untersuchung der
tribologischen Bedingungen in der
Blechmassivumformung - Bereitstellung
von tribologischen Modellversuchen und
Realisierung von Tailored Surfaces
LFT, 174 Seiten, 102 Bilder, 34 Tab. 2016.
ISBN 978-3-87525-412-9.

Band 289: Thomas Senner
Methodik zur virtuellen Absicherung
der formgebenden Operation des
Nasspressprozesses von
Gelege-Mehrschichtverbunden
LFT, 156 Seiten, 96 Bilder, 21 Tab. 2016.
ISBN 978-3-87525-414-3.

Band 290: Sven Kreitlein
Der grundoperationsspezifische
Mindestenergiebedarf als Referenzwert
zur Bewertung der Energieeffizienz in
der Produktion
FAPS, 185 Seiten, 64 Bilder, 30 Tab. 2016.
ISBN 978-3-87525-415-0.

Band 291: Christian Roos
Remote-Laserstrahlschweißen verzinkter
Stahlbleche in Kehlnahtgeometrie
LPT, 123 Seiten, 52 Bilder, 0 Tab. 2016.
ISBN 978-3-87525-416-7.

Band 292: Alexander Kahrmanidis
Thermisch unterstützte Umformung von
Aluminiumblechen
LFT, 165 Seiten, 103 Bilder, 18 Tab. 2016.
ISBN 978-3-87525-417-4.

Band 293: Jan Tremel
Flexible Systems for Permanent
Magnet Assembly and Magnetic Rotor
Measurement / Flexible Systeme zur
Montage von Permanentmagneten und
zur Messung magnetischer Rotoren
FAPS, 152 Seiten, 91 Bilder, 12 Tab. 2016.
ISBN 978-3-87525-419-8.

Band 294: Ioannis Tsoupis
Schädigungs- und Versagensverhalten
hochfester Leichtbauwerkstoffe unter
Biegebeanspruchung
LFT, 176 Seiten, 51 Bilder, 6 Tab. 2017.
ISBN 978-3-87525-420-4.

Band 295: Sven Hildering
Grundlegende Untersuchungen zum
Prozessverhalten von Silizium als
Werkzeugwerkstoff für das
Mikroscherschneiden metallischer Folien
LFT, 177 Seiten, 74 Bilder, 17 Tab. 2017.
ISBN 978-3-87525-422-8.

Band 296: Sasia Mareike Hertweck
Zeitliche Pulsformung in der
Lasermikromaterialbearbeitung –
Grundlegende Untersuchungen und
Anwendungen
LPT, 146 Seiten, 67 Bilder, 5 Tab. 2017.
ISBN 978-3-87525-423-5.

Band 297: Paryanto
Mechatronic Simulation Approach for
the Process Planning of Energy-Efficient
Handling Systems
FAPS, 162 Seiten, 86 Bilder, 13 Tab. 2017.
ISBN 978-3-87525-424-2.

Band 298: Peer Stenzel
Großserientaugliche Nadelwickeltechnik
für verteilte Wicklungen im
Anwendungsfall der E-Traktionsantriebe
FAPS, 239 Seiten, 147 Bilder, 20 Tab. 2017.
ISBN 978-3-87525-425-9.

Band 299: Mario Lušić
Ein Vorgehensmodell zur Erstellung
montageführender Werkerinformations-
systeme simultan zum
Produktentstehungsprozess
FAPS, 174 Seiten, 79 Bilder, 22 Tab. 2017.
ISBN 978-3-87525-426-6.

Band 300: Arnd Buschhaus
Hochpräzise adaptive Steuerung und
Regelung robotergeführter Prozesse
FAPS, 202 Seiten, 96 Bilder, 4 Tab. 2017.
ISBN 978-3-87525-427-3.

Band 301: Tobias Laumer
Erzeugung von thermoplastischen
Werkstoffverbunden mittels simultanem,
intensitätsselektivem
Laserstrahlschmelzen
LPT, 140 Seiten, 82 Bilder, 0 Tab. 2017.
ISBN 978-3-87525-428-0.

Band 302: Nora Unger
Untersuchung einer thermisch unterstütz-
ten Fertigungskette zur Herstellung
umgeformter Bauteile aus der härtesten
Aluminiumlegierung EN AW-7020
LFT, 142 Seiten, 53 Bilder, 8 Tab. 2017.
ISBN 978-3-87525-429-7.

Band 303: Tommaso Stellin
Design of Manufacturing Processes for
the Cold Bulk Forming of Small Metal
Components from Metal Strip
LFT, 146 Seiten, 67 Bilder, 7 Tab. 2017.
ISBN 978-3-87525-430-3.

Band 304: Bassim Bachy
Experimental Investigation, Modeling,
Simulation and Optimization of Molded
Interconnect Devices (MID) Based on
Laser Direct Structuring (LDS) / Experi-
mentelle Untersuchung, Modellierung, Si-
mulation und Optimierung von Molded
Interconnect Devices (MID) basierend auf
Laser Direktstrukturierung (LDS)
FAPS, 168 Seiten, 120 Bilder, 26 Tab. 2017.
ISBN 978-3-87525-431-0.

Band 305: Michael Spahr
Automatisierte Kontaktierungsverfahren
für flachleiterbasierte
Pkw-Bordnetzsysteme
FAPS, 197 Seiten, 98 Bilder, 17 Tab. 2017.
ISBN 978-3-87525-432-7.

Band 306: Sebastian Suttner
Charakterisierung und Modellierung
des spannungszustandsabhängigen Werk-
stoffverhaltens der Magnesium-
legierung AZ31B für die numerische
Prozessauslegung
LFT, 150 Seiten, 84 Bilder, 19 Tab. 2017.
ISBN 978-3-87525-433-4.

Band 307: Bhargav Potdar
A reliable methodology to deduce thermo-
mechanical flow behaviour of
hot stamping steels
LFT, 203 Seiten, 98 Bilder, 27 Tab. 2017.
ISBN 978-3-87525-436-5.

Band 308: Maria Löffler
Steuerung von Blechmassivumformpro-
zessen durch maßgeschneiderte
tribologische Systeme
LFT, viii u. 166 Seiten, 90 Bilder, 5 Tab.
2018. ISBN 978-3-96147-133-1.

Band 309: Martin Müller
Untersuchung des kombinierten Trenn-
und Umformprozesses beim Fügen artun-
gleicher Werkstoffe mittels
Schneidclinchverfahren
LFT, xi u. 149 Seiten, 89 Bilder, 6 Tab.
2018. ISBN: 978-3-96147-135-5.

Band 310: Christopher Kästle
Qualifizierung der Kupfer-Drahtbond-
technologie für integrierte Leistungs-
module in harschen Umgebungs-
bedingungen
FAPS, xii u. 167 Seiten, 70 Bilder, 18 Tab.
2018. ISBN 978-3-96147-145-4.

Band 311: Daniel Vipave
Eine Simulationsmethode für das
3-Rollen-Schubbiegen
LFT, xiii u. 121 Seiten, 56 Bilder, 17 Tab.
2018. ISBN 978-3-96147-147-8.

Band 312: Christina Ramer
Arbeitsraumüberwachung und autonome
Bahnplanung für ein sicheres und
flexibles Roboter-Assistenzsystem
in der Fertigung
FAPS, xiv u. 188 Seiten, 57 Bilder, 9 Tab.
2018. ISBN 978-3-96147-153-9.

Band 313: Miriam Rauer

Der Einfluss von Poren auf die
Zuverlässigkeit der Lötverbindungen
von Hochleistungs-Leuchtdioden
FAPS, xii u. 209 Seiten, 108 Bilder, 21 Tab.
2018. ISBN 978-3-96147-157-7.

Band 314: Felix Tenner

Kamerabasierte Untersuchungen der
Schmelze und Gasströmungen beim
Laserstrahlschweißen verzinkter Stahlble-
che
LPT, xxiii u. 184 Seiten, 94 Bilder, 7 Tab.
2018. ISBN 978-3-96147-160-7.

Band 315: Aarief Syed-Khaja

Diffusion Soldering for High-temperature
Packaging of Power Electronics
FAPS, x u. 202 Seiten, 144 Bilder, 32 Tab.
2018. ISBN 978-3-87525-162-1.

Band 316: Adam Schaub

Grundlagenwissenschaftliche Unter-
suchung der kombinierten Prozesskette
aus Umformen und Additive Fertigung
LFT, xi u. 192 Seiten, 72 Bilder, 27 Tab.
2019. ISBN 978-3-96147-166-9.

Band 317: Daniel Gröbel

Herstellung von Nebenformelementen
unterschiedlicher Geometrie an Blechen
mittels Fließpressverfahren der Blechmas-
sivumformung
LFT, x u. 165 Seiten, 96 Bilder, 13 Tab.
2019. ISBN 978-3-96147-168-3.

Band 318: Philipp Hildenbrand

Entwicklung einer Methodik zur Herstel-
lung von Tailored Blanks mit definierten
Halbzeugeigenschaften durch einen
Taumelprozess
LFT, ix u. 153 Seiten, 77 Bilder, 4 Tab. 2019.
ISBN 978-3-96147-174-4.

Band 319: Tobias Konrad

Simulative Auslegung der Spann- und
Fixierkonzepte im Karosserierohbau:
Bewertung der Baugruppenmaßhaltigkeit
unter Berücksichtigung schwankender
Einflussgrößen
LFT, x u. 203 Seiten, 134 Bilder, 32 Tab.
2019. ISBN 978-3-96147-176-8.

Band 320: David Meinel

Architektur applikationsspezifischer
Multi-Physics-Simulationskonfiguratoren
am Beispiel modularer Triebzüge
FAPS, xii u. 166 Seiten, 82 Bilder, 25 Tab.
2019. ISBN 978-3-96147-184-3.

Band 321: Andrea Zimmermann

Grundlegende Untersuchungen zum Ein-
fluss fertigungsbedingter Eigenschaften
auf die Ermüdungsfestigkeit kaltmassiv-
umgeformter Bauteile
LFT, ix u. 160 Seiten, 66 Bilder, 5 Tab.
2019. ISBN 978-3-96147-190-4.

Band 322: Christoph Amann

Simulative Prognose der Geometrie
nassgepresster Karosseriebauteile aus
Gelege-Mehrschichtverbunden
LFT, xvi u. 169 Seiten, 80 Bilder, 13 Tab.
2019. ISBN 978-3-96147-194-2.

Band 323: Jennifer Tenner

Realisierung schmierstofffreier Tiefzieh-
prozesse durch maßgeschneiderte Werk-
zeugoberflächen
LFT, x u. 187 Seiten, 68 Bilder, 13 Tab.
2019. ISBN 978-3-96147-196-6.

Band 324: Susan Zöller
Mapping Individual Subjective Values to Product Design
KTmfk, xi u. 223 Seiten, 81 Bilder, 25 Tab.
2019. ISBN 978-3-96147-202-4.

Band 325: Stefan Lutz
Erarbeitung einer Methodik zur semiempirischen Ermittlung der Umwandlungskinetik durchhärtender Wälzlagerstähle für die Wärmebehandlungssimulation
LFT, xiv u. 189 Seiten, 75 Bilder, 32 Tab.
2019. ISBN 978-3-96147-209-3.

Band 326: Tobias Gnibl
Modellbasierte Prozesskettenabbildung rührreibgeschweißter Aluminiumhalbzeuge zur umformtechnischen Herstellung höchstfester Leichtbau-strukturen
LFT, xii u. 167 Seiten, 68 Bilder, 17 Tab.
2019. ISBN 978-3-96147-217-8.

Band 327: Johannes Bürner
Technisch-wirtschaftliche Optionen zur Lastflexibilisierung durch intelligente elektrische Wärmespeicher
FAPS, xiv u. 233 Seiten, 89 Bilder, 27 Tab.
2019. ISBN 978-3-96147-219-2.

Band 328: Wolfgang Böhm
Verbesserung des Umformverhaltens von mehrlagigen Aluminiumblechwerkstoffen mit ultrafeinkörnigem Gefüge
LFT, ix u. 160 Seiten, 88 Bilder, 14 Tab.
2019. ISBN 978-3-96147-227-7.

Band 329: Stefan Landkammer
Grundsatzuntersuchungen, mathematische Modellierung und Ableitung einer Auslegungsmethodik für Gelenkantriebe nach dem Spinnenbeinprinzip
LFT, xii u. 200 Seiten, 83 Bilder, 13 Tab.
2019. ISBN 978-3-96147-229-1.

Band 330: Stephan Rapp
Pump-Probe-Ellipsometrie zur Messung transienter optischer Materialeigenschaften bei der Ultrakurzpuls-Lasermaterialbearbeitung
LPT, xi u. 143 Seiten, 49 Bilder, 2 Tab.
2019. ISBN 978-3-96147-235-2.

Band 331: Michael Scholz
Intralogistics Execution System mit integrierten autonomen, servicebasierten Transportentitäten
FAPS, xi u. 195 Seiten, 55 Bilder, 11 Tab.
2019. ISBN 978-3-96147-237-6.

Band 332: Eva Bogner
Strategien der Produktindividualisierung in der produzierenden Industrie im Kontext der Digitalisierung
FAPS, ix u. 201 Seiten, 55 Bilder, 28 Tab.
2019. ISBN 978-3-96147-246-8.

Band 333: Daniel Benjamin Krüger
Ein Ansatz zur CAD-integrierten muskuloskelettalen Analyse der Mensch-Maschine-Interaktion
KTmfk, x u. 217 Seiten, 102 Bilder, 7 Tab.
2019. ISBN 978-3-96147-250-5.

Band 334: Thomas Kuhn

Qualität und Zuverlässigkeit laserdirekt-
strukturierter mechatronisch integrierter
Baugruppen (LDS-MID)

FAPS, ix u. 152 Seiten, 69 Bilder, 12 Tab.
2019. ISBN: 978-3-96147-252-9.

Band 335: Hans Fleischmann

Modellbasierte Zustands- und Prozess-
überwachung auf Basis sozio-cyber-physi-
scher Systeme

FAPS, xi u. 214 Seiten, 111 Bilder, 18 Tab.
2019. ISBN: 978-3-96147-256-7.

Band 336: Markus Michalski

Grundlegende Untersuchungen zum
Prozess- und Werkstoffverhalten bei
schwingungsüberlagerter Umformung

LFT, xii u. 197 Seiten, 93 Bilder, 11 Tab.
2019. ISBN: 978-3-96147-270-3.

Band 337: Markus Brandmeier

Ganzheitliches ontologiebasiertes
Wissensmanagement im Umfeld der
industriellen Produktion

FAPS, xi u. 255 Seiten, 77 Bilder, 33 Tab.
2020. ISBN: 978-3-96147-275-8.

Band 338: Stephan Purr

Datenerfassung für die Anwendung
lernender Algorithmen bei der Herstel-
lung von Blechformteilen

LFT, ix u. 165 Seiten, 48 Bilder, 4 Tab.
2020. ISBN: 978-3-96147-281-9.

Band 339: Christoph Kiener

Kaltfließpressen von gerad- und schräg-
verzahnten Zahnrädern

LFT, viii u. 151 Seiten, 81 Bilder, 3 Tab.
2020. ISBN 978-3-96147-287-1.

Band 340: Simon Spreng

Numerische, analytische und empirische
Modellierung des Heißcrimpprozesses

FAPS, xix u. 204 Seiten, 91 Bilder, 27 Tab.
2020. ISBN 978-3-96147-293-2.

Band 341: Patrik Schwingenschlögl

Erarbeitung eines Prozessverständnisses
zur Verbesserung der tribologischen
Bedingungen beim Presshärten

LFT, x u. 177 Seiten, 81 Bilder, 8 Tab.
2020. ISBN 978-3-96147-297-0.

Band 342: Emanuela Affronti

Evaluation of failure behaviour
of sheet metals

LFT, ix u. 136 Seiten, 57 Bilder, 20 Tab.
2020. ISBN 978-3-96147-303-8.

Band 343: Julia Degner

Grundlegende Untersuchungen zur
Herstellung hochfester Aluminiumblech-
bauteile in einem kombinierten Umform-
und Abschreckprozess

LFT, x u. 172 Seiten, 61 Bilder, 9 Tab.
2020. ISBN 978-3-96147-307-6.

Band 344: Maximilian Wagner

Automatische Bahnplanung für die Auftei-
lung von Prozessbewegungen in syn-
chrone Werkstück- und Werkzeugbewe-
gungen mittels Multi-Roboter-Systemen

FAPS, xxi u. 181 Seiten, 111 Bilder, 15 Tab.
2020. ISBN 978-3-96147-309-0.

Band 345: Stefan Härter

Qualifizierung des Montageprozesses
hochminiaturisierter elektronischer Bau-
elemente

FAPS, ix u. 194 Seiten, 97 Bilder, 28 Tab.
2020. ISBN 978-3-96147-314-4.

Band 346: Toni Donhauser
Ressourcenorientierte Auftragsregelung in einer hybriden Produktion mittels betriebsbegleitender Simulation
FAPS, xix u. 242 Seiten, 97 Bilder, 17 Tab. 2020. ISBN 978-3-96147-316-8.

Band 347: Philipp Amend
Laserbasiertes Schmelzkleben von Thermoplasten mit Metallen
LPT, xv u. 154 Seiten, 67 Bilder. 2020. ISBN 978-3-96147-326-7.

Band 348: Matthias Ehlert
Simulationsunterstützte funktionale Grenzlagenabsicherung
KTmfk, xvi u. 300 Seiten, 101 Bilder, 73 Tab. 2020. ISBN 978-3-96147-328-1.

Band 349: Thomas Sander
Ein Beitrag zur Charakterisierung und Auslegung des Verbundes von Kunststoffsubstraten mit harten Dünnschichten
KTmfk, xiv u. 178 Seiten, 88 Bilder, 21 Tab. 2020. ISBN 978-3-96147-330-4.

Band 350: Florian Pilz
Fließpressen von Verzahnungselementen an Blechen
LFT, x u. 170 Seiten, 103 Bilder, 4 Tab. 2020. ISBN 978-3-96147-332-8.

Band 351: Sebastian Josef Katona
Evaluation und Aufbereitung von Produktsimulationen mittels abweichungsbehafteter Geometriemodelle
KTmfk, ix u. 147 Seiten, 73 Bilder, 11 Tab. 2020. ISBN 978-3-96147-336-6.

Band 352: Jürgen Herrmann
Kumulatives Walzplattieren. Bewertung der Umformeigenschaften mehrlagiger Blechwerkstoffe der ausscheidungshärtbaren Legierung AA6014
LFT, x u. 157 Seiten, 64 Bilder, 5 Tab. 2020. ISBN 978-3-96147-344-1.

Band 353: Christof Küstner
Assistenzsystem zur Unterstützung der datengetriebenen Produktentwicklung
KTmfk, xii u. 219 Seiten, 63 Bilder, 14 Tab. 2020. ISBN 978-3-96147-348-9.

Band 354: Tobias Gläsel
Prozessketten zum Laserstrahlschweißen von flachleiterbasierten Formspulenwicklungen für automobiler Traktionsantriebe
FAPS, xiv u. 206 Seiten, 89 Bilder, 11 Tab. 2020. ISBN 978-3-96147-356-4.

Band 355: Andreas Meinel
Experimentelle Untersuchung der Auswirkungen von Axialschwingungen auf Reibung und Verschleiß in Zylinderrollenlagern
KTmfk, xii u. 162 Seiten, 56 Bilder, 7 Tab. 2020. ISBN 978-3-96147-358-8.

Band 356: Hannah Riedle
Haptische, generische Modelle weicher anatomischer Strukturen für die chirurgische Simulation
FAPS, xxx u. 179 Seiten, 82 Bilder, 35 Tab. 2020. ISBN 978-3-96147-367-0.

Band 357: Maximilian Landgraf
Leistungselektronik für den Einsatz dielektrischer Elastomere in aktorischen, sensorischen und integrierten sensomotorischen Systemen
FAPS, xxiii u. 166 Seiten, 71 Bilder, 10 Tab. 2020. ISBN 978-3-96147-380-9.

Band 358: Alireza Esfandiyari
Multi-Objective Process Optimization
for Overpressure Reflow Soldering in Elec-
tronics Production
FAPS, xviii u. 175 Seiten, 57 Bilder, 23 Tab.
2020. ISBN 978-3-96147-382-3.

Band 359: Christian Sand
Prozessübergreifende Analyse komplexer
Montageprozessketten mittels
Data Mining
FAPS, XV u. 168 Seiten, 61 Bilder, 12 Tab.
2021. ISBN 978-3-96147-398-4.

Band 360: Ralf Merkl
Closed-Loop Control of a Storage-Sup-
ported Hybrid Compensation System for
Improving the Power Quality in Medium
Voltage Networks
FAPS, xxvii u. 200 Seiten, 102 Bilder, 2 Tab.
2021. ISBN 978-3-96147-402-8.

Band 361: Thomas Reitberger
Additive Fertigung polymerer optischer
Wellenleiter im Aerosol-Jet-Verfahren
FAPS, xix u. 141 Seiten, 65 Bilder, 11 Tab.
2021. ISBN 978-3-96147-400-4.

Band 362: Marius Christian Fechter
Modellierung von Vorentwürfen in der
virtuellen Realität mit natürlicher
Fingerinteraktion
KTmfk, x u. 188 Seiten, 67 Bilder, 19 Tab.
2021. ISBN 978-3-96147-404-2.

Band 363: Franziska Neubauer
Oberflächenmodifizierung und Entwick-
lung einer Auswertemethodik zur Verschleißcharakterisierung im Presshär-
teprozess
LFT, ix u. 177 Seiten, 42 Bilder, 6 Tab.
2021. ISBN 978-3-96147-406-6.

Band 364: Eike Wolfram Schäffer
Web- und wissensbasierter Engineering-
Konfigurator für roboterzentrierte Auto-
matisierungslösungen
FAPS, xxiv u. 195 Seiten, 108 Bilder, 25 Tab.
2021. ISBN 978-3-96147-410-3.

Band 365: Daniel Gross
Untersuchungen zur kohlenstoffdioxidba-
sierten kryogenen Minimalmengen-
schmierung
REP, xii u. 184 Seiten, 56 Bilder, 18 Tab.
2021. ISBN 978-3-96147-412-7.

Band 366: Daniel Junker
Qualifizierung laser-additiv gefertigter
Komponenten für den Einsatz im Werk-
zeugbau der Massivumformung
LFT, vii u. 142 Seiten, 62 Bilder, 5 Tab.
2021. ISBN 978-3-96147-416-5.

Band 367: Tallal Javied
Totally Integrated Ecology Management
for Resource Efficient and Eco-Friendly
Production
FAPS, xv u. 160 Seiten, 60 Bilder, 13 Tab.
2021. ISBN 978-3-96147-418-9.

Band 368: David Marco Hochrein
Wälzlager im Beschleunigungsfeld – Eine
Analysestrategie zur Bestimmung des Rei-
bungs-, Axialschub- und Temperaturver-
haltens von Nadelkränzen –
KTmfk, xiii u. 279 Seiten, 108 Bilder,
39 Tab. 2021. ISBN 978-3-96147-420-2.

Band 369: Daniel Gräf
Funktionalisierung technischer Oberflä-
chen mittels prozessüberwachter aerosol-
basierter Drucktechnologie
FAPS, xxii u. 175 Seiten, 97 Bilder, 6 Tab.
2021. ISBN 978-3-96147-433-2.

Band 370: Andreas Gröschl
Hochfrequent fokusabstandsmodulierte
Konfokalsensoren für die Nanokoordina-
tenmesstechnik
FMT, x u. 144 Seiten, 98 Bilder, 6 Tab.
2021. ISBN 978-3-96147-435-6.

Band 371: Johann Tüchsen
Konzeption, Entwicklung und
Einführung des Assistenzsystems D-DAS
für die Produktentwicklung elektrischer
Motoren
KTmfk, xii u. 178 Seiten, 92 Bilder, 12 Tab.
2021. ISBN 978-3-96147-437-0.

Band 372: Max Marian
Numerische Auslegung von Oberflächen-
mikrostrukturen für geschmierte tribologi-
sche Kontakte
KTmfk, xviii u. 276 Seiten, 85 Bilder, 45
Tab. 2021. ISBN 978-3-96147-439-4.

Band 373: Johannes Strauß
Die akustooptische Strahlformung in der
Lasermaterialbearbeitung
LPT, xvi u. 113 Seiten, 48 Bilder.
2021. ISBN 978-3-96147-441-7.

Band 374: Martin Hohmann
Machine learning and hyper spectral im-
aging: Multi Spectral Endoscopy in the
Gastro Intestinal Tract towards Hyper
Spectral Endoscopy
LPT, x u. 137 Seiten, 62 Bilder, 29 Tab.
2021. ISBN 978-3-96147-445-5.

Band 375: Timo Kordaß
Lasergestütztes Verfahren zur selektiven
Metallisierung von epoxidharzbasierten
Duromeren zur Steigerung der Integrati-
onsdichte für dreidimensionale mecha-
tronische Package-Baugruppen
FAPS, xviii u. 198 Seiten, 92 Bilder,
24 Tab. 2021. ISBN 978-3-96147-443-1.

Band 376: Philipp Kestel
Assistenzsystem für den wissensbasierten
Aufbau konstruktionsbegleitender Finite-
Elemente-Analysen
KTmfk, xviii u. 209 Seiten, 57 Bilder,
17 Tab. 2021. ISBN 978-3-96147-457-8.

Band 377: Martin Lerchen
Messverfahren für die pulverbettbasierte
additive Fertigung zur Sicherstellung der
Konformität mit geometrischen Pro-
duktspezifikationen
FMT, x u. 150 Seiten, 60 Bilder, 9 Tab.
2021. ISBN 978-3-96147-463-9.

Band 378: Michael Schneider
Inline-Prüfung der Permeabilität in
weichmagnetischen Komponenten
FAPS, xxii u. 189 Seiten, 79 Bilder, 14 Tab.
2021. ISBN 978-3-96147-465-3.

Band 379: Tobias Sprügel
Sphärische Detektorflächen als Unterstüt-
zung der Produktentwicklung zur Daten-
analyse im Rahmen des Digital
Engineering
KTmfk, xiii u. 213 Seiten, 84 Bilder, 33 Tab.
2021. ISBN 978-3-96147-475-2.

Band 380: Tom Häfner

Multipulseffekte beim Mikro-Materialabtrag von Stahllegierungen mit Pikosekunden-Laserpulsen
LPT, xxviii u. 159 Seiten, 57 Bilder, 13 Tab.
2021. ISBN 978-3-96147-479-0.

Band 381: Björn Heling

Einsatz und Validierung virtueller Absicherungsmethoden für abweichungs-behaftete Mechanismen im Kontext des Robust Design
KTmfk, xi u. 169 Seiten, 63 Bilder, 27 Tab.
2021. ISBN 978-3-96147-487-5.

Band 382: Tobias Kolb

Laserstrahl-Schmelzen von Metallen mit einer Serienanlage – Prozesscharakterisierung und Erweiterung eines Überwachungssystems
LPT, xv u. 170 Seiten, 128 Bilder, 16 Tab.
2021. ISBN 978-3-96147-491-2.

Band 383: Mario Meinhardt

Widerstandselementschweißen mit gestauchten Hilfsfügeelementen - Umformtechnische Wirkzusammenhänge zur Beeinflussung der Verbindungsfestigkeit
LFT, xii u. 189 Seiten, 87 Bilder, 4 Tab.
2022. ISBN 978-3-96147-473-8.

Band 384: Felix Bauer

Ein Beitrag zur digitalen Auslegung von Fügeprozessen im Karosseriebau mit Fokus auf das Remote-Laserstrahlschweißen unter Einsatz flexibler Spanntechnik
LFT, xi u. 185 Seiten, 74 Bilder, 12 Tab.
2022. ISBN 978-3-96147-498-1.

Band 385: Jochen Zeitler

Konzeption eines rechnergestützten Konstruktionssystems für optomechatronische Baugruppen
FAPS, xix u. 172 Seiten, 88 Bilder, 11 Tab.
2022. ISBN 978-3-96147-499-8.

Band 386: Vincent Mann

Einfluss von Strahloszillation auf das Laserstrahlschweißen hochfester Stähle
LPT, xiii u. 172 Seiten, 103 Bilder, 18 Tab.
2022. ISBN 978-3-96147-503-2.

Band 387: Chen Chen

Skin-equivalent opto-/elastofluidic in-vitro microphysiological vascular models for translational studies of optical biopsies
LPT, xx u. 126 Seiten, 60 Bilder, 10 Tab.
2022. ISBN 978-3-96147-505-6.

Band 388: Stefan Stein

Laser drop on demand joining as bonding method for electronics assembly and packaging with high thermal requirements
LPT, x u. 112 Seiten, 54 Bilder, 10 Tab. 2022.
ISBN 978-3-96147-507-0

Band 389: Nikolaus Urban

Untersuchung des Laserstrahlschmelzens von Neodym-Eisen-Bor zur additiven Herstellung von Permanentmagneten
FAPS, x u. 174 Seiten, 88 Bilder, 18 Tab.
2022. ISBN: 978-3-96147-501-8.

Band 390: Yiting Wu

Großflächige Topographiemessungen mit einem Weißlichtinterferenzmikroskop und einem metrologischen Rasterkraftmikroskop
FMT, xii u. 142 Seiten, 68 Bilder, 11 Tab.
2022. ISBN: 978-3-96147-513-1.

Band 391: Thomas Papke

Untersuchungen zur Umformbarkeit hybrider Bauteile aus Blechgrundkörper und additiv gefertigter Struktur
LFT, xii u. 194 Seiten, 71 Bilder, 16 Tab.
2022. ISBN 978-3-96147-515-5

Band 392: Bastian Zimmermann
Einfluss des Vormaterials auf die mehrstufige Kaltumformung vom Draht
LFT, xi u. 182 Seiten, 36 Bilder, 6 Tab.
2022. ISBN 978-3-96147-519-3.

Band 393: Harald Völkl
Ein simulationsbasierter Ansatz zur Auslegung additiv gefertigter FLM-Faserverbundstrukturen
KTmfk, xx u. 204 Seiten, 95 Bilder, 22 Tab.
2022. ISBN 978-3-96147-523-0.

Band 394: Robert Schulte
Auslegung und Anwendung prozessangepasster Halbzeuge für Verfahren der Blechmassivumformung
LFT, x u. 163 Seiten, 93 Bilder, 5 Tab.
2022. ISBN 978-3-96147-525-4.

Band 395: Philipp Frey
Umformtechnische Strukturierung metallischer Einleger im Folgeverbund für mediendichte Kunststoff-Metall-Hybridbauteile
LFT, ix u. 180 Seiten, 83 Bilder, 7 Tab.
2022. ISBN 978-3-96147-534-6.

Band 396: Thomas Johann Luft
Komplexitätsmanagement in der Produktentwicklung - Holistische Modellierung, Analyse, Visualisierung und Bewertung komplexer Systeme
KTmfk, xiii u. 510 Seiten, 166 Bilder, 16 Tab. 2022 ISBN 978-3-96147-540-7.

Abstract

Complexity is one of the great challenges of our time. Therefore, especially in development, engineers must master the constantly increasing complexity in diverse domains in order to realize product and thus company success. For the appropriate modelling of the many different elements and relations of complex systems, cross-domain, integrative and thus holistic modelling approaches are required, which allow context-related, useful mathematical analysis as well as visualization possibilities in order to enable developers a multi-criteria evaluation of certain action alternatives. However, currently existing approaches are only suitable to a limited extent due to methodological deficits, their high modelling effort, limited analysis and visualization options, or their low support for multi-criteria evaluation.

Therefore, this thesis introduces a concept to support developers in holistic complexity management in product development by providing contextually appropriate methods for integrated modelling, analysis, visualization, and evaluation of complex systems for goal-oriented decision making. For effective and efficient application of the methodology, an IT tool is presented that supports developers in their holistic complexity management tasks.

Komplexität ist eine der großen Herausforderungen unserer Zeit. Daher müssen Ingenieure besonders in der Entwicklung die stetig steigende Komplexität in diversen Domänen beherrschen, um den Produkt- und damit Unternehmenserfolg zu realisieren. Für die zweckmäßige Modellierung der vielen verschiedenen Elemente und Relationen komplexer Systeme sind domänenübergreifende, integrative und damit holistische Modellierungsansätze erforderlich, welche jeweils kontextbezogen nützliche, mathematische Analyse- sowie Visualisierungsmöglichkeiten erlauben um Entwicklern eine multikriterielle Bewertung bestimmter Handlungsalternativen zu ermöglichen. Derzeit bestehende Ansätze eignen sich hierfür jedoch unter anderem aufgrund methodischer Defizite, ihres hohen Modellierungsaufwands, eingeschränkter Analyse- und Visualisierungsmöglichkeiten oder ihrer geringen Unterstützung bei der multikriteriellen Bewertung nur eingeschränkt.

Daher wird in der vorliegenden Arbeit ein Konzept vorgestellt, mit dem Entwickler beim holistischen Komplexitätsmanagement in der Produktentwicklung unterstützt werden, indem kontextbezogen jeweils geeignete Methoden für die integrierte Modellierung, Analyse, Visualisierung und Bewertung komplexer Systeme für eine zielgerichtete Entscheidungsfindung zur Verfügung gestellt werden. Zur effektiven und effizienten Anwendung der Methodik wird ein IT-Werkzeug präsentiert, das Entwickler bei deren Aufgaben im Rahmen des holistischen Komplexitätsmanagements unterstützt.

