

## **Geleitwort**

Das Heinz Nixdorf Institut der Universität Paderborn sowie die Fraunhofer Projektgruppe Entwurfstechnik Mechatronik sind interdisziplinäre Forschungszentren für das Zusammenwirken von Informatik und Technik. Das übergeordnete Ziel des von mir vertretenen Fachgebiets ist die Steigerung der Innovationskraft von Industrieunternehmen des Maschinenbaus und verwandter Branchen. Ein wesentlicher Erfolgsgarant und Schwerpunkt meines Fachgebiets ist das Systems Engineering für multidisziplinäre Systeme.

Im Zeitalter von „Industrie 4.0“ führt der zunehmende Anteil an Informations- und Kommunikationstechnik zu einer Komplexitätssteigerung von mechatronischen Systemen und deren Entwicklungsprozessen. Interdisziplinäres Denken und Handeln sowie ein ganzheitliches Systemverständnis unter den beteiligten Akteuren werden wichtiger denn je. Ein wesentlicher Stellhebel zur Entwicklung komplexer mechatronischer Systeme ist die Wiederverwendung von Lösungswissen. Wir nutzen hierfür sog. Lösungsmuster, die Wissen zur Lösung von abstrakten Funktionen formal repräsentieren. In entsprechenden Ausprägungen führt ihr Einsatz zu einer signifikanten Effizienzsteigerung in der Entwicklung jedweder Artefakte.

Vor diesem Hintergrund hat Herr Anacker ein Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme entwickelt. Schlüsselement ist die konsequente Übertragung des Paradigmas der Lösungsmuster auf die gesamte Produktentstehung. Hierfür werden diverse Lösungsmuster sowie deren Wechselwirkungen untereinander in einer Art Mustersprache abgebildet. Speziell für den fachdisziplinübergreifenden Entwurf liefert Herr Anacker Beschreibungsmittel für Lösungsmuster, eine Methode zur Identifikation von Lösungsmustern sowie eine Methode zur Anwendung von Lösungsmustern im Entwurf von komplexen technischen Systemen.

Mit seiner Dissertation bewegt sich Herr Anacker auf einem hochaktuellen und sehr herausfordernden Gebiet, das z.T. noch grundlegende Impulse benötigt. Er leistet wesentliche integrative Beiträge zu mehreren Forschungsgebieten, die bislang eher isoliert betrachtet wurden: Wissensmanagement in der Produktentstehung, Entwicklungsmethodik des Maschinenbaus von morgen und Systems Engineering. Damit wird auch das hohe Maß der Kohärenz dieser drei Gebiete unterstrichen. Anhand anspruchsvoller Validierungsbeispiele belegt Herr Anacker zudem die Praxisrelevanz und -tauglichkeit des Instrumentariums.



# **Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme**

zur Erlangung des akademischen Grades eines  
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)  
der Fakultät Maschinenbau  
der Universität Paderborn

genehmigte  
DISSERTATION

von  
Dipl.-Ing. Harald Anacker  
aus Paderborn

Tag des Kolloquiums:	22. Mai 2015
Referent:	Prof. Dr.-Ing. Jürgen Gausemeier
Korreferent:	Prof. Dr.-Ing. habil. Ansgar Trächtler



## Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Produktentstehung am Heinz Nixdorf Institut der Universität Paderborn und während meiner Tätigkeit bei der Fraunhofer Projektgruppe Entwurfstechnik Mechatronik. Sie ist das Ergebnis meiner wissenschaftlichen Arbeit im Rahmen von Forschungs- und Industrieprojekten.

Mein herzlicher Dank gilt Herrn Professor Dr.-Ing. Jürgen Gausemeier für die stets fordernde fachliche als auch persönliche Aus- und Weiterbildung in meiner Zeit als wissenschaftlicher Mitarbeiter. Ich danke ihm besonders für das große Vertrauen in meine Arbeit und in meine Person, das er mir auch in herausfordernden Zeiten stets entgegen gebracht hat.

Herrn Professor Dr.-Ing. habil. Ansgar Trächtler danke ich für die Übernahme des Ko-referats sowie die angenehme Zusammenarbeit in den zurückliegenden Projekten und während der andauernden Tätigkeit in der Projektgruppe.

Besonders bedanken möchte ich mich bei Dr.-Ing. Roman Dumitrescu. Er war es, der mich noch während des Studiums in der Zeit als studentische Hilfskraft für den Weg hin zur Promotion begeisterte und daher maßgeblichen Anteil an dieser Weichenstellung in meiner beruflichen Laufbahn hat. Mit großer Dankbarkeit betrachte ich die zurückliegenden sieben gemeinsamen Jahre und seine konsequente Förderung meiner Person.

Ferner danke ich allen Arbeitskollegen am Lehrstuhl und in der Projektgruppe für die tolle Zusammenarbeit und das angenehme Arbeitsklima. Für die stets konstruktiven Diskussionen im Rahmen meiner Dissertation danke ich besonders: Dr.-Ing. Frank Bauer, Anja Czaja, Stefan Dziwok, Peter Iwanek, Arno Kühn, Thomas Schierbaum und Thorsten Westermann. Darüber hinaus danke ich allen Studierenden, die mich bei meiner Arbeit durch ihre Masterarbeiten oder durch ihre studentische Hilfstätigkeit unterstützt haben.

Besonders hervorzuheben ist der Dank an meine Familie: in erster Linie der Dank an meine Eltern, die mir das Studium und die Promotion ermöglicht und mich stets unterstützt haben; aber auch der Dank an meine Schwestern und Schwiegereltern für die fortwährende Motivation.

Mein größtes Dankeschön gilt jedoch meiner Frau Jenni! *Einfach Danke ...*

... für die Kraft, die du mir die ganzen Jahre über gegeben hast,  
... für das Verständnis, das du mir entgegengebracht hast und  
... für die Geduld und das Zurückstecken, wenn die Abende oder die Wochenenden im Zeichen der Dissertation standen.

...



## Liste der veröffentlichten Teilergebnisse

- [DAG10] DUMITRESCU, R.; ANACKER, H.; GAUSEMEIER, J.: Specification of Solution Patterns for the Conceptual Design of Advanced Mechatronic Systems. In: Proceedings of the International Conference on Advances in Mechanical Engineering (ICAME2010), 2.-5. Dezember, Shah Alam, Malaysia, 2010
- [ADD+11] ANACKER, H.; DOROCIAC, R.; DUMITRESCU, R.; GAUSEMEIER, J.: Integrated Tool-Based Approach for the Conceptual Design of Advanced Mechatronic Systems. In: Proceedings of IEEE International Systems Conference (SysCon2011), 3.-6. April, Montreal, Kanada, 2011
- [GSA+11] GAUSEMEIER, J.; SCHÄFER, W.; ANACKER, H.; BAUER, F.; DZIWOK, S.: Einsatz semantischer Technologien im Entwurf mechatronischer Systeme. In: 8. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 294, 19.-20. Mai, Paderborn, 2011
- [DAB+12] DUMITRESCU, R.; ANACKER, H.; BAUER, F.; GAUSEMEIER, J.: Computer Support for the Identification of Solution Patterns for the Conceptual Design of Advanced Mechatronic Systems. In: Proceedings of 11th Biennial Conference on Engineering Systems Design and Analysis (ESDA2012), 2.-4. Juli, Nantes, Frankreich, 2012
- [ADG+12] ANACKER, H.; DUMITRESCU, R.; GAUSEMEIER, J.; DZIWOK, S.; SCHÄFER, W.: Solution Patterns of Software Engineering for the System Design of Advanced Mechatronic Systems. In: Proceedings of the 13th International Workshop on Research and Education in Mechatronics (REM2012), 21.-23. November, Paris, Frankreich, 2012
- [GAC+13] GAUSEMEIER, J.; ANACKER, H.; CZAJA, A.; WASSMANN, H.; DUMITRESCU, R.: Auf dem Weg zu intelligenten technischen Systemen. In: 9. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 310, 18.-19. April, Paderborn, 2013
- [ASD+13] ANACKER, H.; SCHIERBAUM, T.; DUMITRESCU, R.; GAUSEMEIER, J.: Solution Patterns to support the knowledge intensive design process of intelligent technical systems. In: Proceedings of International Conference on Engineering Design (ICED 2013), 19.-22. August, Seoul, Korea, 2013
- [ADG+13] ANACKER, H.; DUMITRESCU, R.; GAUSEMEIER, J.; LOW, C.-Y.: Identification of reusable Controller Strategies for the System Design of Advanced Mechatronic Systems. In: Proceedings of the 2nd International Conference on Advances in Mechanical Engineering (ICAME2013), 28.-29. August, Malacca, Malaysia, 2013
- [ADG13] ANACKER, H.; DUMITRESCU, R.; GAUSEMEIER, J.: Solution Patterns for the integrated Design of Mechatronic Products and related Production Systems. In: Proceedings of the 14th International Workshop on Research and Education in Mechatronics (REM2013), 6.-7. Juni, Wien, Österreich, 2013
- [ADG+14] ANACKER, H.; DUMITRESCU, R.; GAUSEMEIER, J.; IWANEK, P.; SCHIERBAUM, T.: Methodology for the identification of self-optimising potentials for mechatronic systems. In: Proceedings of 2nd Joint Symposium on System-integrated Intelligence: New Challenges for Product and Production Engineering (SysInt), 2.-4. Juli, Bremen, Deutschland, 2014
- [ABG+14] ANACKER, H.; BAUER, F.; GAUSEMEIER, J.; SCHIERBAUM, T.: Systementwurf mit Hilfe von Lösungsmustern aus dem Semantic Web. In: GAUSEMEIER, J.; TRÄCHTLER, A.; SCHÄFER, W. (Hrsg.): Semantische Technologien im Entwurf mechatronischer Systeme – Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten. Carl Hanser Verlag, München, 2014, S.142-164
- [RAD14] RABE, M.; ANACKER, H.; DUMITRESCU, R.: Methodology for the Identification of Solution Patterns within Mechatronic Systems. In: Proceedings of Proceedings of the 15th International Workshop on Research and Education in Mechatronics (REM2014), 9.-11. September, Elgouna, Ägypten, 2014



## Zusammenfassung

Die effiziente Entwicklung fortgeschrittener mechatronischer Systeme ist eine Herausforderung. Interdisziplinäres Denken und Handeln ist aufgrund steigender Komplexität auf Gesamtsystemebene wichtiger denn je. Dieses muss durch die Wiederverwendung etablierten Lösungswissens unterstützt werden. Lösungsmuster bieten das Potential, relevantes Wissen bedarfsgerecht zu dokumentieren und zu vernetzen. Dies gilt für alle Bereiche der Produktentstehung, besonders für den fachdisziplinübergreifenden Systementwurf.

Im Hinblick auf einen *lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* wird ein *Instrumentarium* vorgestellt. Grundlage bildet eine Beschreibung des Paradigmas der Lösungsmuster in der Produktentstehung. Das Instrumentarium enthält Techniken zur Identifizierung, Dokumentation und Anwendung von Lösungsmustern für den Systementwurf. Es werden eine einheitliche Strukturierung für Lösungsmuster sowie ein multidimensionaler Wissensraum erarbeitet. Diese werden anschließend in zwei eigens entwickelte Vorgehensmodelle integriert.

Die Anwendung des Instrumentariums erfolgt anhand von zwei Beispielen. Die methodische Identifizierung von Lösungsmustern wird an einem Tellerseparator gezeigt – einem komplexen mechatronischen System. Anwendungsbeispiel des Vorgehens für einen lösungsmusterbasierten Systementwurf sind kooperierende Deltaroboter.

## Summary

The efficient development of advanced mechatronic systems is still a challenge. Due to the increasing complexity on the general system level, interdisciplinary thinking and acting become more important than ever. This requires the reuse of well-established solution knowledge. In this regard, solution patterns have the potential to document and link relevant knowledge as needed in terms of a proactive allocation. This applies to all areas of product engineering, especially for the multidisciplinary system design.

Therefore an *instrument for a solution pattern-based design of advanced mechatronic systems* was developed. Its basis is an extensive description of the paradigm of solution patterns in product engineering. In addition, the instrument includes techniques for the identification, documentation and use of solution patterns for the system design. Hence a uniform structuration of solution patterns, a multidimensional knowledge space as well as two procedure models were developed and combined.

The use of the instrument is carried out in two examples. The methodical identification of solution patterns is shown on a separator – a complex mechatronic system. An application example of the procedure towards a solution pattern-based system design is a cooperating robot system.



Inhaltsverzeichnis	Seite
1 Einleitung .....	1
1.1 Problematik.....	1
1.2 Zielsetzung .....	3
1.3 Vorgehensweise .....	4
2 Problemanalyse .....	7
2.1 Begriffsdefinitionen und Ausrichtung der Arbeit.....	7
2.2 Mechatronische Systeme .....	8
2.2.1 Grundstruktur mechatronischer Systeme .....	9
2.2.2 Klassen mechatronischer Systeme .....	10
2.2.3 Fortgeschrittene mechatronische Systeme.....	12
2.3 Interdisziplinäre Produktentstehung .....	17
2.3.1 3-Zyklen-Modell nach GAUSEMEIER .....	17
2.3.2 Entwicklung mechatronischer Systeme – VDI2206 .....	19
2.3.3 Systems Engineering.....	22
2.3.4 Model-Based Systems Engineering.....	25
2.4 Wissensmanagement mit Lösungsmustern .....	27
2.4.1 Wissensformen.....	27
2.4.2 SECI-Modell .....	29
2.4.3 Kernprozesse des Wissensmanagements nach PROBST.....	30
2.4.4 Lösungsmuster – ein geeigneter Wissensmanagementansatz? .....	32
2.4.5 Lösungsmuster in der Produktentstehung – Geschichte .....	36
2.5 Problemabgrenzung .....	40
2.6 Anforderungen an die Arbeit.....	43
3 Stand der Technik .....	47
3.1 Überblick – Lösungsmuster in der Produktentstehung .....	47
3.1.1 Universal Design Theory nach GRABOWSKI.....	47
3.1.2 Musterhierarchie nach CLOUTIER .....	49
3.1.3 SE Entwurfsmuster-Metamodell nach PFISTER .....	51
3.1.4 SE Mustersprache nach SIMPSON.....	54
3.2 Lösungswissen für den Systementwurf .....	55
3.2.1 Techniken für die Wissensrepräsentation.....	56
3.2.1.1 CONSENS.....	56

---

3.2.1.2	UML – Unified Modeling Language .....	58
3.2.1.3	SysML – Systems Modeling Language .....	60
3.2.1.4	PrEMISE.....	61
3.2.1.5	Modelica® .....	63
3.2.2	Fachdisziplinübergreifende Lösungsmuster .....	64
3.2.2.1	Systemarchitekturmuster nach CLOUTIER .....	64
3.2.2.2	Lösungsmuster mit SysML nach WEILKINS .....	66
3.2.2.3	Lösungsmuster für selbstoptimierende Systeme nach DUMITRESCU .....	68
3.2.3	Fachdisziplinspezifische Lösungsmuster.....	71
3.2.3.1	Lösungsmuster im Maschinenbau nach SUHM .....	71
3.2.3.2	Entwurfsmuster nach SALUSTRI.....	74
3.2.3.3	Koordinationsmuster der Softwaretechnik nach DZIWOK .....	75
3.2.3.4	Ansätze der Regelungstechnik nach FÖLLINGER.....	77
3.2.3.5	Muster der Regelungstechnik nach SANZ/ZALEWSKI.....	79
3.3	Systematiken für den Einsatz von Lösungsmustern.....	80
3.3.1	Identifizierung von Systemarchitekturmustern nach KALAWSKY.....	81
3.3.2	Musterbasierter Entwurf mechatronischer Systeme nach GAUSEMEIER ET AL.....	82
3.3.3	Musterbasierter Entwurf der selbstoptimierenden Informationsverarbeitung nach DUMITRESCU.....	84
3.4	Bewertung und Handlungsbedarf .....	86
4	Instrumentarium für einen lösungsmusterbasierten Entwurf .....	89
4.1	Das Instrumentarium im Überblick.....	89
4.2	Das Paradigma der Lösungsmuster in der Produktentstehung .....	90
4.2.1	Lösungsmuster – Begriffsklärung .....	91
4.2.2	Klassifikation von Lösungsmustern in der Produktentstehung .....	91
4.2.2.1	Lösungsmuster in der Strategischen Produktplanung .....	92
4.2.2.2	Lösungsmuster in der Produktentwicklung .....	94
4.2.2.3	Lösungsmuster in der Produktionssystementwicklung.....	100
4.2.3	Wirkgefüge der Lösungsmuster in der Produktentstehung....	102
4.3	Lösungswissen für den Systementwurf .....	105
4.3.1	Einheitliche Strukturierung von Lösungsmustern.....	106
4.3.2	Ausgewählte Beispiele von Lösungsmustern .....	108

---

4.3.2.1	Softwareintensives Lösungsmuster „Zusammenarbeit synchronisieren“ .....	108
4.3.2.2	Multidisziplinäres Lösungsmuster „Servoantrieb“ ..	113
4.3.3	Charakterisierung von Lösungsmustern für den Systementwurf.....	116
4.3.3.1	Dimension 1: Art der Wissensrepräsentation.....	116
4.3.3.2	Dimension 2: Spezialisierung des Lösungswissens .....	118
4.3.3.3	Dimension 3: Aggregation des Lösungswissens ...	119
4.3.3.4	Multidimensionaler Wissensraum .....	120
4.3.3.5	Zusammenspiel zwischen Wissensraum und Entwurfsebenen.....	123
4.4	Vorgehensmodell zur Identifizierung von Lösungsmustern für den Systementwurf.....	126
4.4.1	Phase 1: Informationsakquisition.....	127
4.4.2	Phase 2: Systemanalyse .....	128
4.4.3	Phase 3: Interaktive Analyse .....	129
4.4.4	Phase 4: Dokumentation von Lösungsmustern .....	130
4.5	Vorgehensmodell für einen lösungsmusterbasierten Systementwurf.....	131
4.5.1	Phase 1: Initiale Zielbestimmung.....	132
4.5.2	Phase 2: Suche nach Lösungsmustern .....	133
4.5.3	Phase 3: Konkretisierung der Zielbestimmung .....	134
4.5.4	Phase 4: Suche nach spezialisierten Lösungsmustern .....	135
4.5.5	Phase 5: Lösungsmusterbasierte Systemmodellierung.....	136
5	Anwendung und Bewertung .....	139
5.1	Anwendung des Vorgehensmodells zur Identifizierung von Lösungsmustern für den Systementwurf am Beispiel „Separator“....	139
5.1.1	Phase 1: Informationsakquisition.....	140
5.1.2	Phase 2: Systemanalyse .....	142
5.1.3	Phase 3: Interaktive Analyse .....	147
5.1.4	Phase 4: Dokumentation von Lösungsmustern .....	150
5.2	Anwendung des Vorgehensmodells für einen lösungsmusterbasierten Systementwurf am Beispiel „Kooperierende Deltaroboter“ .....	151
5.2.1	Phase 1: Initiale Zielbestimmung.....	152
5.2.2	Phase 2: Suche nach Lösungsmustern .....	155
5.2.3	Phase 3: Konkretisierung der Zielbestimmung .....	156
5.2.4	Phase 4: Suche nach spezialisierten Lösungsmustern .....	158
5.2.5	Phase 5: Lösungsmusterbasierte Systemmodellierung.....	160
5.3	Bewertung der Arbeit anhand der Anforderungen .....	163

---

6	Zusammenfassung und Ausblick .....	167
7	Abkürzungsverzeichnis .....	171
8	Literaturverzeichnis .....	173

## **Anhang**

A1	Ergänzungen zum Stand der Technik .....	A-1
	A1.1 Beispiele für Lösungsmuster nach SIMPSON .....	A-1
	A1.2 Beispiel für ein Lösungsmuster nach CLOUTIER .....	A-3
	A1.3 Beispiele für Lösungsmuster nach GAUSEMEIER ET AL. ....	A-5
A2	Ergänzungen zu Kapitel 4.2 – Das Paradigma der Lösungsmuster in der PE .....	A-7
	A2.1 Beispiele für Lösungsmuster der strategischen Produktplanung .....	A-7
	A2.2 Beispiele für Lösungsmuster der Produktentwicklung .....	A-9
	A2.3 Beispiele für Lösungsmuster der Produktionssystementwicklung..	A-10
A3	Ergänzungen zu Kapitel 4.3 – Lösungswissen für den Systementwurf ..	A-13

# 1 Einleitung

Die vorliegende Arbeit ist im Rahmen von Grundlagenforschung am Heinz Nixdorf Institut sowie der anwendungsorientierten Forschung in der Fraunhofer Projektgruppe Entwurfstechnik Mechatronik des Fraunhofer-Instituts für Produktionstechnologie IPT entstanden. Kern der Arbeit sind Lösungsmuster für den Systementwurf, deren wissenschaftliche Untersuchung Gegenstand des Verbundprojekts ENTIME – **ENT**wurfstechnik **I**ntelligente **M**echatronik – war. Ziel des Projekts war ein Instrumentarium für den Einsatz semantischer Technologien im Entwurf intelligenter mechatronischer Systeme. Hierbei handelt es sich um Systeme, die ihr Verhalten selbstständig und bestmöglich auf das aktuelle Systemumfeld einstellen können. Im Projekt ENTIME wurde, in Kooperation mit mehreren Industrieunternehmen, eine fortlaufende Unterstützung des Entwicklungsprozesses durch die bedarfsgerechte Bereitstellung von Lösungswissen, z.B. Lösungsmuster, umgesetzt. In der Zeit bei der Fraunhofer Projektgruppe konnten die gewonnenen Erkenntnisse vertieft und neue Handlungsfelder eröffnet werden. Die Anwendung erfolgte im Rahmen des Spitzenclusters it's OWL<sup>1</sup>. Die vorliegende Arbeit ordnet sich im Themenfeld Systems Engineering, speziell im Model-Based Systems Engineering (MBSE) ein und beschreibt ein *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*.

## 1.1 Problematik

Unser heutiges Leben ist mehr denn je geprägt durch den Einsatz technischer Systeme. Ob im häuslichen Umfeld, in der industriellen Produktion, Kommunikation oder Logistik, sie sind als Hilfsmittel für den Menschen nicht mehr wegzudenken. Es sind Erzeugnisse des modernen Maschinenbaus oder verwandter Branchen, wie z.B. der Automobilindustrie oder Medizintechnik. Die Systeme beruhen dabei auf dem synergetischen Zusammenwirken unterschiedlicher Fachdisziplinen wie Maschinenbau, Elektrotechnik sowie Regelungs- und Softwaretechnik. Der Begriff Mechatronik, bzw. mechatronisches System, bringt dies zum Ausdruck [VDI2206].

Aufgrund der voranschreitenden Entwicklung der Informations- und Telekommunikationstechnik steigt der Funktionsumfang mechatronischer Systeme rasant an. Es eröffnet sich die Perspektive von global verteilten Systemen, die über das Internet miteinander verbunden sind und im Verbund agieren. Ferner werden diesen Systemen Fähigkeiten implementiert, die so in der Vergangenheit nur von biologischen Systemen bekannt waren. Sie agieren teilweise autonom und sind in der Lage, flexibel auf sich ändernde Betriebssituationen zu reagieren. Derart *fortgeschrittene mechatronische Systeme* sind adaptiv, robust, vorausschauend und benutzungsfreundlich [GAC+13], [Dum10].

---

<sup>1</sup> Intelligente Technische Systeme OstWestfalenLippe (it's OWL)

Zentrale Herausforderungen in der Entwicklung dieser Systeme sind die verstärkte Interdisziplinarität und die steigende Komplexität des zu entwickelnden Systems selbst. Die Entwicklung kann nicht aus dem Blickwinkel einer einzelnen Fachdisziplin erfolgen. Unumstritten ist die Tatsache, dass die Expertise der involvierten Fachdisziplinen für die Ausarbeitung von Teilsystemen bzw. Komponenten essentiell ist. Grundlage einer erfolgreichen Entwicklung muss jedoch ein Denken und Handeln sein, bei dem das multidisziplinäre Gesamtsystem im Mittelpunkt steht. Andernfalls ist die Gesamtfunktionalität des zu entwickelnden Systems aufgrund von z.T. separat entstandenen Teillösungen nicht, bzw. nur mit langwierigen und kostspieligen Iterationen, zu gewährleisten. Ein Ansatz, der sich diesen Herausforderungen annimmt, ist das sog. Systems Engineering.

Im Vordergrund des Systems Engineerings steht das Systemdenken. Es erstreckt sich über die gesamte Produktentstehung, ausgehend von einer ersten Produktidee bis hin zum Serienanlauf. Ziel ist ein einheitliches und ganzheitliches Systemverständnis bei allen beteiligten Akteuren. Ein wesentliches Handlungsfeld ist die fachdisziplinübergreifende Systemspezifikation während der modellbasierten Entwicklung. Kern ist ein sog. Systemmodell, das eine abstrakte und ganzheitliche Systembetrachtung ermöglicht. Es bildet als zentrales Modell die Plattform zur Kommunikation und Kooperation über die gesamte Entwicklung hinweg. Es wird während des *Entwurfs* in Zusammenarbeit mit allen beteiligten Disziplinen erarbeitet. Sobald detailreiche Spezifikationen von einzelnen Fachdisziplinen erforderlich werden, finden über das Systemmodell notwendige Synchronisationen statt. So werden beispielsweise Änderungen mit disziplinübergreifenden Auswirkungen im Systemmodell festgehalten und entsprechend kommuniziert [INC12], [HWF+12], [Alt12].

Ein weiterer Erfolgsgarant im Hinblick auf die Beherrschbarkeit derart komplexer Systeme liegt in der Wiederverwendung von einst erfolgreich eingesetztem Lösungswissen [AG12]. Ein in der Literatur vielfach beschriebener Ansatz zur Externalisierung von entwicklungsrelevantem Wissen sind sog. *Lösungsmuster*. Ein Lösungsmuster beschreibt ein Problem sowie den Kern der Lösung für dieses Problem [AIS+77]. Im Gegensatz zu einer detaillierten Lösung sind die Beschreibungen in einem Lösungsmuster abstrahiert, so dass dieses auf weitere Problemstellungen adaptiert werden kann. Die Interpretationen und Anwendungsgebiete von Lösungsmustern sind außerordentlich facettenreich [Clo06]. Lösungsmuster besitzen gegenüber weiteren Ansätzen im Kontext des Wissensmanagements wesentliche Vorteile, was die Forderung nach ihrem durchgängigen Einsatz in der Produktentstehung bekräftigt [Ris98]. Voraussetzung ist jedoch die grundlegende Übertragung des Paradigmas der Lösungsmuster auf die gesamte Produktentstehung. Hierzu bedarf es eines einheitlichen Begriffsverständnisses, Beispielmuster aller Bereiche sowie die Beschreibung ihres Zusammenspiels in einem übergeordneten Wirkgefüge.

Ein besonderes Augenmerk muss dabei nicht zuletzt aufgrund der Potentiale des Model-Based Systems Engineering auf dem fachdisziplinübergreifenden Entwurf liegen. Fach-

disziplinspezifische Lösungsmuster existieren zahlreich, z.B. Wirkprinzipien im Maschinenbau oder Wirkmuster in der Softwaretechnik [ADG+09]. Im Hinblick auf die Besonderheiten in der Erstellung eines Systemmodells ist eine alleinige Zusammenführung von existierendem Lösungswissen wenig zielführend. Es bedarf vielmehr neuartiger z.T. multidisziplinärer Lösungsmuster. Die darin enthaltenen Informationen müssen in einer Terminologie spezifiziert sein, die alle beteiligten Entwickler<sup>2</sup> gleichermaßen verstehen. Die adressierten Lösungsmuster müssen dabei, wenn immer möglich, nahtlos mit fachdisziplinspezifischen Lösungsmustern verknüpft werden können. Hierzu bedarf es einer geeigneten Strukturierung, um das Lösungswissen langfristig zu dokumentieren und Entwicklern adäquat zur Verfügung zu stellen. Der richtige Einsatz von Lösungsmustern im Systementwurf steigert die Effizienz signifikant. Hierfür bedarf es jedoch einer geeigneten Systematik für den Einsatz von Lösungsmustern. Es muss geklärt werden, wie Lösungsmuster für den Systementwurf identifiziert werden können und wie mit ihnen ein fortgeschrittenes mechatronisches System entworfen werden kann.

## 1.2 Zielsetzung

Ziel der vorliegenden Arbeit ist ein *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*. Es soll für ein einheitliches Verständnis von Lösungsmustern in der Produktentstehung sorgen und ferner den Systems Engineer zusammen mit einem interdisziplinären Entwicklerteam während den frühen Phasen des Entwurfs unterstützen.

Das Instrumentarium soll sich aus insgesamt vier Bestandteilen zusammensetzen. Erster Bestandteil und zugleich Grundlage für die vorliegende Arbeit ist eine **umfangreiche Beschreibung des Paradigmas der Lösungsmuster in der Produktentstehung**. Anschließend findet bei der Erarbeitung der weiteren drei Bestandteile eine Fokussierung auf den Systementwurf statt. Als **Techniken für eine Dokumentation des Lösungswissens für den Systementwurf** werden eine einheitliche Strukturierung von Lösungsmustern und eine multidimensionale Skalierung erarbeitet. Diese müssen dem fachdisziplinübergreifenden Charakter des Systementwurfs gerecht werden und die Verzahnung in die beteiligten Fachdisziplinen verdeutlichen. Zwei weitere Bestandteile ergeben zusammen eine übergeordnete Systematik für den lösungsmusterbasierten Systementwurf. Einen Schwerpunkt bildet ein **Vorgehensmodell zur Identifizierung von (multidisziplinären) Lösungsmustern** auf Basis etablierter Produkte. Andererseits soll ein weiteres **Vorgehensmodell die Anwendung von Lösungsmustern im Entwurf fortgeschrittener mechatronischer Systeme** beschreiben.

---

<sup>2</sup> Es wird im Folgenden die maskuline Form verwendet und zwar ausschließlich wegen der einfachen Lesbarkeit. Wenn beispielsweise von Entwicklern oder Ingenieuren die Rede ist, sind selbstredend auch Entwicklerinnen und Ingenieurinnen gemeint.

Die Anwendbarkeit der Bestandteile soll an zwei Systemen nachgewiesen werden. So bildet ein bereits realisiertes System den Ausgangspunkt bei der Identifizierung von Lösungsmustern. Der lösungsmusterbasierte Systementwurf hingegen erfolgt anhand einer Neukonzeption.

### 1.3 Vorgehensweise

Zunächst wird in **Kapitel 2** die einleitende Problematik konkretisiert. Hierzu werden zu Beginn das Forschungsfeld der Arbeit abgesteckt und relevante Begriffe eingeführt. Betrachtungsgegenstand der vorliegenden Arbeit sind fortgeschrittene mechatronische Systeme. Nach der eingehenden Analyse ihrer Charakteristika und Leistungsstufen wird die Entwicklung derartiger Systeme beleuchtet. Es werden sowohl methodische Ansätze wie die VDI 2206 „Entwicklungsmethodik für mechatronische Systeme“ als auch Grundlagen und Handlungsfelder des Systems Engineerings sowie Model-Based Systems Engineerings untersucht. Da Lösungsmuster – und ein damit verbundenes Wissensmanagement – ein wesentliches Handlungsfeld der vorliegenden Arbeit sind, werden diese in einem eigenen Abschnitt umfassend erläutert. In einer Problemabgrenzung werden die grundlegenden Herausforderungen der vorliegenden Arbeit zusammengefasst. Die Problemanalyse schließt mit der Ableitung von Anforderungen an das angestrebte Instrumentarium.

Gegenstand von **Kapitel 3** ist eine Analyse des Stands der Technik. Den Startpunkt bildet die Untersuchung von übergeordneten Ansätzen zur Beschreibung des Paradigmas der Lösungsmuster in der Produktentstehung. Die nachfolgenden Abschnitte fokussieren den Systementwurf. Zunächst werden Techniken für eine abstrakte und fachdisziplinübergreifende Modellierung mechatronischer Systeme und somit für die Wissensrepräsentation in den Lösungsmustern untersucht. Anschließend werden fachdisziplinübergreifende Lösungsmuster analysiert. Da diese eng mit der Expertise der einzelnen Fachdisziplinen verbunden sind, werden ferner disziplinspezifische Lösungsmusteransätze diskutiert. Vorgehensmodelle, die zur Systematik für den Einsatz von Lösungsmustern im Systementwurf beitragen, setzen die Analyse des Stands der Technik fort. Den Abschluss bildet eine Gegenüberstellung der Ansätze mit den in Kapitel 2 ermittelten Anforderungen. Es wird ersichtlich, dass dringender Handlungsbedarf nach dem angestrebten Instrumentarium besteht.

**Kapitel 4** bildet den Kern der vorliegenden Arbeit. Es beinhaltet eine ausführliche Beschreibung des *Instrumentariums für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*. Das Kapitel startet mit einem Überblick über die Bestandteile und ihrem Zusammenwirken. Anschließend werden die einzelnen Bestandteile vorgestellt. Den Anfang bildet die Herleitung einer für die gesamte Produktentstehung gemeinsamen Begriffswelt der Lösungsmuster. Diese bildet die Grundlage des Instrumentariums und resultiert in einem Wirkgefüge von *Lösungsmustern in der Produktentstehung*. Im Hinblick auf eine geeignete *Dokumentation*, wird eine einheitliche

Strukturierung von Lösungsmustern für den Systementwurf sowie deren Charakterisierung entlang dreier Dimensionen eingeführt. Die erzielten Resultate fließen in den darauf aufbauenden Einsatz von Lösungsmustern im Systementwurf ein und komplettieren die Vorgehensmodelle zur *Identifizierung* und *Anwendung* zu einer übergeordneten Systematik.

Die Validierung der Arbeit erfolgt in **Kapitel 5**. Anhand von zwei ausgewählten Anwendungsbeispielen wird die durchgängige Anwendung des entwickelten Instrumentariums mit ihren einzelnen Bestandteilen gezeigt. Bei den Beispielen handelt es sich um den Demonstrator eines Forschungsprojekts und um ein Industrieerzeugnis, anhand derer die Erfüllung der aufgestellten Anforderungen validiert werden können. Die Anwendungsbeispiele eignen sich hervorragend, um neben dem hohen wissenschaftlichen Neuheitsgrad, die Industrierelevanz der erzielten Ergebnisse zu unterstreichen.

Eine Zusammenfassung der Arbeit ist Inhalt von **Kapitel 6**. Ferner wird ein Ausblick auf zukünftige Forschungsfelder gegeben. Der **Anhang** umfasst ergänzende Informationen zum Stand der Technik und zum Instrumentarium.



## 2 Problemanalyse

Ziel der Problemanalyse sind Anforderungen an ein *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*. Hierfür werden in Kapitel 2.1 wesentliche Begriffe definiert und die Ausrichtung der Arbeit beschrieben. Anschließend erfolgt in Kapitel 2.2 eine Erläuterung der Architektur und Funktionsweise fortgeschrittener mechatronischer Systeme. Den Schwerpunkt von Kapitel 2.3 bildet die interdisziplinäre Produktentstehung. In diesem Zusammenhang wird auf die Entwicklungsmethodiken mechatronischer Systeme und deren Herausforderungen eingegangen. Kapitel 2.4 analysiert anschließend Ansätze des Wissensmanagements und das Paradigma der Lösungsmuster für die Produktentstehung. Kapitel 2.5 umfasst eine Problemabgrenzung. Das Resultat der Analyse sind die Anforderungen an das Instrumentarium und deren Beschreibung in Kapitel 2.6.

### 2.1 Begriffsdefinitionen und Ausrichtung der Arbeit

Gegenstand dieser Arbeit ist ein *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*. Unter dem Begriff **Instrumentarium** ist ein kohärentes Wirkgefüge zu verstehen, das einzelne Bestandteile zur Lösung einer bestimmten Aufgabe beinhaltet und diese vereint. Solche Bestandteile sind u.a. Begriffsdefinitionen, Charakterisierungen, Abbildungen wechselseitiger Abhängigkeiten sowie Vorgehensmodelle<sup>3</sup>.

Bei der Bewältigung bestimmter Aufgaben oder Problemstellungen greift der Mensch bewusst oder unbewusst auf ihm bekannte Muster zurück [Kas03]. Im Vordergrund dieser Arbeit stehen dabei sog. **Lösungsmuster**, die die Entwickler im Lösungsfindungsprozess während des Entwurfs mechatronischer Systeme unterstützen. Diese Muster repräsentieren relevantes, z.T. fachdisziplinübergreifendes Lösungswissen für ein spezifisches Problem innerhalb eines bestimmten Kontextes [AIS+77], [Bar98], [Has05].

Der Begriff **Entwurf**, bzw. Systementwurf oder Konzipierung, umfasst die Festlegung eines fachdisziplinübergreifenden Lösungskonzepts [VDI2206]. Dieses enthält alle wesentlichen physikalischen und logischen Wirkungsweisen des zukünftigen Produkts bzw. Systems<sup>4</sup>.

---

<sup>3</sup> Ein *Vorgehensmodell* strukturiert die Entwicklungstätigkeiten nach aufgabenspezifischen Gesichtspunkten. Dabei wird in der Regel auf Hilfsmittel verwiesen, wie z.B. Methoden, Richtlinien oder auch Spezifikationstechniken/Modellierungssprachen.

<sup>4</sup> „Ein System ist eine in einem betrachteten Zusammenhang gegebene Anordnung von Elementen, die miteinander in Wechselwirkung stehen. Diese Anordnung wird aufgrund bestimmter Vorgaben von ihrer Umgebung abgegrenzt“ [DIN 19226].

Das Instrumentarium dient der Entwicklung **fortgeschrittener mechatronischer Systeme**. Bei diesen handelt es sich um zunehmend intelligenter werdende technische Systeme, die auf dem synergetischen Zusammenwirken technischer Fachdisziplinen wie Maschinenbau, Elektrik/Elektronik und Informationstechnik sowie nicht technischer Disziplinen wie höhere Mathematik, Biologie oder Kognitionswissenschaft basieren [VDI2206, S. 14], [ADG+09, S. 5].

Zudem beinhaltet das Instrumentarium eine umfangreiche Übertragung des **Paradigmas der Lösungsmuster** auf die Produktentstehung. Durch die Fokussierung auf den fachdisziplinübergreifenden Entwurf schlägt die Arbeit zugleich eine Brücke zwischen etablierten Herangehensweisen aus der **Konstruktionswissenschaft**<sup>5</sup> und dem **Systems Engineering**<sup>6</sup>. Es werden die Stärken beider Seiten herauskristallisiert und durch die Aufbereitung relevanten Lösungswissens in Form von Lösungsmustern miteinander gebündelt.

## 2.2 Mechatronische Systeme

Durch die rasante Entwicklung der Informationstechnik ergeben sich für den modernen Maschinenbau vermehrt Möglichkeiten, erfolgversprechende Produktinnovationen umzusetzen. Der Begriff Mechatronik bringt dies zum Ausdruck. Es ist ein Kunstwort aus Mechanik und Elektronik und beschreibt das fachdisziplinübergreifende Zusammenwirken von Maschinenbau, Elektrotechnik und Informationstechnik [VDI2206], [GF06], [Ise08]. In der Literatur existiert eine Vielzahl von Definitionen. Eine häufig zitierte Beschreibung liefern HARASHIMA, TOMIZUKA und FUKUDA:

*„[Mechatronics is]... the synergetic integration of mechanical engineering with electronic and intelligent computer control in the design and manufacturing of industrial products and process“ [HTF96, S. 1].<sup>7</sup>*

Im Vergleich zu konventionellen Lösungen ist durch das Zusammenwirken unterschiedlicher Disziplinen eine erhebliche Funktionserweiterung technischer Systeme realisierbar. Neuartige Entwicklungen zeigen, dass sich bereits Systeme mit intelligenten bzw. autonomen Eigenschaften umsetzen lassen [ADG+09, S. 29ff.]

---

<sup>5</sup> Die Konstruktionswissenschaft beschäftigt sich mit der Ableitung von Regeln zur Entwicklung technischer Systeme auf Basis der Analyse des Aufbaus technischer Systeme und der Beziehungen zu ihrem Umfeld [PBF+07, S. 10]

<sup>6</sup> Systems Engineering (SE) ist eine Sichtweise bzw. ein Ansatz, in dessen Kern ein ganzheitliches Systemverständnis steht. Es werden Methoden und Prozesse zur Realisierung von komplexen technischen Systemen bereitgestellt [INC12].

<sup>7</sup> Übersetzung nach [VDI2206, S. 14]: „*Mechatronik beschreibt das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung.*“

Vor diesem Hintergrund erfolgt in Kapitel 2.2.1 zunächst eine Beschreibung der Grundstruktur mechatronischer Systeme. Kapitel 2.2.2 liefert anschließend eine Beschreibung der Klassen mechatronischer Systeme nach GAUSEMEIER ET AL. Ferner folgt in Kapitel 2.2.3 eine Charakterisierung fortgeschrittener mechatronischer Systeme.

### 2.2.1 Grundstruktur mechatronischer Systeme

Mechatronische Systeme bestehen in der Regel aus einem **Grundsystem**, **Sensorik**, **Aktorik** und einer **Informationsverarbeitung**. Einfluss auf das System hat sowohl der Mensch über die Mensch-Maschine-Schnittstelle, als auch die Umgebung, in der das System betrieben wird [VDI2206]. Die Grundstruktur mechatronischer Systeme zeigt Bild 2-1.

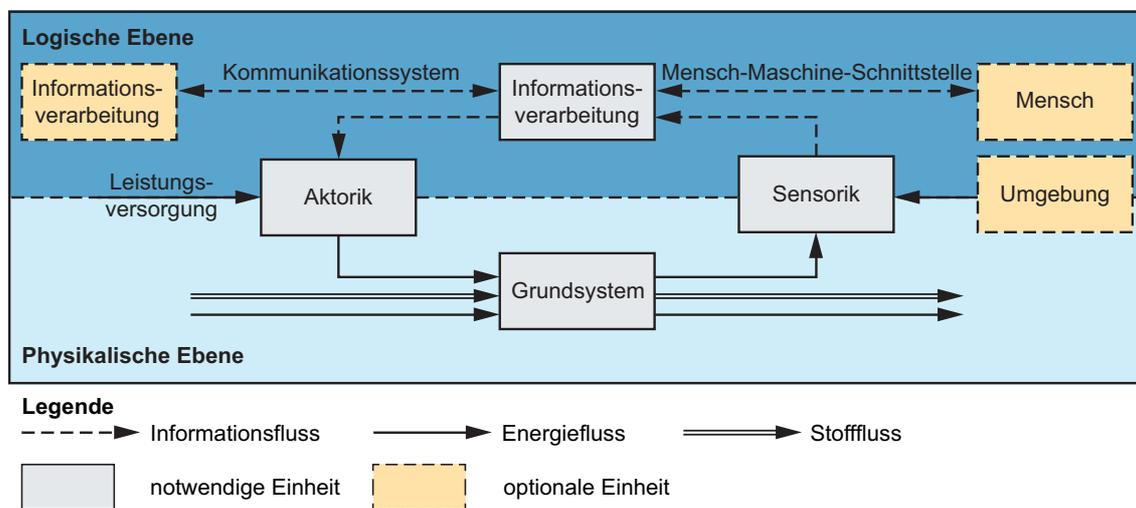


Bild 2-1: Grundstruktur eines mechatronischen Systems [vgl. VDI2206, S. 14]

**Grundsystem:** Das Grundsystem stellt im Normalfall eine mechanische, elektromechanische, hydraulische oder pneumatische Struktur dar. Es ist somit das Kernelement der physikalischen Ebene. Technologische Weiterentwicklungen ermöglichen auch Kombinationen dieser Strukturen.

**Sensorik:** In einem mechatronischen System werden sowohl ausgewählte Zustandsgrößen des Grundsystems als auch äußere Einflüsse der Umgebung von den Sensoren erfasst. Es folgt die Übermittlung der gewonnenen Messwerte an die Informationsverarbeitung.

**Informationsverarbeitung:** Zentrales Element auf der logischen Ebene ist die Informationsverarbeitung. Ihre Aufgabe ist die Ermittlung aller relevanten Einwirkungen, um die Zustandsgrößen des Grundsystems mittels der Aktoren in geeigneter Weise zu beeinflussen.

**Aktorik:** Der Aktor (bzw. Aktuator) hat die Aufgabe, durch eine gewünschte Aktion, unmittelbar an dem Grundsystem Einfluss auf die Zustandsgrößen zu nehmen. Durch den technologischen Fortschritt erweitern sich die Einsatzfelder von Aktoren stetig.

Die unterschiedlichen Systemelemente sind über Flüsse miteinander verbunden [PBF+07]:

- **Stoffflüsse:** Sie beschreiben den Austausch von Stoffen zwischen den Elementen mechatronischer Systeme. Bsp.: Kühlflüssigkeit
- **Energieflüsse:** Neben mechanischer und thermischer Energie beschreiben Energieflüsse auch elektrische Energie und Kenngrößen wie z.B. Kraft oder Strom.
- **Informationsflüsse:** Werden Informationen zwischen den Elementen mechatronischer Systeme ausgetauscht, ist die Verbindung als Signal- oder Informationsfluss zu beschreiben.

### 2.2.2 Klassen mechatronischer Systeme

GAUSEMEIER ET AL. unterteilen mechatronische Systeme in insgesamt drei Klassen. Einige Beispiele sind im Bild 2-2 abgebildet. Die **räumliche Integration von Mechanik und Elektronik** steht bei Systemen der Klasse 1 im Vordergrund. Auf einem kleinen Bauraum soll eine hohe Dichte an mechanischen und elektronischen Funktionsträgern realisiert werden. Wesentliche Erfolgspotentiale liegen in der Miniaturisierung und der Funktionsintegration [GF06], [GAC+13].

Bei mechatronischen Systemen der Klasse 2 liegt der Fokus auf der Verbesserung der Verhaltensweise. Ziel ist es, **Mehrkörpersysteme mit kontrolliertem Bewegungsverhalten** zu entwickeln. Systeme dieser Klasse können durch Sensor-Aktor-Verknüpfungen sowie eine Informationsverarbeitung selbstständig auf Veränderungen in ihrer Umgebung reagieren. Die Hauptaufgabe liegt in der Optimierung der Regelung und in der Automatisierung [GF06].

Die anhaltende Entwicklung der Informations- und Kommunikationstechnik eröffnet faszinierende Perspektiven für mechatronische Systeme, die weit über die bekannten Standards hinausgehen: **Intelligente Mechatronik mit inhärenter Teilintelligenz**. Die Informationstechnik, aber auch nicht technische Disziplinen wie die Kognitionswissenschaft oder die Neurobiologie bringen eine Vielfalt an Methoden, Techniken und Verfahren zur Weiterentwicklung technischer Systeme hervor, die bislang nur von biologischen Systemen bekannt waren. Derartige Systeme besitzen vier zentrale Charaktereigenschaften [Dum10]:

- **Adaptiv:** Sie interagieren mit dem Umfeld und passen sich diesem autonom an. So können sie sich zur Laufzeit, in einem vom Entwickler vorausgedachten Rahmen, weiterentwickeln.

- **Robust:** Sie bewältigen unerwartete und vom Entwickler nicht berücksichtigte Situationen in einem dynamischen Umfeld. Unsicherheiten oder fehlende Informationen können bis zu einem gewissen Grad ausgeglichen werden.
- **Vorausschauend:** Auf der Basis von Erfahrungswissen antizipieren sie die künftigen Wirkungen von Einflüssen; Gefahren werden frühzeitig erkannt und die passenden Strategien zu ihrer Bewältigung ausgewählt.
- **Benutzungsfreundlich:** Sie passen sich dem Benutzerverhalten an und stehen in einer bewussten Interaktion mit dem Benutzer. Dabei bleibt ihr Verhalten für den Benutzer stets nachvollziehbar.

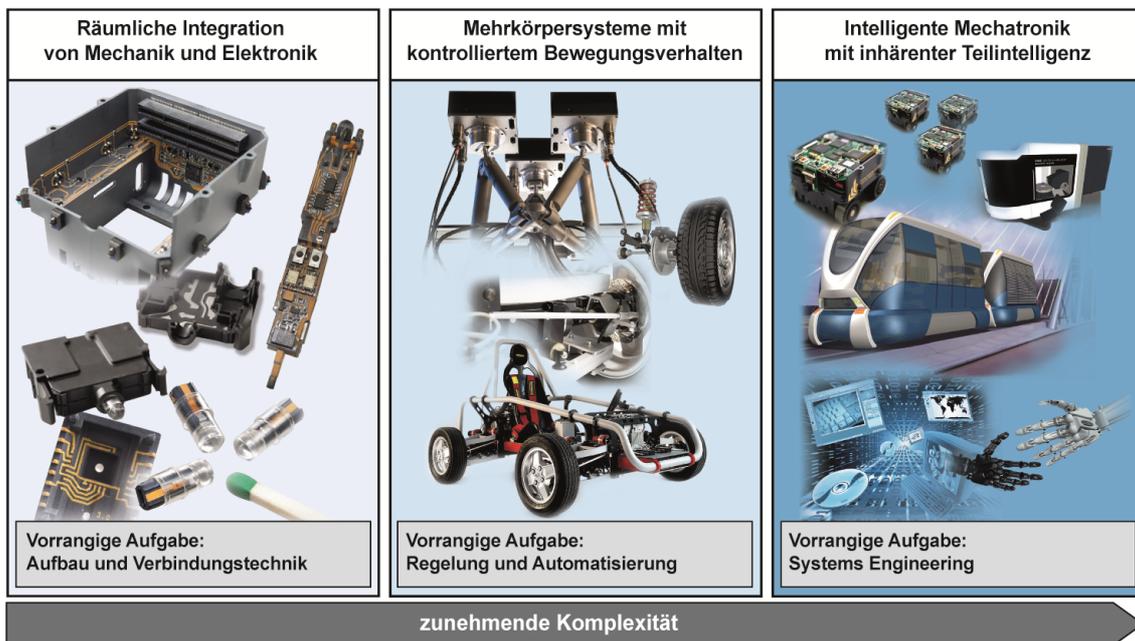


Bild 2-2: Klassen mechatronischer Systeme [vgl. GAC+13]

Aufgrund des sich abzeichnenden Trends hin zu intelligenten technischen Systemen<sup>8</sup>, liegt der Fokus des zu entwickelnden Instrumentariums auf Systemen der zweiten und dritten Klasse. Ziel des Systems Engineerings ist es dabei, diese Systeme als Ganzes zu betrachten, um die Komplexität des Produktes und Entwicklungsgeschehens zu beherrschen. Die im Rahmen dieser Arbeit adressierten Lösungsmuster für solche fortgeschrittenen mechatronischen Systeme gehen mit diesem Grundgedanken einher und beinhalten Lösungswissen aller beteiligten Disziplinen, wie Maschinenbau, Softwaretechnik etc. gleichermaßen. Auf die Besonderheiten des Zusammenwirkens der unterschiedlichen Disziplinen und ihren Wissensbeitrag im Lösungsfindungsprozess wird im folgenden Kapitel eingegangen. Der Bezug zur Referenzarchitektur derartiger Systeme steht dabei im Vordergrund.

<sup>8</sup> Für weitere Informationen z.B. über aktuelle Entwicklungsprojekte sei an dieser Stelle auf den Spitzencluster it's OWL verwiesen: <http://www.its-owl.de/home/>

### 2.2.3 Fortgeschrittene mechatronische Systeme

Im Gegensatz zu konventionellen maschinenbaulichen Erzeugnissen weisen mechatronische Produkte bereits eine erhebliche Funktionssteigerung auf. Dank der Informationsverarbeitung sind sie in der Lage, Einfluss auf das Verhalten des Grundsystems zu nehmen. Es handelt sich dabei um ein rein reaktives System mit starrer Kopplung zwischen Sensorik und Aktorik, welches mit Hilfe fester Vorgaben auf äußere Einflüsse reagieren kann (vgl. Bild 2-3). Diese Vorgaben entstammen der Informationsverarbeitung, die zwischen der Sensorik und Aktorik agiert. Sie ist zugleich ein wesentlicher Ansatzpunkt bei der Weiterentwicklung dieser Systeme. Durch die Integration kognitiver Funktionen<sup>9</sup> sollen derartige Systeme in die Lage versetzt werden, intelligent und flexibel auf veränderte Betriebsbedingungen zu reagieren. Dabei ist es notwendig, die starre Kopplung zwischen Sensorik und Aktorik durch Kognition<sup>10</sup> zu erweitern. Der Ursprung dieser Überlegungen liegt in der Kognitionswissenschaft.

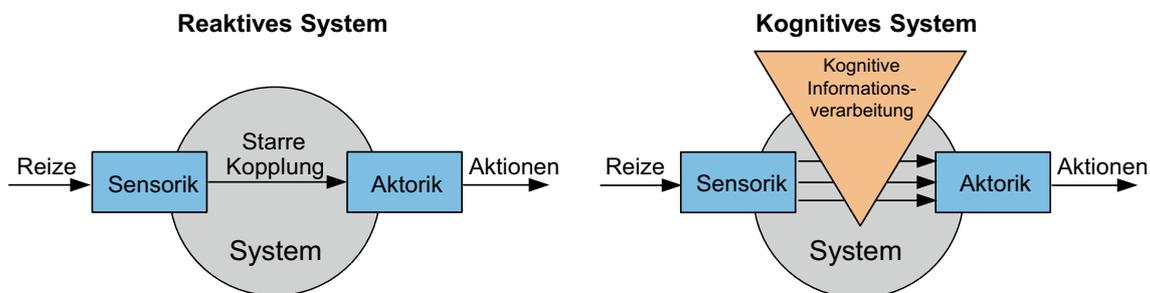


Bild 2-3: Gegenüberstellung eines reaktiven Systems (links) und eines kognitiven Systems (rechts) [Str98, S. 6]

Nach STRUBE besitzen kognitive Systeme folgende Merkmale [GRS03]:

- Aktive Einbindung in die Umgebung und die Fähigkeit, mit ihr Informationen auszutauschen
- Flexible und umgebungsadaptive Handlungssteuerung durch die Repräsentation systemrelevanter Daten der Umwelt
- Lern- und Antizipationsfähigkeit der integrierten Informationsverarbeitung

Durch die Fähigkeiten des Lernens und des Antizipierens sind kognitive Systeme in der Lage, sich an neue Aufgaben und Probleme anzupassen. Lernen bezeichnet dabei den Erwerb neuen Wissens oder die Umstrukturierung von bereits vorhandenem Wissen. Intelligente technische Systeme verfügen aber weiterhin über eine starre Kopplung im

<sup>9</sup> STRUBE versteht unter kognitiven Funktionen u.a.: Wahrnehmen, Erkennen, Enkodieren, Speichern, Erinnern, Denken, Problemlösen, Lernen, Gebrauch der Sprache sowie motorische Steuerung [Str96].

<sup>10</sup> Die moderne Psychologie und Kognitionswissenschaft definiert Kognition als diejenige Fähigkeit, die es Menschen ermöglicht, sich intelligent und flexibel zu verhalten. Nach Strube interveniert Kognition zwischen Reizaufnahme und Verhalten [Str96], [Len02].

Bereich der existentiellen Systemmechanismen. Diese müssen schon aus Gründen der Sicherheit reaktiv und reflexartig ablaufen. Diese Notwendigkeit beruht auf der Überlegung, dass bei komplexen kognitiven Systemen, wie beispielsweise dem Menschen, die starre Kopplung der Sensorik und Aktorik mit der modifizierbaren Kopplung koexistiert. Das aus der Kognitionswissenschaft stammende Dreischichtenmodell für die Verhaltenssteuerung veranschaulicht diese Zusammenhänge (Bild 2-4, links) [Str98]. STRUBE definiert eine Schicht für die nicht kognitive und eine Schicht für die kognitive Regulierung. Die Schnittstelle dieser rein reaktiven und der kognitiven Schicht bildet eine Zwischenebene, welche die assoziative Regulierung beschreibt.

Die unterste Ebene des Schichtenmodells beinhaltet die **nicht kognitive Regulierung**. Aufgrund der starren Kopplung zwischen der Sensorik und der Aktorik findet kein Lernprozess statt. Lediglich in der assoziativen und kognitiven Schicht ist das System lernfähig.

Der Lernprozess in der mittleren Ebene, der **assoziativen Regulierungsschicht**, erfolgt durch klassische oder operante Konditionierung<sup>11</sup>. Die Rede ist hierbei vom assoziativen Lernen. Die klassische Konditionierung geht auf den Psychologen PAVLOV zurück. Es handelt sich um eine Art des Lernens, bei der eine unkonditionierte Reaktion durch einen konditionierten Reiz hervorgerufen wird, der seine Wirkung durch Assoziation mit einem unkonditionierten Reiz erlangte. THORNDIKE<sup>12</sup> zeigte zudem, dass sich diese Reiz-Reaktionsverbindungen durch Wiederholung und positive Bestätigung wesentlich verstärken lassen. Beim operanten Konditionieren nach SKINNER wird die Auftretenswahrscheinlichkeit des Verhaltens durch die darauf folgenden Konsequenzen bestimmt. Wird das Verhalten verstärkt, tritt es zukünftig wiederholt auf. Folgt hingegen eine Bestrafung, wird es zukünftig unterlassen. So können komplexe Verhaltensweisen aus elementaren Lernschritten aufgebaut werden [ZG04].

Die oberste Ebene des Dreischichtenmodells beschäftigt sich mit dem **kognitiven Lernen**. Kognition umfasst alle Arten von Vorgängen, die mit der Aufnahme von Informationen, ihrer Verarbeitung und Speicherung im Gedächtnis sowie ihrer Nutzung und Anwendung verbunden sind. Unter kognitivem Lernen werden alle höheren und bewussten Stufen der Informationsverarbeitung zusammengefasst. Diese beziehen sich auf die Konstruktion von Wissen, die Herausbildung spezifischer Fähigkeiten, die Produktion neuen Wissens, schlussfolgerndes Denken und Urteilen. Kognitives Lernen ermöglicht es, Wissen in neuartigen Situationen anzuwenden und Probleme flexibel zu lösen [See03].

---

<sup>11</sup>Konditionierung ist die Art und Weise, wie Ereignisse, Reize (Stimuli) und Verhalten miteinander assoziiert werden [ZG04]. Es handelt sich dabei um Reiz-Reaktions-Theorien, die „Lernen“ als Kopplung eines bestimmten Umweltreizes mit einer Verhaltensweise definieren [Kel00].

<sup>12</sup>Für das Lernverhalten, welches dem sog. „Lernen durch Versuch und Irrtum“ entspricht, definiert Thorndike das Effekt- sowie das Frequenzgesetz (Gesetz der Übung) [Kel00].

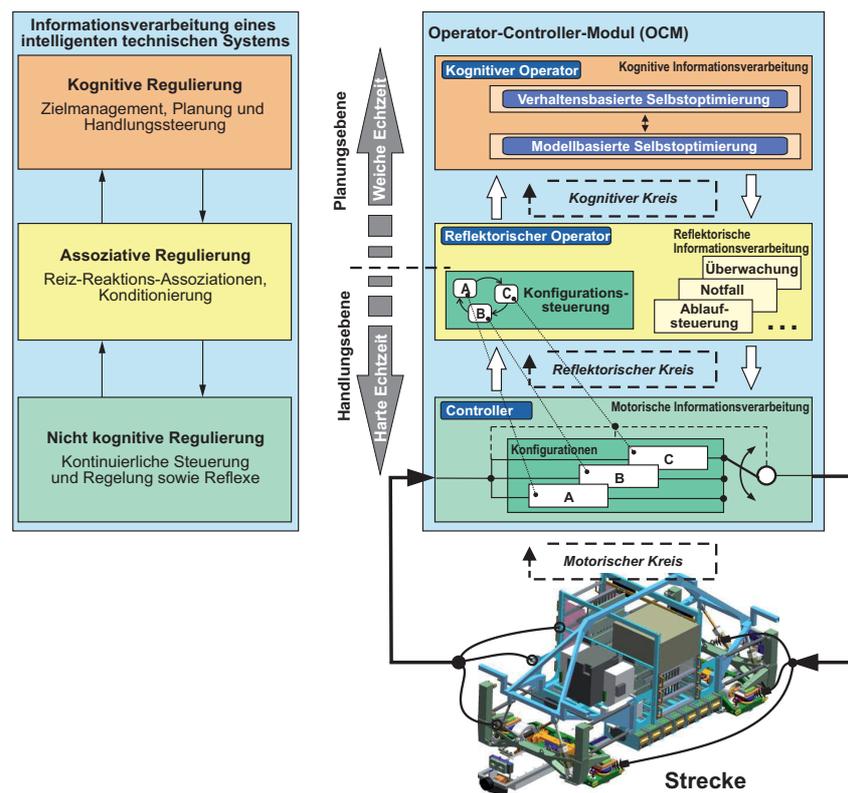


Bild 2-4: Informationsverarbeitung in intelligenten technischen Systemen [GAC+13]

Das Dreischichtenmodell der Verhaltenssteuerung verdeutlicht die hohen Anforderungen an die Umsetzung einer derart komplexen Informationsverarbeitung in technischen Systemen. Insbesondere durch die zunehmende Integration von kognitiven Fähigkeiten, unter Verwendung von Techniken der künstlichen Intelligenz<sup>13</sup>, erhöht sich die Systemkomplexität<sup>14</sup>. Eine Übertragung des Dreischichtenmodells auf technische Systeme liefert NAUMANN [NAU00]. Er entwickelte das Architekturkonzept des Operator-Controller-Moduls (OCM) (Bild 2-4, rechts). Dieses wurde im Sonderforschungsbereich 614 „Selbstoptimierende Systeme des Maschinenbaus“ weiterentwickelt [ADG+09]. Demnach ist ein intelligentes technisches System in vier Einheiten unterteilt: Grundsystem (Strecke), Controller, Reflektorischer Operator und Kognitiver Operator.

### Controller:

Der Controller befindet sich im unteren Bereich des OCM. Über den motorischen Kreis stellt er die Verbindung zwischen dem Grundsystem und der Informationsverarbeitung

<sup>13</sup>Techniken der künstlichen Intelligenz verfolgen das Ziel, basierend auf der Entwicklung von bestimmten Verfahren, kognitive Fähigkeiten in technische Systeme zu integrieren [RN05].

<sup>14</sup>Die Komplexität mechatronischer Systeme oder fortgeschrittener mechatronischer Systeme resultiert aus der steigenden Anzahl von verkoppelten Elementen, die im Rahmen der Ausarbeitung innerhalb verschiedener Fachdisziplinen konkretisiert werden (Heterogenität). Hinzu kommen die Wechselwirkungen zwischen diesen Elementen, die ausschlaggebend für die Komplexität eines Systems sind. [BHL07-ol, S. 8], [VDI2206, S. 4]

her. Die Schnittstellen dieser Verbindung sind die Sensorik und Aktorik. Die Aufgabe des Controllers ist es, das dynamische Verhalten des Grundsystems in gewünschter Weise zu beeinflussen. Dies geschieht durch das Umschalten zwischen unterschiedlichen Konfigurationen. Die Informationsverarbeitung des OCM arbeitet in diesem Bereich unter harten Echtzeitbedingungen<sup>15</sup>.

### **Reflektorischer Operator:**

Der reflektorische Operator (RO) ist über den reflektorischen Regelkreis mit dem Controller verbunden. Er hat die Aufgabe diesen zu überwachen und zu steuern. Über Parameter und Strukturänderungen modifiziert er den Controller, greift aber nicht direkt auf die Aktorik des Systems zu. Vielmehr wechselt der reflektorische Operator zwischen diversen Controllerkonfigurationen (vgl. Bild 2-4, „A“ „B“ „C“). Die Konfigurationen sind dabei eine Kombination aus Reglern und Schaltelementen, die über Signalflüsse miteinander verbunden sind. Darüber hinaus übernimmt der RO Funktionen wie Ablaufsteuerung, Überwachungs- sowie Notfallprozesse. Aufgrund der Verknüpfung zum Controller arbeitet die Informationsverarbeitung im Bereich des RO ebenfalls unter harten Echtzeitbedingungen. Gleichzeitig agiert der RO als Schnittstelle zwischen Controller und kognitivem Operator (KO). Auf der einen Seite nimmt er die Ergebnisse des KO entgegen und leitet sie an den RO weiter. Im Gegenzug übermittelt er die gewonnenen Daten des Controllers an den KO.

### **Kognitiver Operator:**

Aufgabe des kognitiven Operators ist die Umsetzung kognitiver Funktionen zur Realisierung einer Selbstoptimierung. Auf Basis unterschiedlicher Verfahren ist der KO in der Lage, zuvor gewonnenes Wissen zur Verbesserung des Systemverhaltens zu nutzen. Zum Einsatz kommen dabei Planungs- und Lernverfahren, modellorientierte Optimierungsverfahren<sup>16</sup> sowie wissensbasierte Systeme<sup>17</sup>. Aufgrund der zeitlichen Entkopplung der Verfahren zum Verhalten des realen Systems arbeitet die Informationsverarbeitung in diesem Bereich unter weichen Echtzeitbedingungen.

---

<sup>15</sup> Der Begriff Echtzeit stammt aus der Informatik und verlangt, dass das Ergebnis einer Berechnung vor einer fest definierten Zeitschranke vorliegt [Kop97]. Hierbei wird zwischen harter und weicher Echtzeit unterschieden. Harte Echtzeit erfordert das strikte Einhalten der Zeitschranke unter synchroner Durchführung der Berechnung in Bezug auf das Verhalten des realen Systems, z.B.: Airbag. Eine Nichteinhaltung kann schwerwiegende Folgen haben. Weiche Echtzeit setzt ebenfalls die Einhaltung der Zeitschranke voraus. Im Gegensatz zur harten Echtzeit ist die Zeitspanne deutlich größer und die Nichteinhaltung hat keine weiteren Folgen. Es kann zu einem zeitlichen Versatz zwischen der Ergebnisausgabe und dem Systemverhalten kommen, z.B. Audiowiedergabe [KT05].

<sup>16</sup> Modellorientierte Optimierungsverfahren ermöglichen eine vom realen System zeitlich entkoppelte Optimierung, da sie auf Basis vorhandener Modelle agieren. Sie stehen im Gegensatz zu verhaltensorientierten Verfahren, welche unmittelbar auf das aktuelle Systemverhalten reagieren [OHK+02], [HO03].

<sup>17</sup> Der Begriff „wissensbasiertes System“ wird durch die Softwaretechnik geprägt. Durch die Trennung der Wissensbasis und der Wissensverarbeitung ermöglichen derartige Systeme sowohl die Abbildung als auch die Nutzbarkeit vorhandenen Wissens [BK06].

Zusammenfassend lässt sich erkennen, dass der Controller, der reflektorische Operator und der kognitive Operator stark interagieren. Die beteiligten Disziplinen zur Umsetzung einer derart komplexen Informationsverarbeitung sind die Regelungstechnik, die Softwaretechnik, die höhere Mathematik sowie die Techniken der künstlichen Intelligenz.

Fortgeschrittene mechatronische Systeme haben darüber hinaus die Fähigkeit, mit weiteren Systemen zu interagieren (Bild 2-5). Der resultierende Verbund aus einzelnen Systemen muss in der Lage sein, autark auf wechselnde, vom Menschen hervorgerufene Veränderungen reagieren zu können.

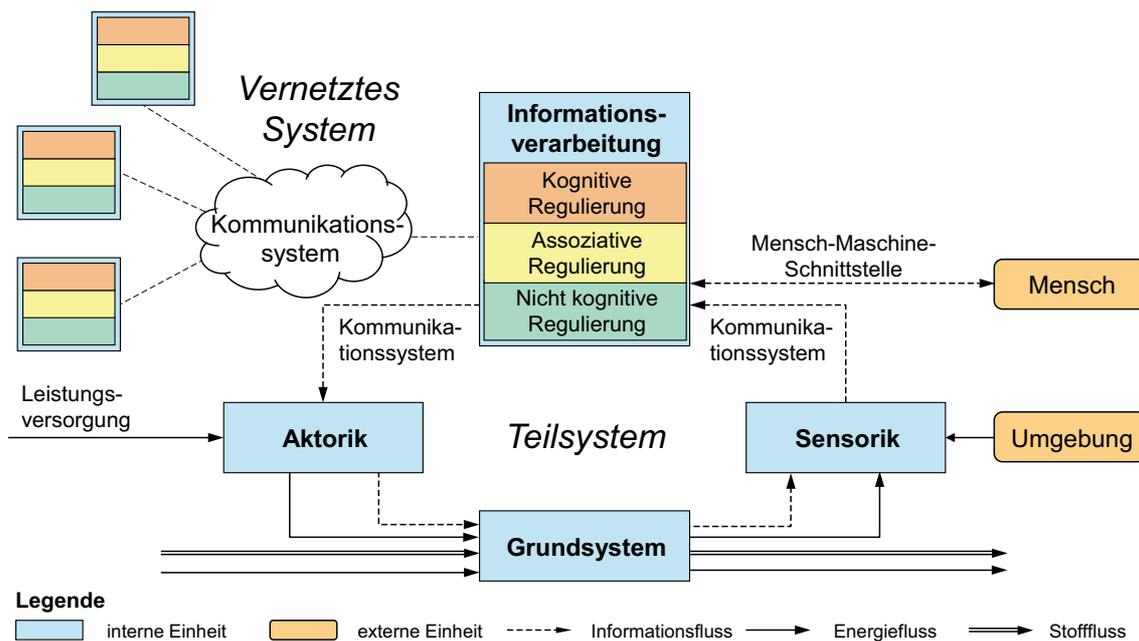


Bild 2-5: Vom intelligenten Teilsystem hin zum vernetzten Cyber-physischen System [GTD13]

Durch die Entwicklung des Internets, wie beispielsweise Cloud Computing, spielt der geographische Standort der einzelnen Systeme im Verbund keine Rolle mehr. Der Trend geht hin zur globalen Vernetzung.

Das angestrebte Instrumentarium hat das Ziel, die Effizienz in der Entwicklung derart komplexer und miteinander interagierender mechatronischer Systeme zu erhöhen. Kern werden z.T. multidisziplinäre Lösungsmuster sein, mit deren Hilfe ein erstes Systemkonzept fachdisziplinübergreifend aufgestellt werden kann. Die Lösungsmuster decken dabei Lösungswissen über das Grundsystem, den Controller, den reflektorischen Operator sowie die Vernetzung zwischen den Systemen ab. Ihr Zusammenspiel in einem übergeordneten Wirkgefüge muss den Besonderheiten der vorgestellten Referenzarchitektur Rechnung tragen und für Transparenz im Hinblick auf fachdisziplinübergreifende Schnittstellen sorgen. Existierende Lösungsmuster für den kognitiven Operator werden berücksichtigt, aber nicht primär adressiert.

## 2.3 Interdisziplinäre Produktentstehung

Im Folgenden wird in Kapitel 2.3.1 das 3-Zyklen-Modell der Produktentstehung nach GAUSEMEIER vorgestellt. Das Modell gliedert die durchzuführenden Tätigkeiten in der Entstehung mechatronischer Systeme. Da ein wesentlicher Schwerpunkt dieser Arbeit auf dem Bereich der Produktentwicklung liegt, wird in Kapitel 2.3.2 die VDI-Richtlinie 2206 beschrieben. Sie bildet nach wie vor den minimalen Konsens über eine Methodik zur Entwicklung mechatronischer Systeme. Anschließend werden in Kapitel 2.3.3 die Handlungsfelder des Systems Engineering analysiert. Komplettiert wird das Kapitel durch eine Untersuchung des Model-Based Systems Engineering in Kapitel 2.3.4. Durch den Vergleich der Konstruktionswissenschaft mit dem Systems Engineering wird das Handlungsfeld des zu erarbeitenden Instrumentariums aufgezeigt.

### 2.3.1 3-Zyklen-Modell nach GAUSEMEIER

Der Produktentstehungsprozess nach GAUSEMEIER erstreckt sich von der Produkt- bzw. Geschäftsidee bis zum Serienanlauf. Nach Bild 2-6 setzt er sich aus folgenden Aufgabenbereichen zusammen: Strategische Produktplanung, Produktentwicklung und Produktionssystementwicklung. Kerngedanke des Modells ist, dass die durchzuführenden Aufgaben im Wechselspiel zueinander stehen und folglich in drei Zyklen durchlaufen werden [GP14, S. 25ff.]. Die einzelnen Zyklen und ihr Zusammenspiel werden im Folgenden näher erörtert.

Von der Geschäftsidee...

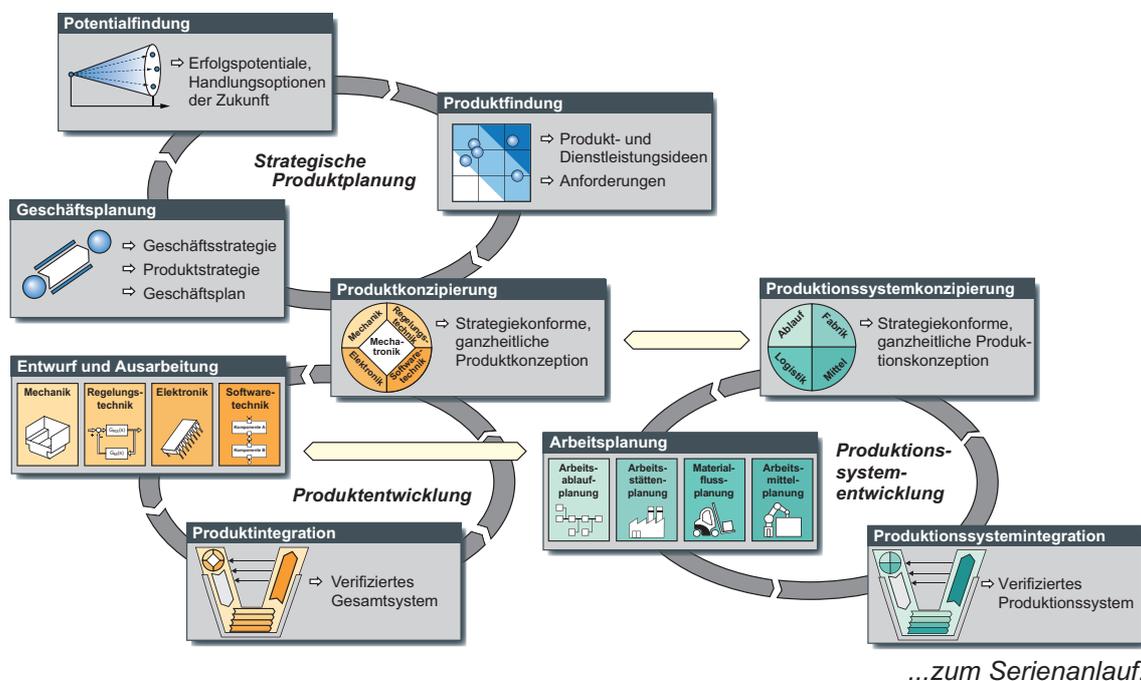


Bild 2-6: 3-Zyklus-Modell der Produktentstehung [GP14, S. 26]

**Erster Zyklus: Strategische Produktplanung**

Die strategische Produktplanung umfasst die Aufgabenbereiche Potentialfindung, Produktfindung, Geschäftsplanung und Produktkonzipierung. Im Rahmen der Potentialfindung sind Erfolgspotentiale der Zukunft sowie die resultierenden Handlungsoptionen zu identifizieren. Die Szenario-Technik, Delphi-Studien, Trend- und Marktanalysen ermöglichen ein methodisches Vorgehen. Unter Berücksichtigung der gefundenen Potentiale kann im Rahmen der Produktfindung die Suche, Auswahl und Erschließung neuer Produkt- und Dienstleistungsideen durchgeführt werden. Grundlegende Hilfsmittel sind das Laterale Denken nach DE BONO, TRIZ oder Technologie-Roadmaps. Die Geschäftsplanung legt zunächst fest, wann, wie und auf welchen Marktsegmenten ein Unternehmen operieren wird. Die resultierende Geschäftsstrategie ist erforderlich, um eine Produktstrategie zu erarbeiten. Aus den Produktstrategien folgt ein Geschäftsplan. Dieser liefert den Nachweis, ob neue Produkte einen attraktiven Return on Investment (ROI) ergeben.

**Zweiter Zyklus: Produktentwicklung**

Die Produktentwicklung setzt sich aus den Aufgabenbereichen Produktkonzipierung, Entwurf und Ausarbeitung sowie Produktintegration zusammen. Die Verbindung der ersten beiden Zyklen erfolgt über die Produktkonzipierung. In diesem Aufgabenbereich steht die Entwicklung prinzipieller Lösungen für das angestrebte Produkt im Vordergrund. Unter Berücksichtigung der funktionalen Anforderungen erfolgen im Anschluss disziplinspezifische Entwürfe und die entsprechenden Ausarbeitungen. Die einzelnen Lösungen werden anschließend im Rahmen der Produktintegration zu einem verifizierten Gesamtsystem zusammengefasst.

**Dritter Zyklus: Produktionssystementwicklung**

Den Ausgangspunkt bildet die Konzipierung des Produktionssystems. Dabei sind die vier Aspekte Arbeitsablaufplanung, Arbeitsmittelplanung, Arbeitsstättenplanung und Produktionslogistik integrativ zu betrachten. Die vier Aspekte sind im Verlauf dieses dritten Zyklus weiter zu konkretisieren.

Produkt- und Produktionssystementwicklung sind parallel und eng aufeinander abgestimmt voranzutreiben. Nur so wird sichergestellt, dass auch alle Möglichkeiten der Gestaltung eines leistungsfähigen und kostengünstigen Erzeugnisses ausgeschöpft werden. Demzufolge verdeutlichen die beiden waagerechten Pfeile in Bild 2-6 die enge Verbindung und den damit verbundenen hohen Abstimmungsbedarf von Produkt- und Produktionssystementwicklung. Dieser beginnt bereits in der Konzipierung und bleibt über den weiteren Verlauf der Konkretisierung in Entwurf und Ausarbeitung bestehen.

### 2.3.2 Entwicklung mechatronischer Systeme – VDI2206

Wie in Kapitel 2.2 beschrieben, beruhen mechatronische Systeme auf dem Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik. Jede Disziplin deckt dabei weitere Spezialisierungsfelder ab: Im Maschinenbau sind dies die Mechanik, Hydraulik, Pneumatik, Mikrotechnik; in der Elektrotechnik sind es die Antriebstechnik, Leistungselektronik oder auch Mikroelektronik; in der Informationstechnik (Regelungstechnik und Softwaretechnik) sind dies die Automatisierungstechnik und die eingebettete Software [Ise08]. Diesem Zusammenspiel, auch der spezialisierten Handlungsfelder, muss eine geeignete Entwicklungsmethodik für mechatronische Systeme gerecht werden.

In diesem Zusammenhang existieren zahlreiche Entwicklungsleitfäden bzw. Methodiken, die in der Regel jedoch auf einzelne Fachdisziplinen ausgerichtet sind. Zu nennen sind z.B. die VDI-Richtlinie 2221 [VDI2221], die Konstruktionslehre nach PAHL/BEITZ [PBF+07], das Y-Modell der Schaltungstechnik [BGH+96] und das V-Modell der Softwareentwicklung [BD93]. Eine Methodik, die in der Wissenschaft und Industrie derzeit den größten Konsens bildet, ist die VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ [VDI2206]. Das grundsätzliche Vorgehen zur fachdisziplinübergreifenden Entwicklung mechatronischer Systeme gliedert sich in drei übergeordnete Bereiche. Diese sind der Problemlösungszyklus auf der Mikroebene, das V-Modell auf der Makroebene sowie die Prozessbausteine für wiederkehrende Arbeitsschritte.

#### Problemlösungszyklus auf der Mikroebene

Der Problemlösungszyklus auf der Mikroebene soll den Produktentwickler bei der Bewältigung vorhersehbarer, aber auch unerwarteter Probleme unterstützen. Er gliedert sich in die Schritte Situationsanalyse/Zielformulierung bzw. Zielübernahme/Situationsübernahme, Analyse/Synthese, Bewertung, Entscheidung sowie Planung des weiteren Vorgehens. Die Prozessplanung lässt sich dabei variabel an die jeweiligen Entwicklungsaufgaben anpassen.

#### V-Modell auf der Makroebene

Das V-Modell zur Beschreibung des grundsätzlichen Vorgehens bei der Entwicklung mechatronischer Systeme stammt ursprünglich aus der Softwaretechnik. Es wurde an die Entwicklungsanforderungen der Mechatronik angepasst (vgl. Bild 2-7). Vor dem Durchlaufen des V-Modells werden Anforderungen an das zu entwickelnde System spezifiziert. Den Abschluss des Vorgehens bildet das entwickelte Produkt. Die einzelnen Bestandteile und Prozessschritte werden nachfolgend detailliert beschrieben, wobei ein besonderer Fokus auf dem Systementwurf liegt.

**Anforderungen:** In Anlehnung an die VDI 2221 gilt es zunächst in der Phase *Planen und Klären der Aufgabe*, die Anforderungen zu spezifizieren. Dies geschieht in Form einer Anforderungsliste, die die Basis für das weitere Vorgehen bildet. Hierbei wird

zwischen funktionalen und nicht funktionale Anforderungen<sup>18</sup> sowie Wünschen unterschieden.

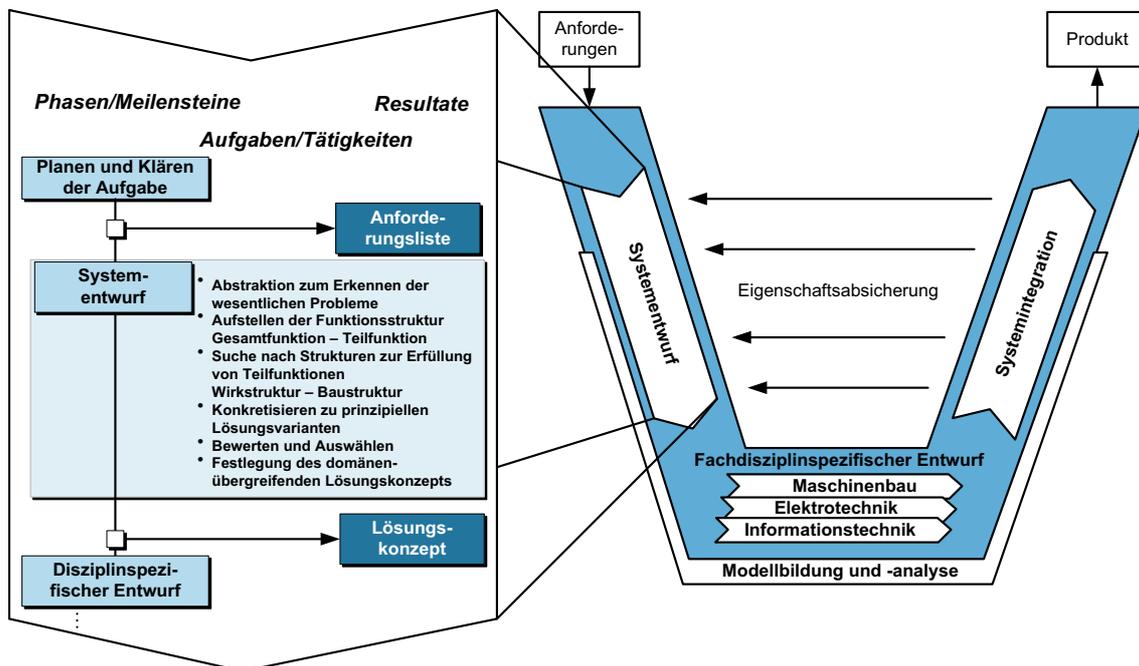


Bild 2-7: Das V-Modell als Makrozyklus (rechts) und die Tätigkeiten für den Prozessbaustein „Systementwurf“ nach [VDI2206, S. 32]

**Systementwurf:** Der Systementwurf beginnt mit der Abstraktion der Aufgabenstellung zum Erkennen der wesentlichen Probleme. Es ist ein initialer Schritt zur Definition der Funktionen, die von dem System erfüllt werden müssen. Nach PAHL/BEITZ beschreibt eine Funktion *den gewollten Zusammenhang zwischen einem Eingang und einem Ausgang eines Systems, mit dem Ziel eine Aufgabe zu erfüllen* [PBF+07, S. 44]. Eine Funktion wird in der Regel durch eine Substantiv-Verb-Kombination beschrieben (z.B. „Drehmoment erzeugen“). Generell ist darauf zu achten, dass die Funktionen möglichst lösungsneutral formuliert sind. Die Darstellung der Funktionen erfolgt durch eine Funktionsstruktur oder eine Funktionshierarchie. Bei einer Funktionshierarchie steht an oberster Stelle die Gesamtfunktion. Diese wird so lange in Teilfunktionen zerlegt, bis passende Lösungsprinzipien<sup>19</sup> gefunden werden [VDI2221, S. 10]. Insbesondere für maschinenbauliche Problemstellungen existieren Konstruktionskataloge, die dem Entwickler die Suche vereinfachen und ihm ein möglichst facettenreiches Spektrum an Lösungen präsentieren sollen [KK98], [Rot01].

<sup>18</sup>Der Ursprung dieser Unterteilung liegt in der Softwaretechnik. Für weitere Informationen sei an dieser Stelle verwiesen auf [BBK+09], [SBD+10]. Funktionale Anforderungen beziehen sich auf das Systemverhalten. Ein Beispiel für nicht funktionale Anforderungen sind z.B. Qualitätsanforderungen.

<sup>19</sup>Die angesprochenen Lösungsprinzipien sind Kernbestandteil der vorliegenden Arbeit und werden daher in Kapiteln 2.4.4 und Kapitel 4.3. im Detail beschrieben.

Aufbauend auf den Suchergebnissen findet die Konzeptsynthese statt. Kernelement ist die sog. Wirkstruktur<sup>20</sup>. Diese bildet Systemelemente sowie deren Merkmale und Beziehungen zueinander ab. Die Wirkstruktur veranschaulicht somit den grundsätzlichen Aufbau und die prinzipielle Wirkungsweise des zu entwickelnden Systems. Zur besseren Übersichtlichkeit und Komplexitätsbeherrschung auf Gesamtsystemebene lassen sich mehrere Systemelemente zu logischen Gruppen zusammenfassen. Im Zuge der Konkretisierung wird die Wirkstruktur in stark vereinfachte Bau- und Komponentenstrukturen überführt. Eine Baustruktur stellt gestaltbehaftete Systemelemente (Synonym: Bauteile) und deren Anordnung im Raum sowie deren logische Aggregation zu Baugruppen dar [PBF+07, S. 56]. Sie ist Ausgangspunkt für den späteren fachdisziplinspezifischen Entwurf im Maschinenbau. Das Pendant zu nicht gestaltbehafteten bzw. informationsverarbeitenden Systemelementen ist die Komponentenstruktur. Diese beschreibt das Zusammenspiel von (Software-)Komponenten<sup>21</sup> in einem Gefüge [Oes05].

Generell werden im Rahmen des Systementwurfs mehrere potentielle Lösungsvarianten erarbeitet. Diese werden unter bestimmten Gesichtspunkten, in der Regel mit einer Nutzwertanalyse bewertet, um die beste Variante herauszukristallisieren. Als Ergebnis des Systementwurfs liegt eine fachdisziplinübergreifende Produktkonzeption vor.

**Fachdisziplinspezifischer Entwurf:** Im Anschluss an die Erstellung des Produktkonzepts erfolgt eine Konkretisierung der Lösungsansätze in den unterschiedlichen Fachdisziplinen. Dabei ist es erforderlich, die Funktionserfüllung durch detaillierte Rechnungen und Auslegungen sicherzustellen.

**Systemintegration:** Im Rahmen der Systemintegration werden die ausgearbeiteten Lösungen zusammengeführt. Das Ergebnis ist der Gesamtentwurf des mechatronischen Systems.

**Eigenschaftsabsicherung:** Die Eigenschaftssicherung dient während der Systemintegration der Ergebnisüberprüfung anhand der Anforderungen.

**Modellbildung und -analyse:** Die Modellbildung und -analyse beinhaltet fortlaufend die rechnergestützte Entwicklung anhand von Modellen<sup>22</sup>.

---

<sup>20</sup>Eine Abgrenzung zu Begriffen wie Systemmodell oder Produktmodell, die in der Literatur in diesem Zusammenhang häufig genannt werden, findet in Kapitel 2.3.4 statt.

<sup>21</sup> SZYPERSKI definiert eine Softwarekomponente wie folgt: „A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties” [Szy02, S. 34].

<sup>22</sup>Nach STACHOWIAK ist: Ein Modell eine beschränkte Repräsentation (Abstraktions-, Verkürzungsmerkmal) von Entitäten und Beziehungen der realen Welt (Abbildungsmerkmal) mit einer eindeutigen Korrespondenz für ein festes Zeitintervall und für einen vorher festgelegten Zweck (pragmatisches Merkmal) [Sta73, S. 131f.].

**Produkt:** Das Resultat des V-Modells ist das Produkt. In der Regel ist es notwendig, das Vorgehensmodell mehrmals zu durchlaufen, um die Konkretisierung und den damit verbundenen Reifegrad des Endproduktes zu erhöhen.

### **Prozessbausteine für wiederkehrende Tätigkeiten**

Für häufig wiederkehrende Tätigkeiten im Rahmen der Entwicklung werden sog. Prozessbausteine formuliert. Im Fokus des Systementwurfs steht dabei die Suche nach vorhandenen Lösungen in den unterschiedlichen Domänen, die zugleich Kern der vorliegenden Arbeit ist. Die Analyse dieses Vorgehensmodells verdeutlicht, dass die Ausarbeitungen stets domänenspezifisch erfolgen. Erst die Überprüfung der Funktionserfüllung gibt Aufschluss darüber, ob eine Kombination der einzelnen Lösungen möglich ist. Diese Vorgehensweise hat zur Folge, dass evtl. Unverträglichkeiten erst am Ende des Systementwurfs festgestellt werden können.

### **VDI 2206 im Hinblick auf intelligente mechatronische Systeme**

Experten sind sich einig, dass eine Kluft zwischen der rasant ansteigenden Produktkomplexität fortgeschrittener mechatronischer Systeme und ihrer Beherrschung durch Entwicklungsmethodiken, wie der VDI 2206, besteht und zukünftig weiter ansteigen wird [GDS13]. Gründe liegen zweifelsohne in der voranschreitenden Entwicklung der Informations- und Telekommunikationstechnologien. Ihr Durchdringen des modernen Maschinenbaus erweitert die Systemgrenzen stetig. Die Rede ist heute von global verteilten cyber-physischen Systemen, die untereinander vernetzt sind und im Verbund agieren (vgl. Kapitel 2.2.3). Die Auswirkungen auf die gesamte Produktentstehung (vgl. Kapitel 2.3.1) sind enorm: Mehr Disziplinen als bisher müssen in die Entwicklung eingebunden werden (z.B. Experten zum Thema Sicherheit in global-verteilten Netzen); es müssen Produktstrategien neu durchdacht und Architekturen ggf. angepasst werden; neuartige Geschäftsmodelle sind zu entwickeln etc. Die Notwendigkeit, das System als Ganzes, zu betrachten, zu beherrschen, zu entwickeln und zu vermarkten ist derzeit größer denn je. Die VDI 2206 stößt dabei an ihre Grenzen. Ein Ansatz, dem in diesem Zusammenhang eine erfolversprechende Rolle zugesprochen wird, ist das sog. Systems Engineering.

### **2.3.3 Systems Engineering**

„Das Ganze ist mehr als die Summe seiner Teile“ [Rop12, S. 25]. Dieser Grundgedanke des Systems Engineering ist bereits mehr als 2300 Jahre alt und geht zurück auf den Philosophen ARISTOTELES. Eine vielfach zitierte Definition von Systems Engineering liefert HITCHINS mit „*Systems Engineering is the art and science of creating whole solutions to complex problems*“ [Hit07, S. 91]. Die Wurzeln liegen in der Systemtheorie, der Modelltheorie und der Kybernetik [Rop12, S. 25ff.]. Im Mittelpunkt des Systems

Engineerings, wie es heutzutage z.B. von der INCOSE<sup>23</sup> vorangetrieben wird, steht das Systemdenken. Ziel ist das Verständnis und die Gestaltung komplexer Systeme [HWF+12, S. 33]. Die INCOSE definiert Systems Engineering wie folgt:

*„Systems Engineering ist ein interdisziplinärer Ansatz und soll die Entwicklung von Systemen methodisch ermöglichen. SE fokussiert ein ganzheitliches und zusammenwirkendes Verständnis der Stakeholder Anforderungen, der Entdeckung von Lösungsmöglichkeiten und der Dokumentation von Anforderungen sowie das Synthetisieren, Verifizieren, Validieren und die Entwicklung von Lösungen. Das gesamte Problem wird während der Konzeptentwicklung bis zur Systementwicklung betrachtet. Das Systems Engineering stellt hierfür geeignete Methoden, Prozesse und Best Practices bereit“ [INC12, S. 7].<sup>24</sup>*

Für HABERFELLNER ET AL. ist Systems Engineering eine „methodische Komponente bei der Problemlösung“ [HWF+12, S. 27]. Es wird betont, dass die Methodik allein weder ausreicht, um ein Problem zu lösen, noch eine dominante Rolle spielen darf. HABERFELLNER ET AL. strukturieren Systems Engineering in die in Bild 2-8 dargestellten übergeordneten Bereiche: Die **SE-Denkweise** und den **Problemlösungsprozess** [HWF+12, S. 28ff.].

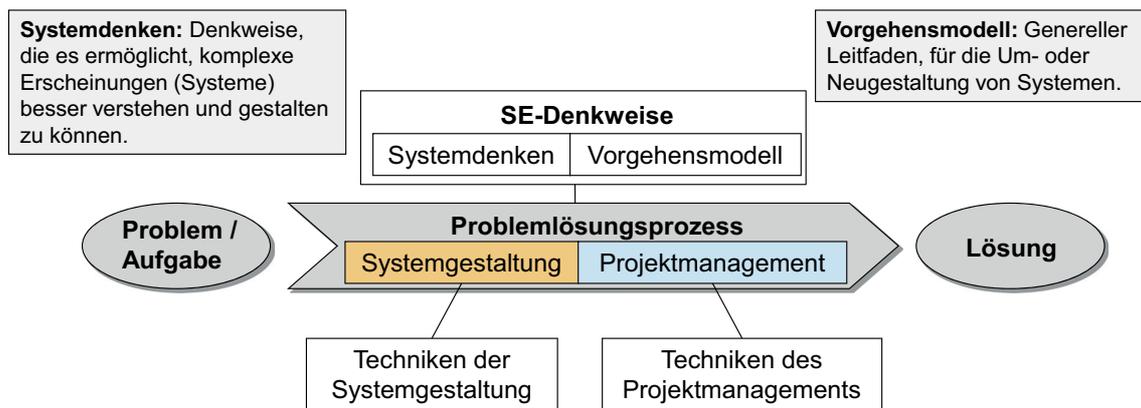


Bild 2-8: Konzept des Systems Engineering [vgl. HWF+12, S. 28]

Die **SE-Denkweise**, die das Systemdenken und das SE-Vorgehensmodell umfasst, bildet den Kern des Systems Engineering. Das *Systemdenken* symbolisiert eine spezielle Denkweise, um komplexe Situationen, Sachverhalte und letztlich Systeme verstehen, strukturieren und gestalten zu können. Im Vordergrund steht die ganzheitliche Systembetrachtung. Dabei wird das System vor dem Hintergrund einer spezifischen Problemstellung oder Situation analysiert und notwendige Zusammenhänge aufgedeckt. Das *Vorgehensmodell* hingegen ist eine Art Leitfaden, um die gesamte Entwicklung in be-

<sup>23</sup>INCOSE: International Council on Systems Engineering; siehe auch: [www.incose.org](http://www.incose.org)

<sup>24</sup> Deutsche Übersetzung des Originals [INC10]

herrschaare Teilprozesse zu zergliedern. Je nach Komplexitat des zu entwickelnden Systems kann das SE-Vorgehensmodell aus einzelnen Komponenten aggregiert werden. Hierbei sind vier Grundgedanken hervorzuheben [HWF+12, S. 57ff.]:

- Top-Down-Vorgehen: Ziel ist es, ausgehend vom Ganzen, das Betrachtungsfeld schrittweise zu verkleinern, Losungen (bzw. Losungsmuster) zu finden und diese abschlieend in der Synthese zusammenzufuhren.
- Varianten: Durch diesen Denkansatz sollen moglichst viele Losungsmoglichkeiten bei der Systemsynthese ins Kalkul gezogen werden. Im Kontext des angestrebten Instrumentariums kann ein Losungsmuster fur den Systementwurf alle notwendigen Informationen ber eine Variante reprasentieren.
- Phasenablauf: Die Systementwicklung wird in eine zeitliche Reihenfolge gegliedert.
- Problemlosungszyklus: Es handelt sich um einen Vorgehensleitfaden, der das Losen von Problemen unterstutzt. Ein Beispiel fur einen derartigen Leitfaden ist das V-Modell der VDI-Richtlinie 2206 fur die Entwicklung mechatronischer Systeme.

Der zweite Bestandteil des SE-Konzepts ist der **Problemlosungsprozess**. Dieser wird unterteilt in die *Systemgestaltung* und das *Projektmanagement*. Letzteres reprasentiert organisatorische Aspekte wie die Verhandlung des Projektauftrags, Festlegung eines Projektteams etc. Die Systemgestaltung umfasst die Tatigkeiten fur den inhaltlichen Aufbau einer Losung. Im Rahmen des fachdisziplinbergreifenden Systementwurfs wird dabei die sog. Systemarchitektur<sup>25</sup> erarbeitet, die nachgelagerte Entwicklungstatigkeiten determiniert. Dies geschieht auf Basis von berlegungen und Entscheidungen aus der strategischen Produktplanung, wie z.B. einer Plattformstrategie.

Gegenstand aktueller Forschungstatigkeiten im Systems Engineering ist die Beherrschung der Komplexitat auf Gesamtsystemebene, insbesondere um in der fruhen Phase die Basis fur eine Konzeptentscheidung zu erarbeiten. In der Literatur werden dem Systems Engineering dabei folgende Aufgabenbereiche zugeordnet: Anforderungsdefinition, -analyse und -management; Systemdesign und -analyse; Verifikation und Validierung; Organisation und Koordination [INC12], [Wei06], [FMS12]. Die Effizienzsteigerung in diesen Aufgabenbereichen durch Nutzung abstrakter Modelle ist aktuell eine der Hauptforderungen aus der Industrie und Kern des sog. Model-Based Systems Engineering [GDS13]. Die steigende Komplexitat fortgeschrittener mechatronischer Systeme erfordert mehr denn je eine ganzheitliche Abbildung und Herangehensweise (vgl. Kapitel 2.2.3).

---

<sup>25</sup>Die Erlauerung der Systemarchitektur und die Einbettung im Kontext Model-Based Systems Engineering geschieht in Kapitel 2.3.4.

### 2.3.4 Model-Based Systems Engineering

Ziel des Model-Based Systems Engineering ist ein fachdisziplinübergreifender Entwurf und die Analyse komplexer Systeme mit Hilfe von Modellen. Zentraler Baustein ist das sog. Systemmodell, das eine **abstrakte und ganzheitliche Systembetrachtung** ermöglicht (vgl. Bild 2-9). Es repräsentiert alle erforderlichen Informationen mit geeigneten Diagrammen, die alle partizipierenden Disziplinen gleichermaßen lesen und interpretieren können. Eine initiale Erstellung des Systemmodells erfolgt im Zusammenspiel aller beteiligten Disziplinen im Rahmen des Systementwurfs bzw. der Konzipierung. Es bildet die Basis für die Kommunikation und Kooperation entlang einer Produktentwicklung von der frühen Phase bis zur Verifikation und Validierung. Während des fachdisziplinspezifischen Entwurfs dient es in erster Linie zur **transparenten Beschreibung der fachdisziplinübergreifend relevanten Informationen**. Das Systemmodell ist erforderlich, um die Komplexität auf Gesamtsystemebene beherrschen und die Entwicklung managen zu können.

#### Abstrakte und ganzheitliche Systembetrachtung

Ein Systemmodell enthält sämtliche Informationen, um ein komplexes technisches System, wie z.B. ein mechatronisches System, ganzheitlich zu beschreiben. Drei Aspekte sind dabei wesentlich: **Anforderungen**, **Systemarchitektur** und **Verhalten** [Alt12, S. 9].

**Anforderungen** beschreiben, was das zu entwickelnde System leisten muss. Sie werden in der Regel in Textform festgehalten [Alt12, S. 10]. Es existieren jedoch auch Ansätze, Anforderungen formal mit Hilfe von Modellen zu beschreiben. Die Ursprünge dieser Herangehensweise liegen in der Informatik. Vielfach zitierte Arbeiten im Kontext Systems Engineering liefern HOOD ET AL. Sie erarbeiten Vorgehensweisen und grafische Modellierungssprachen, um Anforderungen fachdisziplinübergreifend verständlich und durchgängig zu spezifizieren [Hoo07, S. 46ff.] Anforderungen werden unterschieden in Wunsch-/Festforderung sowie funktionale/nicht funktionale Anforderungen<sup>26</sup> [Alt12, S. 10f.], [PBF+07, S. 214ff.]. Festforderungen müssen erfüllt werden, andernfalls ist die erarbeitete Lösungsidee zu verwerfen. Wünsche sind hingegen optional. Es sollte jedoch versucht werden, diese zu erfüllen [PBF+07, S. 214ff.]. Funktionale Anforderungen machen Aussagen über das Systemverhalten, nicht funktionale Anforderungen beschreiben hingegen Qualitätsansprüche z.B. bzgl. Zuverlässigkeit, Effizienz oder Benutzbarkeit [ISO25010].

Definitionen und Synonyme vom Begriff **Systemarchitektur** existieren zahlreich. In Anlehnung an ULRICH und HABERFELLNER ET AL. beschreibt eine Systemarchitektur die Struktur eines Systems in Form von *Funktionen, Systemelementen und deren Beziehungen zueinander*, sowie die *Zusammenhänge zwischen Funktionen und Systemelementen*

---

<sup>26</sup>Vgl. Kapitel 2.3.2

[Ulr95, S. 2], [HWF+12, S. 183]. Systemelemente stellen Softwarekomponenten oder physische Bauteile dar. Systemelemente lassen sich zu Modulen/Baugruppen aggregieren. Die Systemarchitektur ermöglicht eine statische Sicht auf das System und wird mit voranschreitender Entwicklung konkretisiert und kontinuierlich gepflegt. Ein vielfach verwendetes Synonym ist der Begriff Wirkstruktur [PBF+77, S. 255ff.].

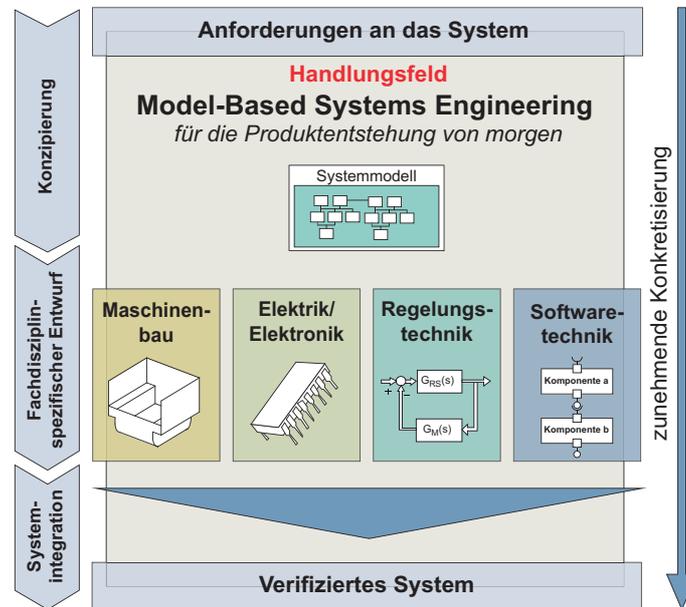


Bild 2-9: Handlungsfelder des Model-Based Systems Engineering

Komplettiert wird das Systemmodell durch die Abbildung des zeitlich veränderbaren **Verhaltens**. Die Abbildungsmöglichkeiten sind durch die unterschiedlichen Aspekte wie „Thermik“, „Dynamik“ etc. sehr vielseitig. In der Regel beinhaltet die Darstellung des Verhaltens mechatronischer Systeme auch die zeitlichen Zusammenhänge im Informationsaustausch zwischen den Systemelementen [Alt12, S. 14ff.].

### Transparente Beschreibung der disziplinübergreifend relevanten Informationen

Entscheidend für den Erfolg oder Misserfolg in der Entwicklung mechatronischer Systeme sind die Systemschnittstellen, ihre eindeutige Beschreibung, ihre kontinuierliche Pflege und die Kommunikation bei Änderungen. Die beteiligten Entwickler sind Experten in ihrer Disziplin, verlieren jedoch häufig die Einbettung ihrer Komponente in das Gesamtsystem aus dem Blickfeld. Die Unterschiede in den Denk- und Begriffswelten der einzelnen Disziplinen führen darüber hinaus häufig zu Missverständnissen bei Abstimmungen. In einem übergeordneten Systemmodell sieht die Industrie und Forschung derzeit das größte Potential, diesen Problemen entgegenzuwirken.

Das Systemmodell und die damit einhergehenden Methoden des MBSE geben den Entwicklern Ausdrucksmittel an die Hand, die jede Disziplin gleichermaßen versteht. Ziel ist eine Plattform, an der alle beteiligten Ingenieure partizipieren und die ihnen die Möglichkeit des Austausches bietet. Entscheidend ist die richtige Abstraktionsebene. Es muss über relevante Sachverhalte diskutiert werden können, ohne sich dabei in (z.B.

disziplinspezifischen) Details zu verlieren. Dies beinhaltet auch die uneindeutigen Bezeichnungen der Systemelemente, der Parameter und der Einheiten. Das Systemmodell wird zu Beginn einer jeden Entwicklung initial mit allen Disziplinen erarbeitet. Es bildet somit die Basis für den detaillierten fachdisziplinspezifischen Entwurf. Alle auf Gesamtsystemebene relevanten Änderungen sind im Systemmodell nachzuhalten. Nur auf diese Weise ist eine Rückverfolgbarkeit und abschließende Verifikation gegenüber den spezifizierten Anforderungen möglich. Je nach Analyseziel können aus dem Systemmodell unterschiedliche Analysen initiiert werden, z.B. die Simulation des Bewegungsverhaltens von Mehrkörpersystemen in Modelica<sup>®</sup> [GTS14].

## 2.4 Wissensmanagement mit Lösungsmustern

Parallel zur steigenden Systemkomplexität und den zuvor beschriebenen Bestrebungen des (Model-Based) Systems Engineerings, dieser zu begegnen, erfordern stetig steigende Marktanforderungen und kürzere Produktlebenszyklen eine höhere Effizienz in der Entwicklung. In diesem Zusammenhang spielt die Wiederverwendung existierender Artefakte eine immer tragendere Rolle. Ein geeignetes Wissensmanagement, nicht zuletzt bedingt durch den demografischen Wandel, muss Bestandteil eines jeden Unternehmens sein. Es hat, als Grundlage für Kreativität und Innovationen, signifikanten Anteil am nachhaltigen Unternehmenserfolg [PRR06], [GLL12], [AG12]. Insbesondere die interdisziplinäre Produktentstehung intelligenter technischer Systemen ist ein äußerst wissensintensiver Prozess, der geeignete Herangehensweisen zum Explizieren, Speichern und Kombinieren von relevantem Wissen erfordert [GDS13]. In der Regel handelt es sich um implizites Wissen, das entweder personengebunden – und somit nur stillschweigend verfügbar (Best Practices, Lessons Learned etc.) – oder unerkannt in Wissensquellen (Entwicklungsdokumente und -modelle etc.) bzw. in deren Vernetzung vorliegt. Es erfordert daher eine adäquate Aufbereitung und Nutzung dieses Wissens, denn dies ist ein wesentlicher Faktor zur Beherrschung der Komplexität in der Entwicklung fortgeschrittener mechatronischer Systeme (vgl. Kap. 2.2.3). Einem erfolgreichen Wissensmanagement muss ein Konzept zugrunde liegen, das den zeitlichen Aufwand für die Wissensaufbereitung so gering wie möglich hält und eine proaktive Unterstützung während der Entwicklung ermöglicht. Ferner muss das Konzept einige Barrieren überwinden: strukturelle (Verteilung von Aufgaben und Kompetenzen auf unterschiedliche Organisationseinheiten), prozessuale (hohe Dynamik im Entwicklungsgeschehen) und politische (Befindlichkeiten Einzelner/Gruppen innerhalb einer Organisation) [LBW93], [GHK+06].

### 2.4.1 Wissensformen

Ein Schwerpunkt der vorliegenden Arbeit liegt auf dem Umgang mit vorhandenem Wissen bzw. Lösungswissen. Eine einheitliche Definition des Begriffes Wissen existiert in der Literatur nicht. Die Spannweite der fachspezifischen Auslegungen reicht von

philosophischen Ansätzen bis hin zu Beschreibungen im Kontext der wissensorientierten Unternehmensführung [Nor05], [PRR06]. Eine für diese Arbeit zutreffende Definition liefern PROBST ET AL.:

*„Wissen bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden. Wissen entsteht als individueller Prozess in einem spezifischen Kontext und manifestiert sich in Handlungen“ [PRR06, S. 40].*

Demnach beruht Wissen auf Daten und Informationen. Die Prozessschritte zur Generierung von Wissen fasst NORTH in Form der Wissenstreppe zusammen (vgl. Bild 2-10). Den Ausgangspunkt für die Erzeugung von Wissen bilden Zeichen. Durch die Anwendung spezifischer Ordnungsregeln (Syntax, Codes) werden aus Zeichen (Buchstaben, Ziffern etc.) Daten. Daten sind demnach beliebige Abfolgen von Zeichen, die noch nicht interpretiert sind, z.B. die Ziffernfolge 31. Diese Daten werden erst dann zu Informationen, wenn sie in einem bestimmten Kontext interpretiert werden oder ihnen eine gewisse Relevanz zugeschrieben wird, z.B. Außentemperatur 31°C. Für den Betrachter sind diese Informationen jedoch wertlos. Erst durch die Vernetzung mit aktuellen oder in der Vergangenheit liegenden Informationen entsteht Wissen. Die Vernetzung der Informationen wird dabei durch das Bewusstsein der Individuen gesteuert. Da Wissen auf Informationen beruht, ist dies auch die Form, in der es gespeichert und weitergegeben wird [Nor05], [PRR06].

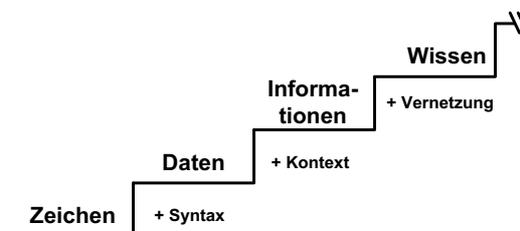


Bild 2-10: Wissenstreppe nach NORTH (Ausschnitt) [Nor05, S. 32]

In der Literatur wird der Begriff Wissen in der Regel weiter unterteilt in implizites und explizites Wissen. Diese Einteilung ist auf den Philosophen und Naturwissenschaftler POLANYI zurückzuführen [Nor05], [NT97], [Pol85].

### **Implizites Wissen:**

Nach POLANYI ist implizites Wissen<sup>27</sup> nicht formalisiertes Wissen. Es ist demnach genau jener Teil des Wissens, der nicht vollständig mit Worten ausgedrückt wird, bzw.

<sup>27</sup>Für eine detailliertere Beschreibung des Begriffes „implizites Wissen“ sei auf [Pol66, S. 4ff.] verwiesen.

ausgedrückt werden kann. Es ist als eine Art körperliche Reaktion oder verinnerlichte Handlung zu verstehen, die Personen in bestimmten Situationen abrufen. Dabei umfasst das implizite Wissen sowohl kognitive als auch körperliche Fertigkeiten.

So verfügen z.B. Mitarbeiter eines Unternehmens, die über einen langen Zeitraum ihren Beruf ausüben, neben ihrem Fachwissen vor allem über Erfahrungen, wobei sie implizites, nicht mitteilbares Wissen nutzen. Sie sind daher im Gegensatz zu Berufsanfängern in der Lage, Sachverhalte besser zu beurteilen oder spezielle Handlungen auszuführen [Pol85].

Die zu fokussierenden Lösungsmuster sollen dazu beitragen, jenes Erfahrungswissen der Mitarbeiter zu formalisieren. Die kritische Definition des impliziten Wissens als nicht formalisierbares Wissen nach POLANYI wird für die vorliegende Arbeit nicht übernommen. Implizites Wissen wird vielmehr als stillschweigend verfügbares Wissen verstanden. Es existiert unerkannt in Entwicklungsdokumenten und in den Köpfen der Experten. Es soll in Form von Lösungsmustern repräsentiert und weiteren Personen zur Verfügung gestellt werden.

### **Explizites Wissen**

Explizites Wissen ist hingegen jenes formalisierte Wissen, welches in einer abstrakten Form, wie beispielsweise Sprache, vorliegt. Diese Art des Wissens lässt sich daher transferieren und kommunizieren. Ein Beispiel ist die Weitergabe von Wissen durch Lehrbücher.

## **2.4.2 SECI-Modell**

Das SECI-Modell (**S**ocialisation, **E**xternalization, **C**ombination, **I**nternalization) beschreibt den Ablauf der Wissenserzeugung in Unternehmen. Als eine Art Referenzmodell bildet es die Grundlage der angestrebten Systematik für den Einsatz von Lösungsmustern. Das Modell gliedert sich in die vier aufeinander folgenden Prozesse Sozialisation, Externalisierung, Internalisierung und Kombination, die kontinuierlich durchlaufen werden. Das Wissen wird dabei innerhalb eines Unternehmens von individuellem Wissen auf höhere Organisationsstufen wie Personengruppen oder die gesamte Organisation gehoben (vgl. Bild 2-11) [NT97, S. 84ff.].

**Sozialisation (Austausch von implizitem Wissen):** Bei der Sozialisation kommt es zum Austausch von implizitem Wissen zwischen Individuen durch gemeinsame Erfahrungen, Beobachtungen oder Nachahmung von bestimmten Handlungen. Der Wissensaustausch verläuft weitestgehend stillschweigend. Daher ist die Übertragung von vorhandenem Wissen nur begrenzt möglich. Ein Beispiel für diesen Prozess ist die Integration eines neuen Mitarbeiters in die Denk- und Arbeitsroutinen einer Abteilung.

**Externalisierung (implizites Wissen wird zu explizitem Wissen):** Die Umwandlung von implizitem Wissen in explizites Wissen verlangt nach Möglichkeiten, schwer be-

schreibbare Zusammenhänge allgemeinverständlich wiederzugeben. Es ist beispielsweise möglich, etwas durch eine Metapher zu umschreiben oder Analogien zu bekannten Sachverhalten zu bilden. Weitere Möglichkeiten bestehen in der Verwendung von Visualisierungshilfen wie Modellen und Diagrammen. Individuen arbeiten hierfür in Gruppen zusammen, um das Wissen aufzuzeichnen und somit übertragbar zu machen. Am Ende dieses Prozesses besitzen die beteiligten Individuen einer Gruppe das gleiche explizite Wissen.

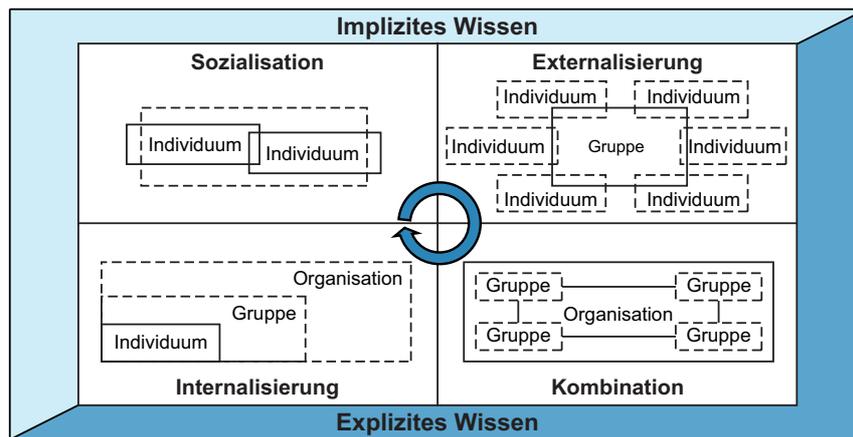


Bild 2-11: SECI-Modell nach NONAKA und TAKEUCHI [vgl. NT97, S. 84]

**Kombination (Austausch von explizitem Wissen):** Das im Externalisierungsprozess gewonnene Wissen wird im Rahmen der Kombination durch den Zusammenschluss verschiedener Gruppen und deren Gruppenwissen erweitert. Dieses explizite Wissen wird in der Regel durch offizielle Handlungsroutinen und Abläufe repräsentiert. Durch die Nutzung der Informationstechnologie ist das Wissen folglich für die Mitglieder der gesamten Organisation zugänglich.

**Internalisierung (explizites Wissen wird zu implizitem Wissen):** Die Mitarbeiter verfügen nun über das im Kombinationsprozess generierte Wissen und wenden dieses anschließend in ihren unterschiedlichen Abteilungen an. Dabei wird das explizit vorliegende Wissen der Organisation, zumindest teilweise, wieder in implizites Wissen der Individuen umgewandelt.

Die Analyse des SECI-Modells verdeutlicht, dass der Einsatz von Lösungsmustern eine übergreifende Plattform für die Dokumentation und Pflege von Wissen voraussetzt. Nur so ist es möglich, dass sich Ingenieure oder Experten der beteiligten Disziplinen über spezifische Sachverhalte und ihr Zusammenwirken austauschen können.

### 2.4.3 Kernprozesse des Wissensmanagements nach PROBST

PROBST ET AL. fassen die auftretenden Herausforderungen im Umgang mit Wissen innerhalb eines Unternehmens in übergeordnete Kategorien zusammen. Aus den notwendigen Aktivitäten zur Problembewältigung ergeben sich sechs Kernprozesse des Wis-

sensmanagements (vgl. Bild 2-12) [PRR06, S. 31ff.]. Diesen muss das angestrebte Instrumentarium, insbesondere im Hinblick auf den musterbasierten Systementwurf, gerecht werden.

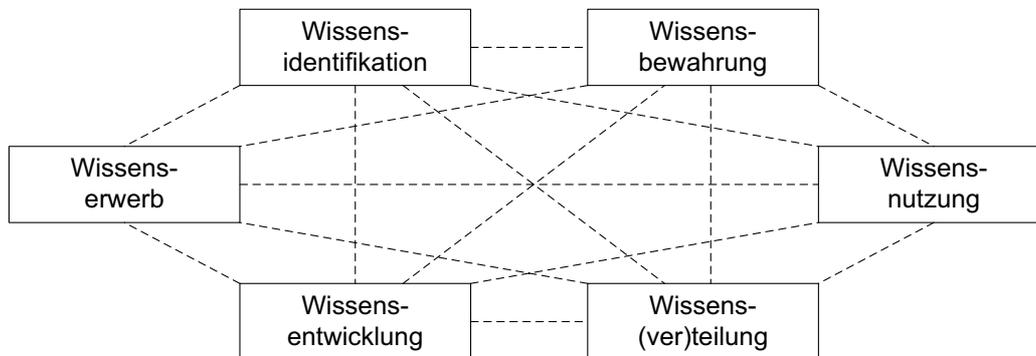


Bild 2-12: Kernprozesse des Wissensmanagements [PRR06, S. 32]

**Wissensidentifikation:** Der Kernprozess Wissensidentifikation beschäftigt sich mit der Fragestellung, wie intern und extern Transparenz über vorhandenes Wissen geschaffen werden kann. Diese Transparenz ist notwendig, um interne und externe Daten, Informationen und Fähigkeiten zu erfassen, zu speichern und anschließend nutzen zu können.

**Wissenserwerb:** Bei der systematischen Umsetzung eines Wissensmanagements liegt ein wesentlicher Schwerpunkt auf dem Wissenserwerb. Externe Quellen, wie Kunden oder Lieferanten, stellen ein erhebliches und oft unausgeschöpftes Potential des Wissenserwerbs dar. Darüber hinaus ist insbesondere für kleine Unternehmen mit einer geringen Entwicklungstiefe der Erwerb von Experten-Know-how von hoher Bedeutung.

**Wissensentwicklung:** Ziel der Wissensentwicklung ist die Produktion neuer Fähigkeiten, neuer Produkte, besserer Ideen sowie leistungsfähigerer Prozesse. Eine Voraussetzung für den Unternehmenserfolg ist hierbei die bereichsübergreifende Wissensentwicklung. Diese beruht im Wesentlichen auf der Nutzung der Kreativität der eigenen Mitarbeiter aller Abteilungen.

**Wissens(ver)teilung:** Um vorhandenes Wissen für die gesamte Organisation nutzbar zu machen, ist die Verteilung von Informationen oder Erfahrungen erforderlich. Ein Schwerpunkt liegt hierbei auf der Wissensübertragung von der individuellen auf die Gruppen- oder Organisationsebene.

**Wissensnutzung:** Ziel und Zweck des Wissensmanagements ist die Wissensnutzung. Diese beschreibt den produktiven Einsatz des organisationalen Wissens. Eine effektive Nutzung beruht dabei insbesondere auf einer erfolgreich umgesetzten Wissensentwicklung und -verteilung. Auch der Wissenserwerb kann einen hohen Einfluss haben, ist jedoch durch eine Vielzahl von Barrieren hinsichtlich der Nutzung fremden Wissens begrenzt einsetzbar.

**Wissensbewahrung:** Die Basis eines nachhaltigen Wissensmanagements ist die gezielte Bewahrung von Erfahrungen und Informationen. Dieser Prozess der Wissensbewah-

nung beruht dabei auf der Nutzung verschiedener organisationaler Speichermedien für Wissen.

#### 2.4.4 Lösungsmuster – ein geeigneter Wissensmanagementansatz?

Aufbauend auf grundsätzlichen Abläufen des Wissensmanagements setzt sich dieses Kapitel mit der Fragestellung auseinander, ob Lösungsmuster ein geeigneter Wissensmanagementansatz sind. Dies gilt sowohl zur Unterstützung des fachdisziplinübergreifenden Entwurfs, als auch für seine Einbettung in die Produktentstehung. In diesem Zusammenhang wird dabei der These gefolgt, dass aus einer Kombination der Stärken aus der Konstruktionswissenschaft und dem Systems Engineering die erfolgversprechende Gesamtmethodik für die Entwicklung fortgeschrittener mechatronischer Systeme resultiert. In diesem Zusammenhang sind folgende Forderungen aus der Industrie wesentlich für die Weiterentwicklung und Anwendbarkeit des Systems Engineering: Interdisziplinarität, Durchgängigkeit und Modellbasiertheit [GDS13]. Für die Identifizierung eines geeigneten Konzepts erfolgt daher zu Beginn eine Gegenüberstellung etablierter Ansätze für das Wissensmanagement mit den genannten Forderungen (vgl. Bild 2-13).

Bewertung: ○ = Potential nicht vorhanden ● = Potential vorhanden	Interdisziplinarität	Modellbasiertheit	Durchgängigkeit
Knowledge Maps/ Topic Maps	●	○	●
Template	●	●	●
Feature	○	●	○
Wirkprinzipien	○	○	○
Patente – Innovative Prinzipien	●	○	○
Muster	●	●	●
...			

Bild 2-13: Gegenüberstellung – Wissensmanagementansätze und Forderungen aus der Industrie

*Interdisziplinarität* beschreibt die Art und Weise der Wissensrepräsentation und -präsentation. Das Konzept sollte für alle an der Produktentwicklung beteiligten Fachdisziplinen gleichermaßen verständlich sein. *Modellbasiertheit* steht für die Forderung nach einer Wissensrepräsentation in Form von Modellen und die damit einhergehende proaktive Integration in die Entwicklung. Die Eigenschaft *Durchgängigkeit* fordert die kontinuierliche Anwendung von nur einem Konzept über alle Phasen der Produktentwicklung hinweg. So muss beispielsweise Lösungswissen für die Konzipierung nahtlos mit dem Lösungswissen für den Softwareentwurf verbunden sein.

**Knowledge Maps** oder **Topic Maps** ist ein Wissensmanagementansatz bei dem die individuelle Wissensaufbereitung mit Hilfe semantischer Technologien erfolgt. Ziel ist die Vernetzung modular aufgebauter textbasierter Dokumente, z.B. Steckbriefe über Lessons Learned oder Best Practices [Web10], [Mau11]. Der wesentliche Nachteil ist, dass die Wissenspräsentation nicht mit Modellen aus der Entwicklung oder Prozessplanung erfolgt, sondern textuell oder mit Hilfe von Skizzen.

Ein **Template** (engl. für Vorlage) definiert eine stark vorstrukturierte Lösung im Sinne einer Schablone mit fest definierten Variationspunkten und Variablen. Es ist ein Ansatz, der je nach Ausprägung, den aufgezeigten Forderungen gerecht werden kann.

In Anlehnung an die VDI 2218 sind **Feature** informationstechnische Elemente, die Bereiche von besonderem (technischen) Interesse von einzelnen oder mehreren Produkten darstellen [VDI2218]. Im Maschinenbau ist die Rede von Funktions-, Prinzip-, Bau- oder auch Form-Feature. Die Interpretation des Begriffs Feature ist über die Fachdisziplinen hinweg jedoch keineswegs einheitlich [MBD+05]. Eine vielfach zitierte Definition in der Softwareentwicklung liefern KANG ET AL. in der FODA (Feature Oriented Domain Analysis). Demnach ist in der Übersetzung des englischsprachigen Originals ein Feature:

*„Ein auffälliges oder unterscheidungskräftiges für den Kunden sichtbares Qualitätsmerkmal oder Charakteristikum eines oder mehrerer Softwaresysteme“ [vgl. KCH+90, S. 3].*

Die unterschiedlichen Definitionen von Feature und Variationen bei der Nutzung sind Barrieren für eine durchgängige Übertragung auf die Produktentstehung.

**Wirkprinzipien**<sup>28</sup> sind vor allem im klassischen Maschinenbau bekannt und etabliert. Es existieren zahlreiche Konstruktionskataloge in Form von Büchern oder Softwarewerkzeugen [Rot01] [Lug06-ol]. Für ein durchgängiges Wissensmanagement über alle Bereiche der Produktentstehung eignen sich Wirkprinzipien nicht, sollten aber in geeigneter Weise berücksichtigt werden. Nach DUMITRESCU repräsentieren Wirkprinzipien elementare Lösungsprinzipien auf denen komplexe *Lösungsmuster* beruhen. Als Pendant auf Seiten der Software werden Verfahren oder Algorithmen genannt [Dum10].

Die Nutzung von **Patenten** ist außerordentlich erfolgversprechend in der systematischen Generierung von neuartigen Produktideen. So definierte z.B. ALTSCHULLER in der Theorie des erfinderischen Problemlösens (TRIZ) wiederkehrende innovative Prinzipien aus einer Analyse von ca. 200.000 Patenten; die Rede ist auch von wiederkehrenden *Denkmustern* zur Lösung von Problemen [GHK+06].

---

<sup>28</sup>Ein Wirkprinzip beschreibt den Grundsatz, von dem sich eine bestimmte Wirkung zur Erfüllung einer Funktion ableitet (physikalischer, biologischer, chemischer Effekt (oder Effekte) sowie geometrische und stoffliche Merkmale in Verbindung mit einer Teilfunktion) [PBF+07, S. 54].

Bei der Definition der genannten Begriffe kommt es wiederholt zur Verwendung des Begriffs **Muster**. Die in der Literatur am häufigsten zitierte Definition von Mustern im Kontext der Produktentstehung geht zurück auf den Architekturtheoretiker ALEXANDER. Nach ALEXANDER beschreibt:

*„...jedes Muster ein in unserer Umwelt immer wieder auftretendes Problem sowie den Kern der Lösung dieses Problems, und zwar so, dass man diese Lösung millionenfach anwenden kann, ohne sich je zu wiederholen“ [AIS+95, S. 12].*

Im Umgang mit Mustern ist es entscheidend, das gesamte Problem in Teilprobleme zu zerlegen, für die eine Lösung ausgearbeitet werden kann. Die anschließende Zusammenführung der Teil- bzw. Grundprobleme soll zur Lösung des übergeordneten Gesamtproblems führen. Der Anwender eines solchen Musters erhält erste Lösungsansätze für sein Problem bzw. Teilproblem. Aufgrund der Tatsache, dass in derartigen Mustern lediglich der Problem- und Lösungskern beschrieben wird, ist eine Anpassung an die jeweiligen Randbedingungen des Gesamtproblems notwendig. Um die Anwendung des Musters sicherzustellen, spezifiziert ALEXANDER zusätzlich eine sog. Mustersprache<sup>29</sup>. Diese Sprache soll dazu beitragen, die Muster in ein einheitliches, übersichtliches und zusammenhängendes Format zu bringen. Es stellt die verträgliche Verknüpfung zu über- und untergeordneten Mustern sicher. Hierzu definiert ALEXANDER vier Kategorien, die jedes Muster enthalten muss [AIS+95]:

- **Name** und Beispielbild
- **Kontext**: In dieser Kategorie ist der Kontext, in dem das Muster eingesetzt wird, beschrieben. Zusätzlich wird erklärt, wie es sich in übergeordnete Muster einfügen lässt.
- **Problem**: In diesem Abschnitt des Musters erfolgt eine detaillierte Problembeschreibung.
- **Lösung**: Die Darstellung der Problemlösung ist der wichtigste Teil eines Musters. Die Lösung wird dabei im Sinne einer Anleitung für den Anwender beschrieben. Abschließend folgt eine Erklärung über die Verknüpfung zu untergeordneten Mustern.

BARTER und HASKINS liefern eine vereinfachte Definition, nach der *„ein Muster definiert ist als eine Lösung für ein spezifisches Problem innerhalb eines bestimmten Kontextes“* [Bar98], [Has05]. Die Wissensrepräsentation eines Musters unterteilt sich dabei

---

<sup>29</sup>Nach BECK wird *„eine Menge von Mustern zur Mustersprache, wenn die Ausprägung von Mustern, zu weiteren Mustern führt, die anschließend betrachtet werden sollten“* [BC87]; Vergleich mit natürlicher Sprache: Lösungsmuster = Vokabeln; Beziehung zw. den Mustern = Grammatik; Lösungsmuster + Beziehungen = (Muster)Sprache

in die Kategorien Name, Problem, Lösung und Kontext. Der Kern ist somit eine verallgemeinerte Problem- und Lösungsbeschreibung.

In der Definition ist ein Muster derart abstrakt, dass die zuvor genannten Ansätze teilweise als solche interpretiert werden können. Es ist daher eher ein übergeordnetes Konzept, das etablierte Herangehensweisen geeignet vereinen kann. So können die 40 innovativen Prinzipien von ALTSCHULLER als stark generalisierte Muster verstanden werden, auf die alle Problem- und Lösungsbeschreibungen zurückgeführt werden können. Je stärker ein Problem-Lösungspaar abstrahiert bzw. generalisiert wird, desto größer werden der Anwendungskontext und die Möglichkeiten der spezifischen Ausprägungen. Diesem Grundgedanken folgte ALTSCHULLER bei der Identifizierung der innovativen Prinzipien bei der Analyse von Patenten.

Zahlreiche Untersuchungen und Diskussion zeigen, dass das Konzept der Muster bzw. Lösungsmuster den aufgezeigten Forderungen an ein durchgängiges Wissensmanagement gerecht werden kann. Bei einer konsequenten Übertragung auf die gesamte Produktentstehung besitzt der Musteransatz auch das Potential, die eingangs skizzierten Barrieren (strukturell, prozessual und politisch) zu überwinden (vgl. Kapitel 2.4). Auf Basis eines umfangreichen Überblicks nach RISING und weitergehenden Überlegungen von DEIGENDESCH, SANZ/ZALEWSKI und SUHM lassen sich folgende Vorteile von Lösungsmustern herausstellen [Ris98], [Ris00], [Dei09], [SZ03], [Suh93]:

- **Übertragbarkeit:** Der Aufbau von Lösungsmustern unterliegt einer einheitlichen Struktur. Dies ermöglicht die Übertragbarkeit auf alle an der Produktentstehung beteiligten Bereiche. Lösungsmuster bieten somit die Möglichkeit der sprachenunabhängigen Wiederverwendung bzw. Repräsentation von Wissen.
- **Verbesserung der Kommunikation:** Lösungsmuster verbessern die Kommunikation auf vielfältige Art und Weise. Sie verbessern die Kommunikation zwischen: 1) einem Entwickler und sich selbst, 2) Entwicklern innerhalb eines Teams (fachdisziplinspezifisch) sowie 3) Entwicklern mehrerer Teams (fachdisziplinübergreifend). Sie bilden folglich die Plattform zur Reflektion abgeschlossener Entwicklungstätigkeiten und somit entwicklungsrelevanten Wissens.
- **Langfristige Dokumentation:** Lösungswissen in der Produktentstehung liegt häufig nur implizit vor, z.B. in langjähriger Erfahrung einzelner Mitarbeiter. Dieses Wissen kann in Form von Lösungsmustern externalisiert und somit Dritten zur Verfügung gestellt werden. Ein Lösungsmuster ist ein sehr gut wartbares Dokument, das fortwährend modifiziert und optimiert werden kann. Lösungsmuster eignen sich vor allem im Hinblick auf eine zielgerichtete Suche, da neben dem Musternamen, eine Vielzahl an Informationen das zugrunde liegende Problem charakterisiert.
- **Komplexitätsreduktion:** Lösungsmuster tragen erheblich zur Komplexitätsreduktion umfangreicher Problemstellungen bei. Es bietet sich an, komplexe Ge-

samtprobleme in elementare Probleme zu zergliedern. Dies schließt die Zerlegung der zugehörigen Lösungen mit ein. Lösungsmuster lassen sich daher voneinander abgrenzen, wobei sich starre Klassifikationen und Hierarchien als wenig zielführend erweisen. Vielmehr sind wechselseitige Beziehungen zwischen Lösungsmustern in geeigneter Form zu beschreiben.

- **Effizienzsteigerung:** Mit Hilfe von Lösungsmustern lässt sich die Effizienz in Problemlösungsprozessen signifikant steigern. Es wird die Möglichkeit geschaffen, kollektives Erfahrungswissen zielgerichtet in neue Entwicklungsprojekte einzubringen. Gemäß der Zielsetzung von Ansätzen zum *Best-Practice* und *Lessons-learned* helfen Lösungsmuster, Erfolge aus der Vergangenheit hervorzuheben und Misserfolge zukünftig zu vermeiden.
- **Förderung der Kreativität:** Lösungsmuster enthalten im Vergleich zu detaillierten Lösungen, abstrahierte (generalisierte) Informationen über eine Problemlösungskombination. Der Generalisierungsgrad lässt sich dabei keinesfalls einheitlich festlegen. Mit zunehmender Generalisierung steigt demzufolge die Größe des Anwendungskontextes eines Musters. Ihr Einsatz in einem Lösungsprozess erfordert daher immer eine spezifische Anpassung auf das jeweilige Problem. Eine Vielzahl dokumentierter Lösungsmuster gibt Entwicklern Impulse, ihre eigenen Denkmuster mit dem Wissen des Kollektivs zu reflektieren und zu erweitern. Lösungsmuster unterstützen folglich die Kreativität jedes Einzelnen; die Grundlage für die Entstehung von Innovationen.

#### 2.4.5 Lösungsmuster in der Produktentstehung – Geschichte

Das vorliegende Kapitel präsentiert die Analyse ausgewählter Lösungsmuster in ihrer geschichtlichen Entwicklung. Zweifelsohne beruhen die für die Produktentstehung relevanten Ansätze zu Lösungsmustern auf den Arbeiten des Architekturtheoretikers ALEXANDER [AIS+77]. Der Grundgedanke, Lösungswissen in Form von Problemlösungsbeschreibungen Dritten zur Verfügung zu stellen, wurde anschließend in der Informatik aufgegriffen. Große Popularität genießen dort nach wie vor die Arbeiten von BECK und CUNNINGHAM, der Gang of Four<sup>30</sup>, COPLIEN<sup>31</sup> oder auch RISING [BC87], [GhJ+94], [Cop91], [Ris98]. Die Arbeiten setzen sich intensiv mit grundsätzlichen Fragestellungen im Kontext Lösungsmuster sowie Mustersprachen auseinander. Nach BECK ist eine Mustersprache das Ergebnis von zeitlich aufeinander aufbauenden Mustern. Dies bedeutet, dass die Anwendung von Mustern die Grundlage für die Auswahl weiterer Muster ist [BC87].

---

<sup>30</sup>GAMMA ET AL. verhalfen im Jahre 1994 mit dem Buch „Design Patterns – Elements of Reusable Object-Oriented Software“ den Entwurfsmustern zum Durchbruch in der Softwareentwicklung [GHJ+94]. Sie sind seitdem unter den Spitznamen „Gang of Four“ (GoF) bekannt.

<sup>31</sup>James Couplien war einer der Mitbegründer der PLoP (Pattern Language of Programs) Konferenz.

Eine sehr generische, vor allem auch namentliche, Übertragung des Musteransatzes auf den Maschinenbau liefert GRABOWSKI in der Universal Design Theory [GL00], [GL03]. Er unterteilt Lösungsmuster grundsätzlich in Objektmuster und Prozessmuster. Mit Objektmustern lassen sich Problemlösungen beschreiben, die sich anhand von Objekten manifestieren. Sie sind innerhalb eines definierten Kontextes gültig und unterstützen den Lösungsfindungsprozess. Diese Muster repräsentieren die Gestalt des Objektes elementar und deklarativ<sup>32</sup>. Beispiele sind physische Erzeugnisse oder auch Softwarecode. Ein Prozessmuster beschreibt den aktiven Gebrauch von Objektmustern. Im Gegensatz zur deklarativen Wissensrepräsentation eines Objektmusters handelt es sich dabei um eine prozedurale Wissensrepräsentation. Ein Prozessmuster enthält somit eine Menge von Anweisungen, die zu einem bestimmten Ziel führen [Amb98].

Zusammenfassend lassen sich existierende Ansätze in die folgenden Kategorien unterteilen: **Objektmuster**, **Prozessmuster** und **Mustersprache**. Anhand dieser Unterteilung folgt die Gegenüberstellung von Lösungsmustern im Vergleich zwischen der *Entwicklung mechatronischer Systeme* und dem *Systems Engineering*. Die erwähnte Gegenüberstellung zeigt Bild 2-14. Fundamentale Arbeiten, auf denen die dargestellten Ansätze beruhen, sind farblich rot hervorgehoben. Es sind auch Ansätze aufgeführt, die von der Titulierung her von dem Begriff Lösungsmuster abweichen, jedoch durch die Wiederverwendung bestimmter Prinzipien oder Bausteine als Lösungsmuster interpretiert werden können.

Auf Seiten der **Entwicklung mechatronischer Systeme** existieren traditionelle Ansätze (*Objektmuster*) der Regelungstechnik nach FÖLLINGER oder Wirkprinzipien im klassischen Maschinenbau nach PAHL/BEITZ [Foe72], [PB77]. Nach den Arbeiten von GAMMA ET AL. häuften sich die Ansätze auch mit entsprechender Titulierung. Zu nennen sind u.a. die Lösungsmuster nach GRABOWSKI, ALBERS oder auch DZIWOK [GL00], [AT11], [GSA+11], [DHT12]. Allen Ansätzen ist gemein, dass sie fachdisziplinspezifisch und innerhalb der Disziplin sehr fokussiert sind. So konzentriert sich beispielsweise ALBERS auf Entwurfsmuster in der Mikrotechnik. DZIWOK hingegen beschäftigt sich mit Koordinationsmustern für echtzeitkritische Softwarekommunikation. Eine Ausnahme bilden die Lösungsmuster für die Selbstoptimierung nach DUMITRESCU. Hier werden Ansätze der höheren Mathematik oder der künstlichen Intelligenz fachdisziplinübergreifend behandelt. Die Notwendigkeit nach wiederverwendbaren Prozessbausteinen beschreiben u.a. MURR, BICHLMAIER oder auch KAHL [Mur99], [Bic00], [Kah13]. Sie sind der hohen Dynamik im Entwicklungsgeschehen und der Forderung nach immer kürzer werdenden Time-To-Market Zeiten neuer Produktserien geschuldet. Diesen Anforderungen begegnen die genannten Ansätze unter Zuhilfenahme von Erfahrungswissen in

---

<sup>32</sup> „Die deklarative Wissensrepräsentation konzentriert sich auf die reine Beschreibung von Sachverhalten und macht grundsätzlich keine Angaben zur Anwendung des Wissens für eine konkrete Problemlösung“ [Suh93].

Form von Prozessbausteinen (*Prozessmustern*) während der Projekt- (bzw. Entwicklungsprozess-)planung.

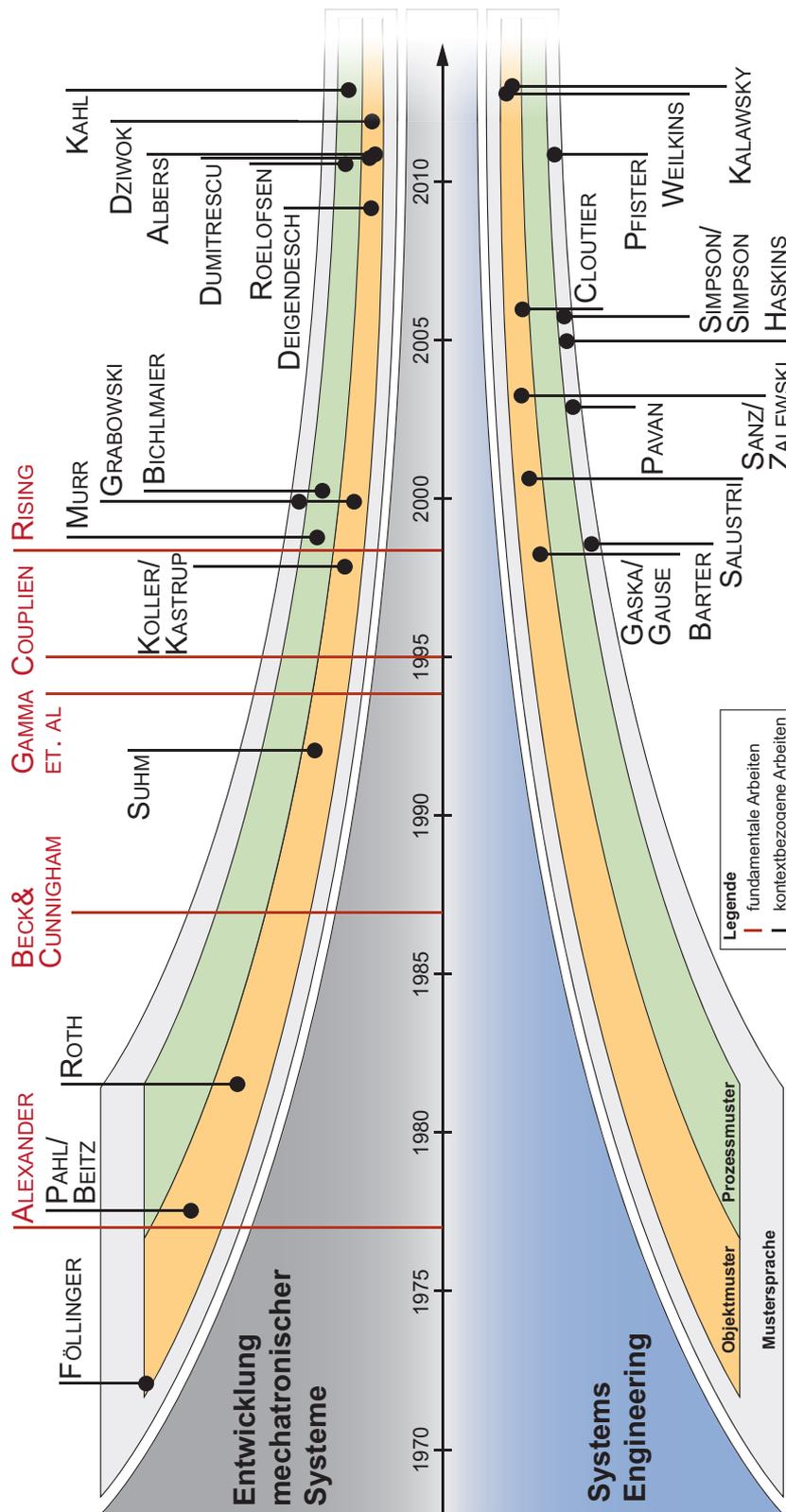


Bild 2-14: Geschichte von Lösungsmustern (Auszug) im Vergleich zwischen Systems Engineering und der Entwicklung mechatronischer Systeme

Auch im Kontext **Systems Engineering** begann der Einsatz von Lösungsmustern nach den Arbeiten von GAMMA ET AL. Zu nennen sind hier u.a. die Arbeiten von GASKA/GAUSE oder CLOUTIER [GG98], [CD06]. Im Gegensatz zu den zuvor geschilderten fachdisziplinspezifischen Ansätzen z.B. von ALBERS wird die Nutzung von Lösungsmustern im Systems Engineering deutlich fachdisziplinübergreifender interpretiert. Dies bezieht sich weniger auf die Art der Wissensrepräsentation sondern vielmehr auf ihren Einsatz in vielen an der Produktentstehung beteiligten Bereichen. Dies verdeutlicht eine Übertragung des Lösungsmusteransatzes auch auf nicht technische Bereiche durch CLOUTIER [CD06]. Er nennt u.a. Organisationsmuster oder Geschäftsmuster sowie deren Zuordnung zu Systemarchitektur-, und Systementwurfsmustern. Es handelt sich hierbei in Bezug auf die Unterteilung nach GRABOWSKI um *Objektmuster*.

Bei Systementwurfsmustern betont CLOUTIER ihren Neuheitsgrad und die Aktualität im Hinblick auf die steigende Komplexität der zu entwickelnden Systeme. Neben den bereits in Kapitel 2.4.4 aufgeführten Vorteilen von Lösungsmustern hebt CLOUTIER nachfolgende Eigenschaften gesondert hervor:

- Die Beschreibung von Lösungsansätzen für das Gesamtsystem in Form von Mustern verbessert das gemeinschaftliche Verständnis aller partizipierenden Stakeholder.
- Ihr fachdisziplinübergreifender Charakter fördert die Team-Kommunikation zwischen Mitgliedern des Systemarchitekturteams und den Entwicklungsteams (z.T. fachdisziplinspezifisch).
- Mit Hilfe von Lösungsmustern lässt sich ein einheitliches Lexikon für Architekturteams erarbeiten. Es beinhaltet wiederverwendbare multidisziplinäre Lösungen für den Austausch zwischen den Systemarchitekten.
- Die Standardisierung in Form einheitlicher Muster fördert die Effizienz in den frühen Phasen des Entwicklungsgeschehens auf Gesamtsystemebene.

Ferner existieren im Systems Engineering Ansätze, die sich mit dem Themenfeld *Mustersprache* auseinandersetzen. Kern einer solchen Sprache sind Beschreibungen der wechselseitigen Beziehungen zwischen den Mustern [SS06]. Zu nennen sind hier u.a. die Arbeiten von PAVAN, SIMPSON oder auch PFISTER [PMZ03], [SS06], [PCH+11]. Sie beschreiben Beziehungsformen zwischen Mustern auf einer Meta-Ebene. Beziehungen können z.B. sein: äquivalent, nachfolgend zu betrachten, zusätzlich erforderlich, in Kombination nicht verträglich [PCH+11]. Relevante Mustersprachen (fachdisziplinübergreifend) bezogen auf den Kontext der vorliegenden Arbeit für die Entwicklung mechatronischer Systeme existieren nicht.

## 2.5 Problemabgrenzung

Zusammenfassend lässt sich festhalten, dass ein wesentlicher Stellhebel für eine effiziente Entwicklung fortgeschrittener mechatronischer Systeme die Wiederverwendung relevanten Lösungswissens ist. Zukünftig wird es in diesem Zusammenhang mehr denn je darauf ankommen, das zu entwickelnde System als Ganzes zu betrachten und zu beherrschen (Kap. 2.3.3). Die zunehmende Intelligenz in den adressierten technischen Systemen (Kap. 2.2) erfordert ein fachdisziplinübergreifendes Denken und Handeln über alle an der Produktentstehung beteiligten Bereiche hinweg (Kap. 2.3). Dies schließt, neben den vorrangig technischen Disziplinen, die nicht technischen Segmente wie z.B. den Vertrieb mit ein. Erfolgversprechende Entwicklungsmethodiken werden zukünftig aus dem Verschmelzen der Denkweisen der Konstruktionslehre (Kap. 2.3.2) und dem (Model-Based) Systems Engineering (Kap. 2.3.4) hervorgehen. Historisch betrachtet lassen sich dort signifikante Stärken für die Entwicklung komplexer technischer Systeme feststellen, die gleichermaßen für Entwicklungsmethodiken als auch für Wissensmanagementansätze mit Lösungsmustern gelten (2.4.5). In der Konstruktionslehre bzw. Entwicklung mechatronischer Systeme sind dies in erster Linie etablierte, jedoch fachdisziplinspezifische Ansätze z.T. basierend auf Objektmustern (z.B. Wirkprinzipien). Im Systems Engineering sind es vor allem die ganzheitlichen fachdisziplinübergreifenden Vorgehensweisen zur Unterstützung des modellbasierten Systementwurfs. Für das angestrebte Instrumentarium sind drei grundsätzliche **Herausforderungen** hervorzuheben:

### **Ungeklärter Lösungsmusterbegriff/-überblick:**

Ansätze zur Wiederverwendung von Lösungswissen in Form von Mustern gibt es zahlreich. Die Interpretation und die Einsatzgebiete sind jedoch äußerst vielfältig. Dies reicht von Beispielen, über die Art der Darstellung, die damit einhergehende Strukturierung bis hin zur Charakterisierung. Gemeint sind z.B. Unterscheidungen wie Objektmuster, Prozessmuster, elementare oder komplexe Muster [GL00], [Suh93]. Da diese Typisierungen aus einzelnen Disziplinen (z.B. dem Maschinenbau) kommen, stellt sich die Frage nach ihrer Allgemeingültigkeit. Eine Beantwortung erfordert eine umfangreiche Übertragung des Mustergedankens auf die Produktentstehung, was die nicht technischen Bereiche einschließt. Es ist daher erforderlich aufbauend auf Strukturierungsansätzen für die Produktentstehung wie z.B. dem 3-Zyklus-Modell nach GAUSEMEIER und Musterhierarchien aus dem Systems Engineering, eine umfangreiche Beschreibung für Lösungsmuster in der Produktentstehung zu erarbeiten [GP14], [CD06]. Nur auf diese Weise kann eine Einbettung von Lösungsmustern für den Systementwurf in eine musterbasierte Produktentstehung gelingen. Damit einher geht die Notwendigkeit, Aufgabenbereiche der Lösungsmuster festzulegen, um sowohl ihre Eigenständigkeit als auch Interaktion aufzuzeigen.

### **Dokumentation interdisziplinären Lösungswissens:**

Die Entwicklung fortgeschrittener mechatronischer Systeme kann nur durch das Zusammenwirken verschiedener Fachdisziplinen erfolgreich sein. Eine Hauptschwierigkeit liegt in der aufwendigen Kooperation und Kommunikation auf Gesamtsystemebene während des Systementwurfs. Hier sind Konzepte zur frühzeitigen, durchgängigen und gleichberechtigten Behandlung des Lösungswissens aus den partizipierenden Disziplinen notwendig. Die Forderung nach Durchgängigkeit birgt besondere Anforderungen aufgrund von unterschiedlichen Denkweisen. Während die eine Disziplin oftmals zu Beginn einer Entwicklung sehr abstrakt in Form von Wirkprinzipien denkt, startet eine andere Disziplin mit formalen Spezifikationen (u.a. mit Hilfe von Modellbibliotheken). Im Rahmen des Systementwurfs müssen derartige Denk- und Herangehensweisen zusammengeführt werden, was ohne weiteres aktuell nicht möglich ist. Das für den Systementwurf notwendige Wissen muss daher geeignet, in Form von Lösungsmustern, aufbereitet und repräsentiert werden. Es muss zudem ein nahtloser Übergang in den fachdisziplinspezifischen Entwurf oder in formale Analyseschritte möglich sein. Hierzu sind Beziehungen zwischen den Lösungsmustern für den Systementwurf mit z.B. Wirkprinzipien oder fachdisziplinspezifischen Lösungsmustern (z.B. Software) zu beschreiben.

### **Fehlende Entwicklungssystematik:**

Die heute etablierten Entwicklungssystematiken werden der Forderung nach einem durchgängigen Wissensmanagement, mit besonderem Fokus auf dem interdisziplinären Systementwurf, nicht gerecht. Sie fokussieren in der Regel nur einen bestimmten Bereich oder eine Disziplin; eine ganzheitliche Betrachtung von wiederverwendbaren z.T. fachdisziplinübergreifenden Produktartefakten findet – wenn überhaupt – nur ansatzweise statt. Die Folge ist in der Regel die Kombination von disziplinspezifischen Teillösungen, die nur selten die bestmögliche Gesamtlösung ergeben. Es ist daher eine Systematik erforderlich, mit der Lösungsmuster für den Systementwurf identifiziert, dokumentiert und angewendet werden können.

Lösungsmuster für den Systementwurf werden zukünftig ein zentraler Baustein in der Entwicklung fortgeschrittener mechatronischer Systeme sein. Sie tragen durch ihren fachdisziplinübergreifenden Charakter und ihren systematischen Einsatz wesentlich zur Effizienzsteigerung im Entwurf und zur Komplexitätsbeherrschung auf Gesamtsystemebene bei. Es besteht daher **Bedarf** nach einem *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*. Es nimmt sich den skizzierten Herausforderungen an und gliedert sich in die in Bild 2-15 abgebildeten Handlungsfelder (HF):

### **Handlungsfeld 1: Das Paradigma der Lösungsmuster in der Produktentstehung**

Basis des Instrumentariums ist eine ausführliche Beschreibung von Lösungsmustern in der Produktentstehung. Grundlage ist eine einheitliche Definition von Lösungsmustern

für die gesamte Produktentstehung. Auf Basis etablierter Strukturierungskonzepte sind Lösungsmuster in der Produktentstehung und ihr Zusammenwirken anhand von Beispielen zu beschreiben. Gefordert ist eine holistische Darstellung, z.B. in Form einer Hierarchie oder Klassifikation, in der Wissen aus den beteiligten Bereichen gleichermaßen berücksichtigt und miteinander in Verbindung gesetzt wird.

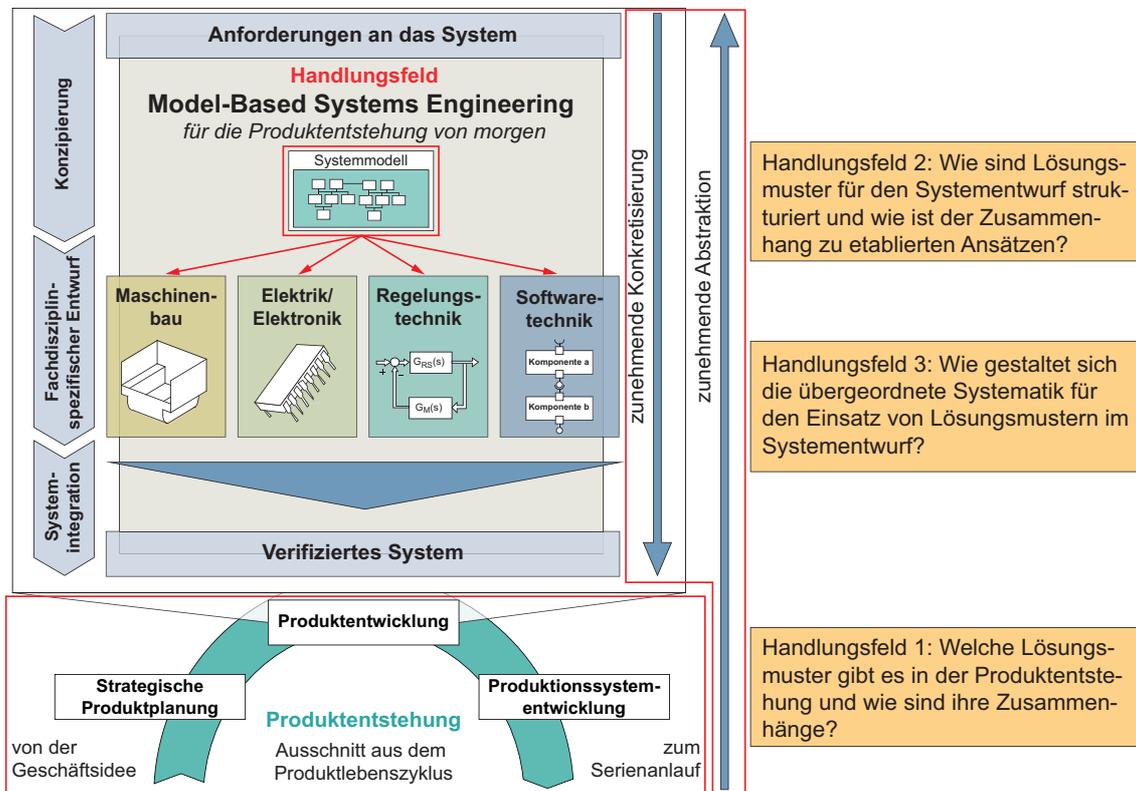


Bild 2-15: Übersicht der Handlungsfelder für das angestrebte Instrumentarium

## Handlungsfeld 2: Lösungsmuster für den Systementwurf – Strukturierung, Charakterisierung und Techniken zur Wissensrepräsentation

Lösungswissen nimmt in der Produktentstehung immer weiter an Bedeutung zu. Im Instrumentarium erfolgt die Dokumentation und Wiederverwendung des Wissens in Form von Lösungsmustern. Aufgrund der hohen Interdisziplinarität ist eine einheitliche Struktur erforderlich, die eine barrierefreie Kooperation der beteiligten Akteure und dem Systems Engineer ermöglicht. Wesentliche Aspekte der Lösungsmuster sind in Anlehnung an die Definition von ALEXANDER zu definieren [AIS+77]. Einmal erstellte Muster liefern eine generalisierte Problem-/Lösungsbeschreibung. Um diese effizient in den Entwicklungsprozess einbinden zu können, bietet es sich an, Wissensinhalte eines Musters zielgruppengerecht (z.B. durch Funktionen oder Systemelemente) darzustellen. Hierzu sind geeignete Techniken zur Spezifikation von komplexen Systemen und Systemzusammenhängen zu identifizieren und ggf. weiterzuentwickeln. Ferner müssen die Lösungsmuster die wesentlichen Arbeitsgebiete eines Systems Engineers – Systementwurf und anschließendes Management der gesamten Produktentwicklung – unterstüt-

zen. Es ist daher zu klären, wie Verbindungen von textbasierten Ansätzen wie z.B. Wirkprinzipien mit modellbasierten Ansätzen wie z.B. Lösungsmustern der Softwaretechnik gestaltet und transparent abgebildet werden können. In diesem Zusammenhang ist zu definieren, welche Dimensionen – Art der Wissensrepräsentation, Aggregation, Generalisierungsgrad, involvierte Fachdisziplinen – die Lösungsmuster für den Systementwurf charakterisieren. Gefordert ist eine multidimensionale Skalierung.

### **Handlungsfeld 3: Übergeordnete Systematik für den Einsatz von Lösungsmustern**

Eine übergeordnete Systematik für den Einsatz von Lösungsmustern im Systementwurf muss dem geschlossenen Wissensmanagementkreislauf (SECI) nach NONAKA/TAKAUCHI gerecht werden [NT97]. Zum einen muss gewährleistet sein, dass Lösungsmuster systematisch auf Basis etablierter (Teil-)Lösungen identifiziert werden können. Dies bedeutet, dass spezialisierte Lösungen abstrahiert, generalisiert und gemeinsam mit dem zugehörigen Problem dokumentiert werden müssen. Ferner muss die Kombination relevanten Lösungswissens, z.B. über die Konstruktion oder die Steuerungssoftware in multidisziplinäre Lösungsmuster, ermöglicht werden. Zum anderen ist sicherzustellen, dass sich das Vorgehen eines lösungsmusterbasierten Systementwurfs u.a. aus Akzeptanzgründen an gängigen Vorgehensweisen orientiert. Entwickler sollen befähigt werden, losgelöst von ihren Denkmustern, effizient unterschiedliche Probleme mit Hilfe von dokumentierten Lösungsmustern zu bewältigen und Innovationen hervorzubringen.

## **2.6 Anforderungen an die Arbeit**

Aus der Problemanalyse resultieren folgende Anforderungen an ein *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*:

**A1) Ganzheitlichkeit:** An der Entstehung mechatronischer Systeme sind neben den technischen Disziplinen wie Maschinenbau oder Softwaretechnik auch die strategische Produktplanung und die Produktionssystemplanung mit ihren unterschiedlichen Vertiefungsrichtungen beteiligt. Voraussetzung für eine übergreifende und barrierefreie Wiederverwendung von Wissen in Form von Lösungsmustern ist eine einheitliche Definition und ein ganzheitlicher Überblick über Lösungsmuster in der Produktentstehung (vgl. HF 1, Kap. 2.3.1. und 2.4.5).

**A2) Charakterisierung von Lösungsmustern:** Die Produktentstehung ist durch eine Vielzahl von Vernetzungen zwischen den beteiligten Disziplinen geprägt. Um die Abstimmung entlang dieser Vernetzung zu fördern, sind Beispielmuster sowie wechselseitige Zusammenhänge zu beschreiben. Dies gilt auf einer sehr abstrakten Ebene für die gesamte Produktentstehung und auf einer detaillierten Ebene für die Lösungsmuster im Hinblick auf den Systementwurf (vgl. HF 1 und 2, Kap. 2.3.1 und 2.4.5).

**A3) Einheitliche Strukturierung in Anlehnung an ALEXANDER:** Die Problemanalyse hat ergeben, dass Muster für die Produktentwicklung eine hohe Variation aufweisen. Um die langfristige Dokumentation sowie Kommunikation und Kooperation ausgehend

vom Systementwurf sicherstellen zu können, ist eine einheitliche Strukturierung der Lösungsmuster erforderlich. Dabei sind die Kategorien (Name, Problem, Lösung, Kontext) der Mustertheorie nach ALEXANDER aufzugreifen. Die Strukturierung stellt die Anwendung der Lösungsmuster ohne große Einarbeitung, z.B. durch das Hinzuziehen weiterer Literatur, sicher (vgl. HF 2, Kap. 2.4.4 und 2.4.5).

**A4) Interdisziplinarität und Verständlichkeit:** Ein Systems Engineer als Generalist koordiniert den gemeinsamen Entwurf intelligenter technischer Systeme mit den Entwicklern aus unterschiedlichen Disziplinen. Die adressierten Lösungsmuster sollen diesen Prozess durch Bereitstellung von Lösungswissen unterstützen. Vor diesem Hintergrund ist es essentiell, dass Techniken zur Spezifikation komplexer Systeme und damit für eine interdisziplinäre Wissensrepräsentation genutzt bzw. weiterentwickelt werden. Durch die Wahl bzw. Ausarbeitung geeigneter Partialmodelle ist sowohl das Problem als auch dessen Lösung eindeutig zu beschreiben. Ein gemeinsames Verständnis ermöglicht es den Entwicklern, fachdisziplin fremde Lösungsprinzipien zu adaptieren und einzusetzen (vgl. HF 2, Kap. 2.2.3 und 2.3.4).

**A5) Durchgängige Dokumentation:** Neben der Koordination des Systementwurfs zählt das Management des Entwicklungsgeschehens zu den Aufgaben eines Systems Engineers. Im Hinblick auf die Durchgängigkeit ist zu klären, wie die Zusammenhänge von textbasierten Ansätzen (z.B. Wirkprinzipien) und modellbasierten Ansätzen (Lösungsmuster der Software) sind. Es stellt sich die Frage nach unterschiedlichen Dimensionen, um Lösungsmuster für den Systementwurf zu ordnen. Unterschiedliche Ansätze zur Komplexitätsreduktion (elementare oder komplexe Muster [Suh93]) oder unterschiedlich verwendete Fachtermini erfordern eine multidimensionale Skalierung (vgl. HF 2, Kap. 2.3.1 und 2.4.5).

**A6) Systematische Vorgehensweise:** Ein lösungsmusterbasierter Systementwurf bedarf einer systematischen Vorgehensweise. Gleiches gilt für die Identifizierung von Lösungsmustern auf Basis etablierter (Teil-)Lösungen. Das Instrumentarium muss eine zielgerichtete und systematische Entwicklung neuer Lösungskonzepte mit Hilfe von Lösungsmustern ermöglichen. Die dafür notwendigen Vorgehensweisen sind durch entsprechende Bestandteile zu gestalten und nachvollziehbar zu beschreiben (vgl. HF 3, Kap. 2.3.2).

**A7) Geschlossener SECI-Kreislauf:** Die Problemanalyse verdeutlicht, dass gängige Wissensmanagementansätze sich am SECI-Kreislauf nach NONAKA/TAKEUCHI orientieren. Dies gilt auch für das angestrebte Instrumentarium. Es ist sicherzustellen, dass implizites Wissen externalisiert und somit geeignet zur Verfügung gestellt wird. Ferner müssen mit Hilfe dieses Wissens neue Lösungen und somit neues Wissen generiert werden können (vgl. HF 3, Kap. 2.4.2 und 2.4.3).

**A8) Abstraktion und Generalisierung:** Die Identifizierung und Dokumentation von Lösungsmustern basiert auf etablierten (Teil-)Lösungen und damit verbundener Expertise aus allen an der Entwicklung beteiligten Experten. Relevantes Wissen ist in der Re-

gel repräsentiert durch Entwicklungsdokumente und/oder Modelle aus den unterschiedlichen Fachdisziplinen (z.B. CAD-Modell des Maschinenbaus oder Funktionsablaufplan der Automatisierungstechnik). Da die beschriebenen Dokumente oder Modelle jedoch konkrete Lösungen beschreiben, sind diese in geeigneter Art und Weise zu abstrahieren bzw. zu generalisieren. Damit wird die Übertragbarkeit auf zukünftige Entwicklungsaufgaben, z.B. von cyber-physischen Systemen, ermöglicht (vgl. HF 3, Kap. 2.2.3 und 2.4.2).

**A9) Orientierung an etablierten Standards im Systementwurf:** Die Problemanalyse zeigt, dass in der Verschmelzung etablierter Entwurfsmethoden oder Denkweisen aus der Entwicklung mechatronischer Systeme und dem Systems Engineering der Stellhebel liegt, zukünftig effizient und effektiv komplexe technische Systeme zu entwickeln. Übereinstimmend wird beschrieben, wie, ausgehend von der lösungsneutralen Sicht, unterschiedliche Konzepte modellbasiert entworfen, ausgewählt und konkretisiert werden (von den Anforderungen über die Funktionen hin zur Struktur und dem Verhalten). Es ist daher sicherzustellen, dass sich das Vorgehen an diesen etablierten Standards orientiert (vgl. HF 3, Kap. 2.2.3, 2.3.2 und 2.3.4).



### **3 Stand der Technik**

Dieses Kapitel gibt einen Überblick über den Stand der Technik. Die Problemanalyse hat gezeigt, dass eine umfassende Beschreibung des Paradigmas der Lösungsmuster in der Produktentstehung fehlt. Vor diesem Hintergrund liefert Kapitel 3.1 eine Untersuchung von Ansätzen, die ähnliche Beschreibungen oder Klassifikationen von Lösungsmustern beinhalten. Schwerpunkt des Instrumentariums ist Lösungswissen für den Systementwurf. Aus diesem Grund werden in Kapitel 3.2 Techniken für eine geeignete Systemspezifikation und somit Wissensrepräsentation in Lösungsmustern analysiert. Hinzu kommt die Untersuchung sowohl fachdisziplinübergreifender als auch fachdisziplinspezifischer Lösungsmusteransätze. Etablierte Ansätze für den Einsatz von Lösungsmustern in einer übergeordneten Systematik analysiert Kapitel 3.3. Hierbei geht es sowohl um die Identifizierung bestehender Lösungsmuster aus etablierten Lösungen, als auch um Ansätze für einen lösungsmusterbasierten Systementwurf. Komplettiert wird der Stand der Technik durch eine Bewertung der Ansätze anhand der Anforderungen aus Kapitel 2. Die gewählte Reihenfolge der untersuchten Ansätze stellt dabei keine Bewertung dar.

#### **3.1 Überblick – Lösungsmuster in der Produktentstehung**

Eine umfassende Übertragung des Paradigmas der Lösungsmuster auf die Produktentstehung mit Beispielmustern und einer Beschreibung von wechselseitigen Abhängigkeiten existiert nicht. Jedoch gibt es Ansätze, die sich mit bestimmten Ausschnitten beschäftigen. Aus diesem Grund liefert Kapitel 3.1.1 eine Beschreibung der Universal Design Theory nach GRABOWSKI mit Objekt- und Prozessmustern. In Kapitel 3.1.2 wird die Musterhierarchie nach CLOUTIER mit ihrem Ursprung im Systems Engineering beschrieben. Abschließend werden in Kapitel 3.1.3 und 3.1.4 übergeordnete Ansätze analysiert, die vorrangig mögliche Wechselwirkungen zwischen Lösungsmustern beschreiben.

##### **3.1.1 Universal Design Theory nach GRABOWSKI**

Ziel der Universal Design Theorie (UDT) ist die Zusammenführung von verschiedenen Ingenieurdisziplinen mit weiteren Disziplinen wie Materialwissenschaften, Chemie oder Pharmazie. Die Theorie fußt auf der Überlegung, Kenntnisse und Wissen aus verschiedenen Disziplinen konsistent, zusammenhängend und einheitlich wiederzugeben und zu nutzen – in Form von Mustern. Im Kern der Methodik steht die Erarbeitung einer Basis für die interdisziplinäre Produktentwicklung [GL00].

Die Theorie basiert auf folgenden Annahmen: Problemlösungen basieren immer auf der Erfahrung und dem Wissen einer bestimmten Disziplin; komplexe Problemstellungen lassen sich in Teilprobleme zerlegen.

Folglich hat jede Disziplin spezialisiertes Lösungswissen, das auf bestimmte Probleme angewandt werden kann. Ziel ist die Übertragbarkeit dieses Lösungswissen auf fachfremde Bereiche. Es soll die Möglichkeit geschaffen werden, bei einem Problem ohne sichtbare Lösung in dem jeweiligen Fachbereich, auf Lösungswissen aus einem anderen Fachbereich zurückgreifen zu können [GL00]. Dieses generalisierte Lösungswissen soll in Form von Lösungsmustern repräsentiert werden.

Für GRABOWSKI ET AL. entsteht ein Lösungsmuster aus Erfahrungswissen von Experten, das sich in konkreten Lösungen manifestiert hat. Nutzen entsteht genau dann, wenn das Wissen über das Problem sowie den Problemlösungsweg nachvollziehbar dokumentiert ist. Lösungsmuster helfen, einen Ausgangszustand in einen Folge- bzw. Zielzustand zu überführen. Wesentliche prozessrelevante Merkmale sind ein hoher Kreativitätsanteil, hoher Vernetzungsgrad der Vorgänge, unscharfe Informationen, hohe Entscheidungsintensität, geringer Formalisierungsgrad sowie geringe Wiederholhäufigkeit. Grundlage zur Darstellung eines solchen Prozesses ist die Beschreibung seiner Zustände sowie die zugehörigen Zustandsänderungen [GL03]. Wie bereits in Kapitel 2.4.5 geschildert, wird in der UDT zwischen Objekt- und Prozessmustern unterschieden. Bild 3-1 zeigt den Zusammenhang von Objekt- und Prozessmustern.

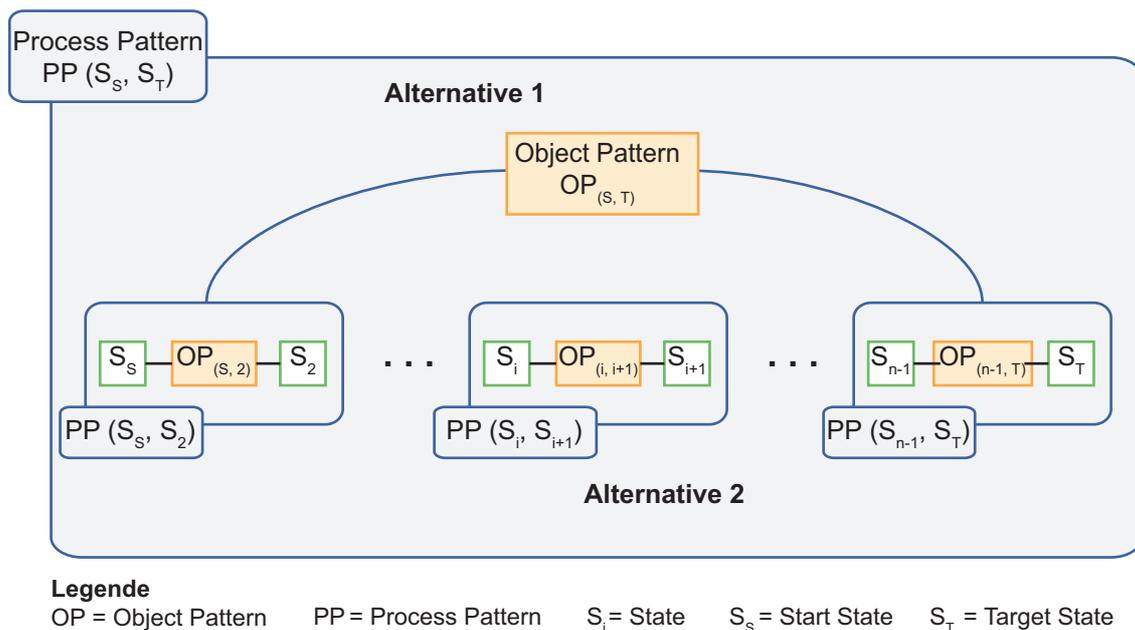


Bild 3-1: Zusammenhang von Objekt- und Prozessmustern in der UDT [vgl. GL00]

Es wird deutlich, dass es sich bei einem Objektmuster um eine statische Lösung handelt und bei einem Prozessmuster um einen in der Zeit ablaufenden Vorgang. Der Ablauf eines Entwicklungsprozesses unterscheidet sich in der Kaskadierung von Objekt- und Prozessmustern. Beispielhaft sind in Bild 3-1 zwei Alternativen für den Entwicklungsprozess dargestellt. Der Ausgangszustand ( $S_S$ , start state) beinhaltet alle Informationen über die Problemstellung. Durch Objekt- und Prozessmuster werden Zwischenzustände ( $S_i$ , intermediate states) und zuletzt der Zielzustand erreicht ( $S_T$ , target state). Sind die

Voraussetzungen bekannt und es gibt bereits eine entwickelte Lösung, dann ist der Zielzustand direkt über bekannte Objektmuster  $OP_{(S,T)}$  und Prozessmuster  $PP(S_S, S_T)$  zu erreichen (Alternative 1). Gibt es hingegen keine bekannten, passenden Objekt- und Prozessmuster, so ist die Expertise der Entwickler gefragt, um den Zielzustand sukzessive zu erreichen. In diesem Entwicklungsprozess werden stufenweise Prozessmuster mit unterschiedlicher Komplexität durchlaufen, wodurch verschiedene Wege aufgrund verschiedener Objektmuster zum Ziel führen (Alternative 2) [GL00].

Ferner definiert GRABOWSKI die folgenden fünf Hypothesen für die Unterteilung und die Anwendung von Lösungsmustern [GL00]:

- **Hypothese 1 (Hypothesis of finite basic elements):** Auf jeder Abstraktionsebene existiert eine endliche Anzahl von Basiselementen.
- **Hypothese 2 (Hypothesis of finite abstraction levels):** Es existiert eine endliche Anzahl von Abstraktionsebenen, um Produkte zu modellieren und Design-Prozesse zu beschreiben.
- **Hypothese 3 (Hypothesis of finite transitions):** Die Anzahl möglicher Übergänge zwischen den einzelnen Abstraktionsebenen ist endlich. Dabei entsteht die Verknüpfung der Hierarchieebenen durch die Verbindungen zwischen den Basiselementen.
- **Hypothese 4 (Hypothesis of invention):** Neue Produkte entstehen durch neue Kombinationen bekannter Basiselemente. Beispiele derartiger Basiselemente sind Funktionen oder physikalische Effekte.
- **Hypothese 5 (Hypothesis of solution finding):** Jede Produkthanforderung bezieht sich auf einen Lösungsbereich. Die Ableitung einer Gesamtlösung ist nur dann widerspruchsfrei, wenn die Anforderungen und deren Lösungen konsistent und verträglich kombiniert wurden.

**Bewertung:** Die Universal Design Theorie enthält ein sehr abstraktes und theoretisches Modell zur Erklärung bestehender Zusammenhänge in der Entwicklung technischer Systeme mit Hilfe von Mustern. GRABOWSKI ET AL. treffen die Unterscheidung zwischen Objekt- und Prozessmuster, wobei eine starke Abhängigkeit zwischen den Musterarten betont wird. In der UDT werden die Überlegungen jedoch nicht auf strategische Bereiche in der Produktentstehung ausgedehnt. Zudem mangelt es an nachvollziehbaren Beispielen, was eine direkte Übertragung dieses theoretischen Modells auf die im Instrumentarium geforderten Anforderungen ausschließt.

### 3.1.2 Musterhierarchie nach CLOUTIER

Im Gegensatz zu den eingangs geschilderten Überlegungen von GRABOWSKI ET AL. wird der Einsatz von Lösungsmustern im Systems Engineering deutlich bereichsübergreifender interpretiert. Dies bezieht sich weniger auf die Art der Wissensrepräsentation in den

Mustern, sondern vielmehr auf ihren Einsatz in vielen an der Produktentstehung beteiligten Bereichen. Dies verdeutlicht die Musterhierarchie nach CLOUTIER (Bild 3-2), in der auch nicht technische Bereiche berücksichtigt werden [CD06].

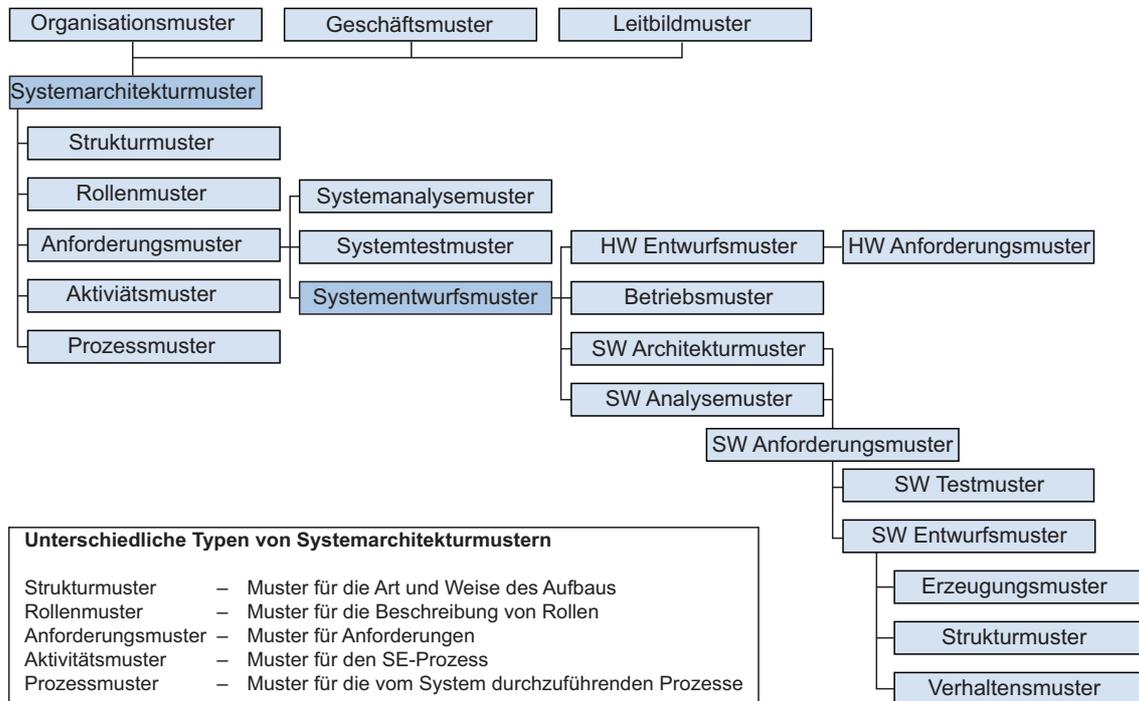


Bild 3-2: Musterhierarchie nach CLOUTIER [vgl. CD06]

Muster auf höchster Ebene sind demzufolge Organisations-, Geschäfts- und Leitbildmuster. Eine Hierarchiestufe tiefer nennt CLOUTIER Systemarchitekturmuster, von denen es fünf unterschiedliche gibt: Strukturmuster (Strukturierungskonzepte z.B. Differentialbauweise), Rollenmuster (charakteristische Rollen beteiligter Akteure), Anforderungsmuster (wiederkehrende Anforderungsbündel), Aktivitätsmuster (SE-Prozessbausteine) und Prozessmuster (Verhalten des Systems). Unter den Anforderungsmustern folgen Muster auf Gesamtsystemebene. Diese konzentrieren sich auf Systementwurf, -analyse und -test. An die Systementwurfsmuster schließen sich noch weitere Mustertypen an, die in die fachdisziplinspezifischen Bereiche der Hardware- und Softwareentwicklung unterteilt werden. Als zentrale Mustertypen im Hinblick auf eine signifikante Effizienzsteigerung in der Entwicklung komplexer technischer Systeme nennt CLOUTIER Systemarchitektur- und Systementwurfsmuster (vgl. Bild 3-2, farblich hervorgehoben).

**Bewertung:** Die Musterhierarchie liefert einen ersten Ansatz für eine breit aufgestellte Übertragung des Lösungsmusteransatzes auf die Entwicklung komplexer technischer Systeme. Es stellt sich jedoch die Frage, wie die genannten Muster konkret aussehen und wie diese zueinander in Beziehung stehen. Hierzu werden außer der Titulierung der Muster keine näheren Angaben gemacht. Die konsequente Weitergabe von Informationen und die Verzahnung der Muster über die Hierarchiestufen hinweg ist jedoch Grundvoraussetzung für deren durchgängige Nutzung.

### 3.1.3 SE Entwurfsmuster-Metamodell nach PFISTER

Nach PFISTER eignen sich Lösungsmuster in erster Linie zur Abstraktion und Generalisierung individueller Lösungen, die in einer übergeordneten Gesamtarchitektur eingebettet sind. Ein Lösungsmuster bietet die Möglichkeit, Expertise von Fachleuten Dritten zur Verfügung zu stellen. Es unterstützt die Anwender in der Identifizierung und Lösung von Problemen durch die Übertragung des bestehenden Wissens bzw. der Erfahrung auf das jeweils vorliegende Problem. PFISTER nennt als Ziele von Lösungsmustern [PCH+11]:

- Leistungssteigerung (Umfang, Relevanz)
- Verbesserung der Funktionssicherheit (etablierte Lösungen, mehrmals kontextbezogen eingesetzt und geprüft)
- Effizienzsteigerung (Zeitersparnis von der Produktidee bis zum Serienanlauf)
- Förderung von gemeinschaftlichem Arbeiten durch Austausch von Wissen auf einer gemeinsamen Plattform

Diese Ziele können erreicht werden, indem relevante Expertise, basierend auf implementierten Lösungen oder Erkenntnissen aus lessons-learned-Prozessen, formalisiert abgelegt werden. Lösungsmuster sind folglich ein Mittel, um weiteren Entwicklern standardisierte Lösungen zugänglich zu machen.

PFISTER empfiehlt die Anwendung von Lösungsmustern im Systems Engineering bei der Synthese von funktionalen und physikalischen Architekturen. In diesem Zusammenhang fungiert ein Lösungsmuster als ein kleines und gut handhabbares Artefakt, das selten isoliert sondern im Zusammenhang mit weiteren betrachtet werden sollte. Das relevante Lösungswissen in einem Lösungsmuster wird mit Hilfe abstrakter physikalischer oder funktionaler Modelle repräsentiert. Es definiert das Zusammenwirken von Systemkomponenten oder -funktionen in generalisierter Art und Weise. Nach PFISTER ist ein Lösungsmuster keine Kreativitätstechnik und beschreibt auch keine wiederverwendbare detaillierte Komponente. In erster Linie ist es das Ergebnis eines mehrmals vorkommenden Lösungsansatzes in ähnlichen Problemstellungen. PFISTER betont, dass Lösungsmuster genau dann zu Innovationen führen, wenn sie auf weitere Problemstellungen in einem ähnlichen Kontext übertragen und geeignet modifiziert werden [PCH+11].

Im Hinblick auf einen modellbasierten Systementwurf sollten Lösungsmuster durch eine formalisierte Sprache beschrieben und in einer Datenbank abgelegt sein. Ferner ist sicherzustellen, dass die Modelle in eine Entwicklungsumgebung eingebunden werden können. Lösungsmuster und ihr Zusammenwirken lassen sich durch Sprachen wie SysML, UML, STEP oder auch die Web Ontology Language (OWL) beschreiben. PFISTER definiert Lösungsmuster in UML oder SysML in Form parametrisierbarer Beziehungen. Hierzu werden unterschiedliche Klassen und Objekte definiert. Diese besitzen

spezifische Rollen und Beziehungen untereinander. Die Lösungsmuster werden in ein übergeordnetes Metamodell für das Systems Engineering eingebettet. Das entsprechende UML-Diagramm<sup>33</sup> ist in Bild 3-3 abgebildet.

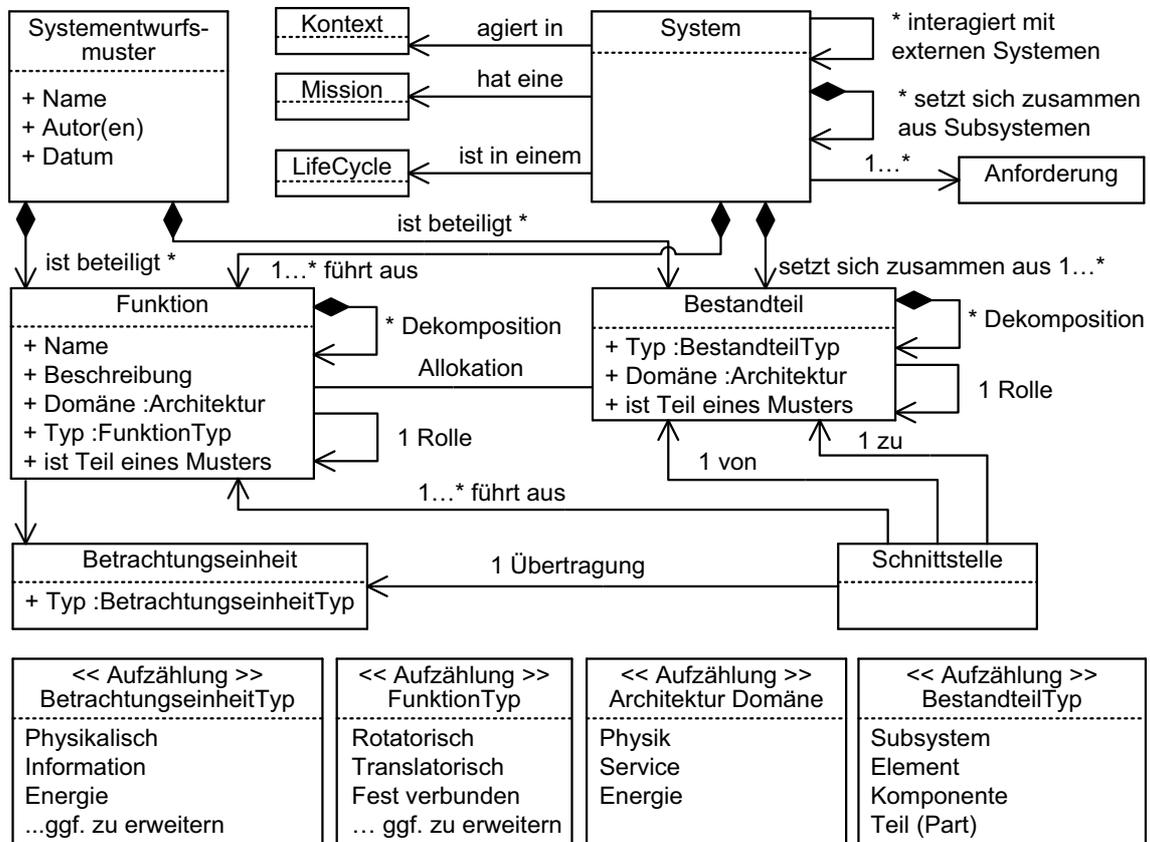


Bild 3-3: Metamodell für ein lösungsmusterbasiertes Systems Engineering nach PFISTER [vgl. PCH+11]

Es basiert auf den Überlegungen von CLOUTIER (vgl. Kapitel 3.1.2), beschreibt jedoch im direkten Vergleich die Beziehungen und Ausprägungen der Klassen und Objekte in einem höheren Formalisierungsgrad. Die Lösungsmuster für Systems Engineering („Systementwurfsmuster“) sind durch Angaben wie „Lösungsmustername, Autor und Datum“ zu spezifizieren und verfügen in dem Metamodell über direkte Beziehungen zu den Klassen „Funktion“ und „Bestandteil“. Die Lösungsmuster sind daher indirekt mit der Klasse „System“ (technisches System) verbunden, das Beziehungen zu weiteren Klassen und zu sich selbst besitzt. Innerhalb einzelner Klassen ist an entsprechender Stelle auf bestimmte Untergruppen bzw. Typen verwiesen, für die jeweils Auswahllisten existieren bzw. zu definieren sind. Ein Beispiel ist die Klasse „Bestandteil“ mit dem Typ „:BestandteilTyp“ bestehend aus Subsystem, Element, Komponenten und Teil (Part). Das Metamodel hat primär den Anspruch, Lösungsmuster in das übergeordnete Systems Engineering formal einzubetten. Eine detailliertere Spezifikation der Lösungs-

<sup>33</sup>Für eine ausführliche Beschreibung der Notation UML siehe Kapitel 3.2.1.2.

muster, auch mit Beziehungen zu weiteren Mustern, beschreibt PFISTER mit dem Meta-modell in Bild 3-4 [PCH+11].

Das Lösungsmuster-Metamodell spezifiziert PFISTER ebenfalls formal mit der UML. Mögliche Beziehungsformen zu weiteren Lösungsmustern sind: Anti-Muster, Äquivalente Muster, Verwandte Muster oder zusätzlich erforderliche Muster. Neben den zuvor bereits genannten Informationen in einem Muster wie Name, Autor und Datum, gibt es weitere Beziehungen, die ein Lösungsmuster beschreiben. Dazu zählen u.a. Angaben über: die zugrundeliegenden Anforderungen, die das Lösungsmuster erfüllt; Applikationen, in denen es erfolgreich eingesetzt wurde; eine Begründung; Pseudonyme oder Schlüsselwörter, die bei der Suche nach Lösungsmustern in einer Datenbank hilfreich sein können [PCH+11].

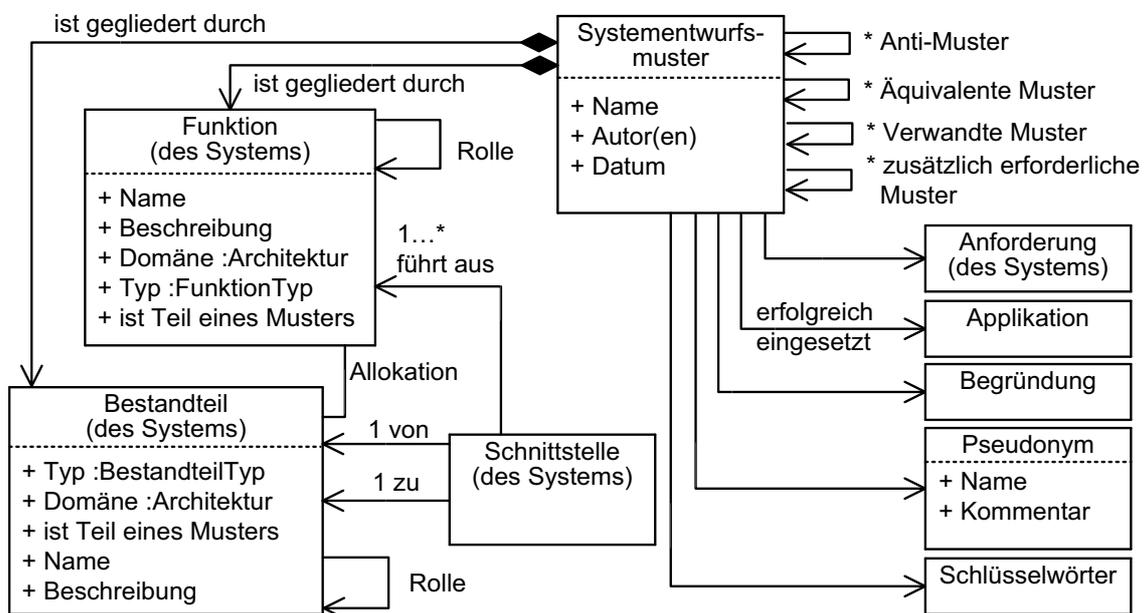


Bild 3-4: Systementwurfsmuster-Metamodell nach PFISTER [vgl. PCH+11]

Lösungsmuster lassen sich mit Hilfe von Funktionen und Bestandteilen des Systems gliedern. PFISTER betont hierbei die Notwendigkeit der modellbasierten Repräsentation. Es ist entscheidend, dass die Problembeschreibung (Funktionen und Anforderungen) durch Modelle abgebildet wird. Gleiches gilt für die Lösungsbeschreibung in Form von generalisierten und abstrakten Bestandteilen des Systems. Das in Bild 3-4 abgebildete Metamodell erlaubt dabei den Einsatz unterschiedlichster Modellierungssprachen. Je nach Einsatzzweck eignen sich Sprachen für die jeweiligen Aspekte wie Dynamik, Funktionen, Verhalten, Struktur etc. [PCH+11].

**Bewertung:** PFISTER stützt sich in seinen Arbeiten auf die Überlegungen zur Musterhierarchie nach CLOUTIER. Er beschreibt mit Hilfe der Modellierungssprache UML weniger die unterschiedlichen Arten von Lösungsmustern wie Prozess- oder Testmuster, sondern vielmehr die allgemeine Einbettung der Entwurfsmuster ins Systems Engineering. PFISTER beschränkt sich dabei auf den Entwurf des technischen Systems. Vor- und

nachgelagerte Bereiche im Sinne der Produktentstehung nach GAUSEMEIER finden keine Berücksichtigung. Dieses wäre aufgrund des hohen Formalisierungsgrads in der gewählten Detailtiefe auch sehr aufwändig. Im Hinblick auf die direkte Implementierung des Lösungsmusteransatzes sind die Ergebnisse von PFISTER sehr hilfreich. Es mangelt jedoch an einem allgemeinverständlichen Überblick über Lösungsmuster in der PE, nicht zuletzt durch das Fehlen von Beispielen.

### 3.1.4 SE Mustersprache nach SIMPSON

Nach SIMPSON eignen sich Lösungsmuster in erster Linie zur langfristigen Speicherung von relevantem Lösungswissen über komplexe Sachverhalte. Unter einem Lösungsmuster versteht SIMPSON die Lösung eines bestimmten Problems innerhalb eines bestimmten Kontextes. Die speziell entwickelte Mustersprache basiert auf Lösungsmustern für das Systems Engineering aus der Literatur und weiteren identifizierten Mustern von SIMPSON. Eine Mustersprache wird beschrieben als eine Sammlung von Mustern und die Beschreibung von wechselseitigen Beziehungen. SIMPSON fasst auch grundsätzliche Überlegungen und Gliederungsansätze im Systems Engineering als Muster auf. Dazu gehören u.a. die drei abstrakten und zudem global anzuwendenden Lösungsmuster<sup>34</sup> [SS06]:

- Es kann alles in Form eines “Systems” beschrieben werden
- Das “Problem-System” ist immer getrennt vom “Lösungs-System”
- Es sind immer “mindestens drei Systeme” gemeinsam zu betrachten:  
Umfeld-System, Produkt-System, Produktionsprozess-System

SIMPSON betont, dass es zahlreiche Beschreibungen gibt, die die globalen Lösungsmuster und deren Beziehungen untereinander weiter verfeinern und detailliert spezifizieren. Dennoch bilden die genannten Lösungsmuster die Grundlage für eine initiale Ontologie. Ziel ist ein übergeordnetes Rahmenwerk, in dem diverse Lösungsmuster der am Systems Engineering beteiligten Bereiche gespeichert, diskutiert und in Zusammenhang zueinander gesetzt werden können. Das Resultat eines mit Lösungsmustern gefüllten Rahmenwerks wäre eine Systems-Engineering-Mustersprache. Zur näheren Erläuterung werden im Folgenden Ansätze für die Mustersprache vorgestellt [SS06].

#### Formen von Systems Engineering Mustersprachen:

Mustersprachen setzen sich nach SIMPSON aus genau zwei Komponenten zusammen: Mustern und den Beziehungen zwischen den Mustern. Die Beziehungen können dabei sehr unterschiedlich sein, sollten jedoch alle das Ziel verfolgen, eine logische und verträgliche Kombination von Lösungsmustern zu ermöglichen. SIMPSON nennt zwei unterschiedliche Arten zur Darstellung der Beziehungen in einer Mustersprache: *Concept*

---

<sup>34</sup> Für weitere Details zu den Lösungsmustern sei auf die Originale im Anhang A1.1 verwiesen.

*Map Style* und *Sequential Concept Map Style*. Er betont dabei die Wichtigkeit einer grafischen Repräsentation zur Beherrschung der zugrundeliegenden Komplexität durch die Anwender.

**Concept Map Style:** Grundgedanke dieser Repräsentationsform für eine Mustersprache ist eine Art Karte. Es handelt sich folglich um eine grafische Repräsentation von Lösungsmustern und deren Beziehungen. Es werden unterschiedliche Beziehungsformen definiert und die Lösungsmuster so in Beziehung zueinander gesetzt.<sup>35</sup>

**Sequential Concept Map Style:** Diese Darstellungsart baut auf der zuvor geschilderten Concept Map Style auf. Neben Querverweisen wie “in Kombination verträglich” oder “in Kombination nicht verträglich” spielt der zeitliche Aspekt eine wesentliche Rolle. Für die Abbildung zeitlicher Zusammenhänge schlägt Simpson drei unterschiedliche Herangehensweisen vor, die individuell mit Lösungsmustern ausgestaltet werden müssen: „Top-Down“; „Center-Out“ oder sequentiell.

**Bewertung:** Als Grundlage für die Mustersprache definiert SIMPSON fundamentale Lösungsmuster im Gesamtkontext System Engineering. Es sind in erster Linie Gliederungsansätze wie die Differenzierung von einem Problem- und einem Lösungssystem. Es handelt sich um sehr allgemeine Gedankengänge, die sehr abstrakt ohne konkrete Beispiele präsentiert werden. Gleiches gilt für die Überlegungen zu der Mustersprache für das Systems Engineering. SIMPSON orientiert sich zwar an gängigen Definitionen für eine Mustersprache, es mangelt jedoch an Beispielmustern und an einer detaillierten Spezifikation der Beziehungsformen. Es werden grafische Darstellungsarten für die Mustersprache vorgeschlagen, eine konkrete Ausprägung bleibt aber aus.

## 3.2 Lösungswissen für den Systementwurf

Um zeit- und kostenintensive Änderungen entlang einer Produktentwicklung so gering wie möglich zu halten, ist eine fachdisziplinübergreifende Systemkonzipierung unerlässlich. Bei der Festlegung des Systemkonzepts im Rahmen des Systementwurfs sind Entwickler aller beteiligten Disziplinen zusammenzuführen. Die Problemanalyse hat gezeigt, dass neben ihrer fachspezifischen Expertise auch Generalisten erforderlich sind, die das Gesamtsystem und relevante Schnittstellen im Blick behalten. Neben einer einheitlichen Terminologie, die die Plattform für eine barrierefreie Kommunikation bildet, bedarf es gleichermaßen einer einheitlichen Aufbereitung und Strukturierung von Lösungswissen aus den unterschiedlichen Disziplinen. Vor diesem Hintergrund werden im Kapitel 3.2.1 Spezifikationstechniken für die Systemmodellierung und somit für die multidisziplinäre Wissensrepräsentation in den Lösungsmustern betrachtet. Kapitel 3.2.2 untersucht fachdisziplinübergreifende Lösungsmuster, die vorrangig an die Gene-

---

<sup>35</sup> Es handelt sich im übertragenen Sinne um eine Art Mind-Map mit Lösungsmustern als Kernelement.

ralisten adressiert sind. Diese müssen nahtlos mit den fachdisziplinspezifischen Ansätzen verbunden sein, die abschließend in Kapitel 3.2.3 analysiert werden.

### 3.2.1 Techniken für die Wissensrepräsentation

Relevantes Expertenwissen für den Systementwurf, das in den Lösungsmustern abgebildet werden soll, bedarf geeigneter Techniken für die Spezifikation und Repräsentation, d.h. Sprachen zur Darstellung von Artefakten und zugehörigem Wissen. Die Problemanalyse hat ergeben, dass das Wissen nach Möglichkeit in der gleichen Notation bzw. Beschreibungssprache abgebildet werden muss, die zum jeweiligen Zeitpunkt in der Entwicklung genutzt wird. Aus diesem Grund werden in den folgenden Kapiteln 3.2.1.1 bis 3.2.1.3 sowohl die semiformale Spezifikationstechnik CONSENS, als auch formale Spezifikationstechniken wie UML und SysML vorgestellt. Zusätzlich werden in Kapitel 3.2.1.4 und 3.2.1.5 die Sprachen PrEMISE und Modelica<sup>®</sup> analysiert.

#### 3.2.1.1 CONSENS

Am Heinz Nixdorf Institut Paderborn wurde im Rahmen des SFB 614 die Spezifikationstechnik CONSENS (**CON**ceptual design **Spec**ification technique for the **EN**gineering of complex **S**ystems) zur Beschreibung der Prinziplösung fortgeschrittener mechatronischer Systeme entwickelt [GFD+08a], [GFD+08b], [DDG+14]. CONSENS umfasst eine Modellierungssprache und eine Vorgehensweise, die im Rahmen des Systementwurfs eingesetzt wird. Die Modellierungssprache gliedert sich in die Aspekte Umfeld, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur, Verhalten und Gestalt (vgl. Bild 3-5). Diese Aspekte werden im Rahmen der Entwicklung rechnerinterner durch Partialmodelle repräsentiert und im Folgenden näher erläutert. An dieser Stelle sei hervorgehoben, dass wechselseitige Beziehungen zwischen den Partialmodellen eine zentrale Rolle einnehmen. Die partialmodellübergreifenden Beziehungen werden zwischen den Konstrukten der Partialmodelle (z.B. Funktion und Systemelement) gebildet. Die sieben ausgearbeiteten Partialmodelle bilden die Ausgangsbasis für die anschließende fachdisziplinspezifische Konkretisierung.

**Umfeld:** Dieses Partialmodell beschreibt das Umfeld sowie dessen Interaktion mit dem zu entwickelnden System. Dabei werden Einflussbereiche wie übergeordnete Systeme oder Einflüsse wie Wärmestrahlung oder Windkraft aufgezeichnet. Zusätzlich findet eine Untersuchung möglicher Wechselwirkungen einzelner Einflüsse statt. In diesem Zusammenhang ist eine Situation eine konsistente Menge von gemeinsam auftretenden Einflüssen. Ferner sind Ausprägungen von Einflüssen, die eine Zustandsänderung hervorrufen, als Ereignis zu kennzeichnen.

**Anwendungsszenarien:** Anwendungsszenarien enthalten für einen bestimmten Fall das zu lösende Problem und deren Lösung. Sie beschreiben dabei in welcher Art und Weise sich das System in einem Zustand oder einer bestimmten Situation verhalten soll. Zu-

sätzlich enthalten sie Angaben, durch welche Ereignisse Zustandsübergänge stattfinden sollen.

**Anforderungen:** Sämtliche Anforderungen an das zu entwickelnde System werden in einer Anforderungsliste zusammengetragen. Diese bildet die Grundlage der späteren Entwicklung und ist fortlaufend zu pflegen. Die Anforderungen werden dabei verbal beschrieben und ggf. durch Attribute und deren Ausprägungen konkretisiert. Mögliche Hilfestellung beim Aufstellen einer Anforderungsliste liefern PAHL/BEITZ in Form von Checklisten<sup>36</sup> [PBF+07].

**Funktionen:** Eine Funktion beschreibt den Zusammenhang zwischen einer Eingangs- und einer Ausgangsgröße. Sie verfolgt dabei das Ziel, eine spezifische Aufgabe zu erfüllen [PBF+07]. Im Rahmen der Entwicklung erfolgt die Darstellung der Funktionen hierarchisch. Die Unterteilung der übergeordneten Funktionen in Subfunktionen findet solange statt, bis sich für die Funktionen sinnvolle Lösungen finden lassen. Funktionen sind demnach die Grundlage für die Anwendung von Lösungsmustern.

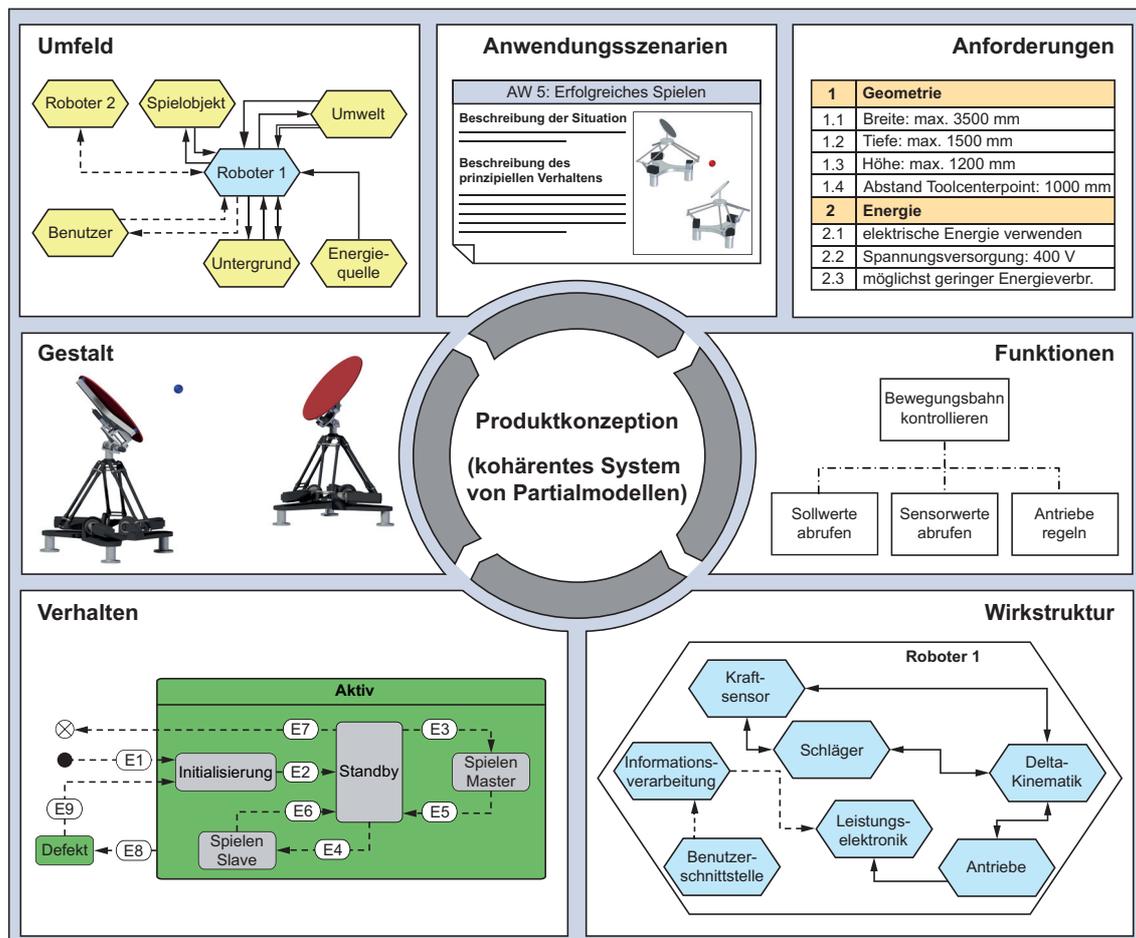


Bild 3-5: Partialmodelle der Spezifikationstechnik CONSENS [GTS14, S. 38]

<sup>36</sup>Für eine detaillierte Beschreibung sei zusätzlich auf [Rot01] und [Ehr03] verwiesen.

**Wirkstruktur:** Mit der Wirkstruktur werden die Systemelemente und deren Beziehungen im System repräsentiert. Sie bildet die gesamte Struktur fortgeschrittener mechatronischer Systeme feingliedert ab. Zur besseren Übersichtlichkeit lassen sich mehrere Systemelemente dabei zu übergeordneten logischen Gruppen zusammenfassen.

**Verhalten:** Das Partialmodell Verhalten lässt sich grundlegend in die drei unterschiedlichen Arten Zustände, Aktivitäten und Sequenz gliedern. Das Modell Verhalten – Zustände bildet dabei sämtliche vorausgedachten und zu berücksichtigenden Systemzustände und Zustandsübergänge ab. Die dabei auftretenden Ablaufprozesse werden in einem weiteren Modell, Verhalten – Aktivitäten, berücksichtigt. Das Partialmodell Verhalten – Sequenz bildet die Wechselwirkungen zwischen mehreren Systemelementen ab. In einer chronologischen Reihenfolge werden dabei die Aktivitäten der interagierenden Systemelemente modelliert.

**Gestalt:** Die Gestaltmodelle werden in der Regel mittels CAD-Systemen erstellt. Diese Modelle sind bereits im Rahmen der Konzipierung zu erarbeiten, da schon frühzeitig die Gestalt des Systems festzulegen ist. Dies betrifft insbesondere Wirkflächen, Wirkorte, Hüllflächen und Stützstrukturen.

**Bewertung:** Die Spezifikationstechnik CONSENS bildet mit den beschriebenen Aspekten und Konstrukten in vielerlei Hinsicht eine gute Ausgangsbasis. Durch die allgemeinverständliche Beschreibung komplexer mechatronischer Zusammenhänge eignet sich diese Technik bei der Erstellung und Anwendung z.T. fachdisziplinübergreifender Lösungsmuster. Es ist zu klären, wie die Zusammenhänge zu zeitlich vor- und nachgelagerten Lösungsansätzen sind. So stellt sich z.B. die Frage, ob und in welchem Umfang die Erkenntnisse aus den Lösungsmustern der Softwaretechnik im Rahmen des Systementwurfs abgebildet und somit berücksichtigt werden können. Es ist zu klären, ob die Spezifikationstechnik CONSENS ggf. modifiziert oder erweitert werden muss.

### 3.2.1.2 UML – Unified Modeling Language

UML (Unified Modeling Language) ist eine grafische Sprache, die im Rahmen der Softwareentwicklung – von der Modellierung bis hin zur Analyse von Softwareprogrammen – angewendet wird<sup>37</sup>. Die Sprache ermöglicht, Strukturen, Architekturen, Verhalten von Systemen sowie Interaktionen mit weiteren Systemen durch Diagramme abzubilden [For07], [JRH+04], [Oes05], [MHH+03]. Wesentliche Bestandteile dieser Diagramme sind **Objekte** und **Klassen**. Sie beinhalten spezifische **Attribute** und **Methoden** und bilden so die Grundlage dieser Modellierungssprache. Die Basismodelle (vgl. Bild 3-6) werden im Folgenden näher erläutert.

---

<sup>37</sup>Die UML wird von der Object Management Group (OMG) entwickelt und ist aktuell in der Version 2.4.1 in der ISO/IEC 19505-1 und ISO/IEC 19505-2 standardisiert [ISO19505-1], [ISO19505-2].

**Objekt:** Ein Objekt ist im Allgemeinen die grafische Repräsentation eines Gegenstandes des Interesses, insbesondere einer Beobachtung, Untersuchung oder Messung. Objekte können Modellierungen von realen Sachverhalten, Dingen oder Begriffen sein (vgl. Bild 3-7, Modellierung einer Person im Kontext einer Universität). Sie besitzen bestimmte Eigenschaften und reagieren mit einem vorgegebenen Verhalten auf definierte Anfragen. Außerdem besitzt jedes Objekt eine Identität, die es von allen anderen Objekten unterscheidet.

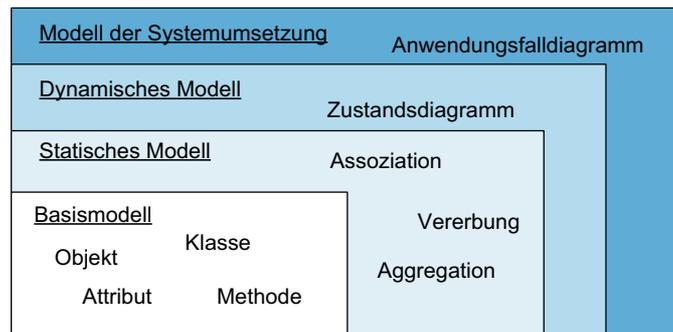


Bild 3-6: Gliederung des objektorientierten Modellverständnisses [vgl. For07, S. 21]

**Klasse:** Eine Klasse beschreibt eine Sammlung von Objekten mit gleichen Eigenschaften (Attributen), gemeinsamer Funktionalität (Methoden), gemeinsamen Beziehungen zu anderen Objekten und gemeinsamer Semantik.

**Attribut:** Die Attribute beschreiben die Eigenschaften eines Objektes bzw. einer Klasse. Alle Objekte einer Klasse besitzen dieselben Attribute, jedoch unterschiedliche Attributwerte.

**Methode:** Eine Methode ist ein Algorithmus. Dieser ist jedem Objekt zugeordnet und kann von diesem abgearbeitet werden. Das Verhalten eines Objektes wird durch eine bestimmte Menge von Methoden ausgedrückt.

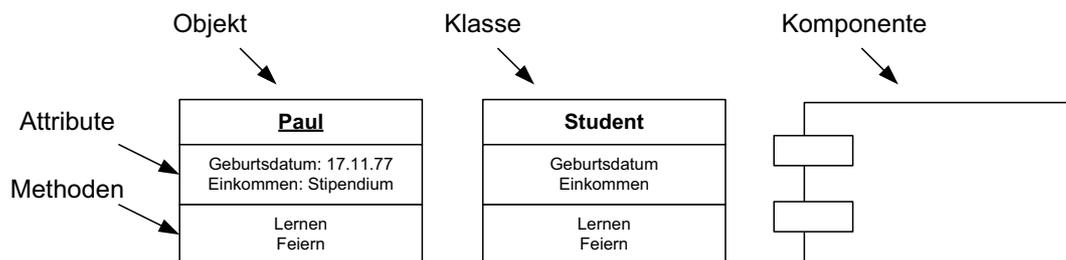


Bild 3-7: Darstellung eines Objektes (links), einer Klasse (mittig) und einer Komponente in UML-Notation [For07, S. 20]

**Komponente:** Da Softwareprogramme im Allgemeinen sehr komplex sind, bietet es sich an, ausführbare und austauschbare Einheiten zu definieren. Diese Einheiten werden als Komponenten bezeichnet und verfügen über festdefinierte Schnittstellen und eine eigene Identität. Intern besteht eine Komponente in der Regel aus einer Menge von Klassen, die das Verhalten realisieren.

**Bewertung:** Die Anwendung von UML ist fester Bestandteil der objekt- bzw. komponentenbasierten Softwareentwicklung. Mit dieser Modellierungssprache ist es möglich, alle notwendigen Aspekte von Software abzubilden. Neben der Repräsentation sämtlicher Modelle existieren feste Definitionen aller Schnittstellen und Verbindungen. Die Sprache eignet sich u.a. für die Darstellung von Metamodellen, ist jedoch für den fachdisziplinübergreifenden Systementwurf, aufgrund der Ausrichtung auf die Software, nicht geeignet.

### 3.2.1.3 SysML – Systems Modeling Language

Die SysML – Systems Modeling Language – ist eine für das Systems Engineering entwickelte Sprache [Alt12], [FMS12]. Im Gegensatz zu z.B. CONSENS ist sie unabhängig von jeglicher Methode erarbeitet worden. Die Basis der SysML ist die UML (vgl. Kapitel 3.2.1.2). Ziel der INCOSE – International Council on Systems Engineering – ist eine standardisierte Sprache zur Modellierung, Analyse und Verifikation komplexer technischer Systeme. Die erste Standardisierung gelang der OMG im April 2006 mit der SysML 1.0. Aktueller Stand 2014 ist die Version 1.3 [OMG12]. Ferner existieren zahlreiche Modifikationen des Standards hinsichtlich bestimmter Schwerpunkte. Ein Beispiel im Hinblick auf die Wiederverwendung mechatronischer Artefakte liefern BARBIERI ET AL. [BKF+14].

Einen Überblick über die verschiedenen Diagramme der SysML zeigt Bild 3-8. Diese werden nachfolgend erläutert [Wei06, S. 157ff.]. Die Diagramme lassen sich auf oberster Ebene unterscheiden in Struktur-, Anforderungs- und Verhaltensdiagramme. Im Vergleich zu der UML sind das Anforderungsdiagramm und das Zusicherungsdiagramm neu entwickelt worden. Die Paket-, Zustands-, Use Case-, Sequenz-Diagramme wurden direkt übernommen, wohingegen das Blockdefinitionsdiagramm und das Interne Blockdefinitionsdiagramm auf die Anforderungen der SysML angepasst wurden.

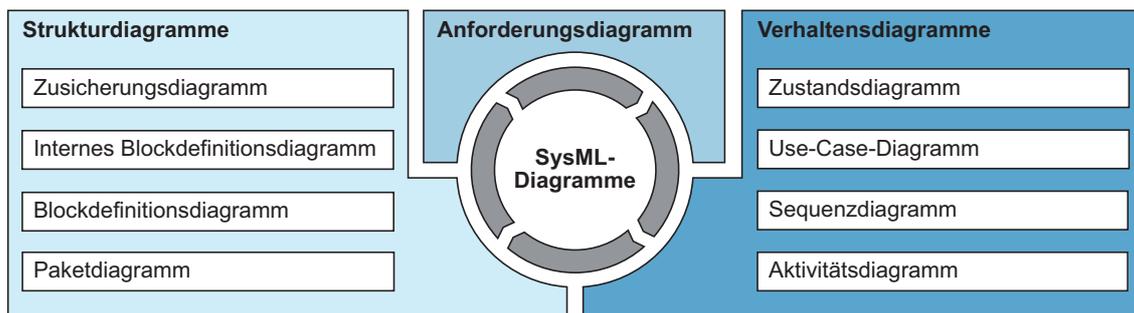


Bild 3-8: Diagramme der SysML [vgl. Wei06, S. 160]

**Strukturdiagramme:** Für die Darstellung von Strukturen bzw. Strukturkonfigurationen des zu entwickelnden Systems werden Blöcke genutzt. Sie eignen sich zur Abbildung informationsverarbeitender oder physikalischer Elemente gleichermaßen. Das Blockdefinitionsdiagramm wird genutzt, um die Beziehung zwischen den Blöcken, ihre Assoziationen, Generalisierungen sowie Abhängigkeiten zu beschreiben. Die Beziehung zwi-

schen Bestandteilen innerhalb eines Blocks werden durch Ports, Konnektoren und Flüsse im internen Blockdiagramm festgehalten. Zusicherungsdiagramme spezifizieren die Beziehungen zwischen Eigenschaften verschiedener Blöcke (z.B. parametrisierte Beziehungen wie physikalische Gesetze).

**Anforderungsdiagramm:** Ziel ist die Darstellung von funktionalen und nicht funktionalen Anforderungen, sowie bestehenden Beziehungen untereinander. Es werden Ableitungs-, Enthält-, Erfüllungs-, Kopie-, Prüf-, Verfeinerungs- und Verfolgungsbeziehungen unterschieden. Gängige Darstellungsarten sind Grafiken und Tabellen.

**Verhaltensdiagramme:** Für die Modellierung des Verhaltens stellt die SysML vier Diagrammtypen bereit. Mit einem Zustandsdiagramm werden mögliche Zustände und Zustandsübergänge abgebildet. Das Use-Case-Diagramm bietet die Möglichkeit, Interaktionen von Benutzern und/oder externen Systemen mit dem zu entwickelnden System zu spezifizieren. Mit Hilfe von Sequenzdiagrammen werden Interaktionen zwischen Systemelementen im Hinblick auf ihren zeitlichen Verlauf dargestellt. Kompliziert werden die Verhaltensdiagramme durch das Aktivitätsdiagramm. Es stellt Systemabläufe inkl. Ein- und Ausgabedaten dar. Es besteht die Möglichkeit der Dekomposition.

**Bewertung:** Die Spezifikationstechnik SysML ermöglicht eine fachdisziplinübergreifende Abbildung eines komplexen technischen Systems und deckt in ihrer Anwendung den kompletten Entwurf ab. Die zum Einsatz kommenden Konstrukte sind nicht nur für die Entwicklung mechatronischer Systeme ausgelegt, da die OMG mit der SysML den Anspruch auf Allgemeingültigkeit verfolgt. Die SysML unterscheidet drei Diagrammtypen, deren Verwendung ohne Methode nur grob vordefiniert ist. Die Anwendung dieser Spezifikationstechnik ist daher nicht immer eindeutig oder intuitiv. Ferner ist die Einarbeitung durch die Vielzahl an Konstrukten sehr zeitintensiv. Letzteres ist vor dem Hintergrund der industriellen Anwendung nahezu ein Ausschlusskriterium. Im Hinblick auf den Einsatz der Sprache zur Abbildung relevanter Wissensinhalte in den Lösungsmustern für den Systementwurf ist diese daher nur bedingt geeignet. Die Inhalte lassen sich zwar übergreifend abbilden, der hohe Aufwand in der Einarbeitung, auch der Experten aus den einzelnen Disziplinen wie z.B. dem Maschinenbau, ist jedoch ein sehr großes Hemmnis.

#### 3.2.1.4 PrEMISE

Am Deutschen Zentrum für Luft- und Raumfahrt (DLR) wurde das PrEMISE entwickelt – Pragmatic Engineering Model for Integrated Systems Engineering. Es ist ein integriertes Produktmodell, das in der frühen Phase des Entwurfs komplexer technischer Systeme eine ganzheitliche Sicht ermöglichen soll. Es hat den Anspruch, deutlich formaler zu sein als z.B. eine Anforderungsliste in Textform und dient in erster Linie zum Datenaustausch zwischen Entwicklungswerkzeugen. Das Produktmodell wird in den Mittelpunkt der Entwicklung gestellt und hat den Anspruch automatisiert Informationen

in weitere Werkzeuge zu transformieren und Änderungen auf Gesamtsystemebene transparent nachvollziehbar zu gestalten [SBE11].

Das PrEMISE-Modell erlaubt u.a. die Modellierung und Aggregation von Systemkomponenten in einer Hierarchie – Flugzeug, Turbine, Turbinenschaufel. Die Komponenten lassen sich mit Hilfe folgender Attribute beschreiben: Name und Beschreibung; Liste von Subkomponenten; Parameter zur Abbildung der Charakteristika einer Komponente; Berechnungsformeln, die eine Modellierung der Abhängigkeiten zwischen den Parameterwerten ermöglichen; Operationsmodi, um Parameterwerte zu beeinflussen; Schnittstellen und Flussbeziehungen zur Darstellung physikalischer Zusammenhänge.

Die logische Architektur in PrEMISE ermöglicht die Darstellung komplexer Zusammenhänge zwischen mehreren Parametern, um mit deren Hilfe Simulationen und Berechnungen von Produktvarianten durchführen zu können. Das Modell wird in einem offenen Textformat (XMI) gespeichert. Dies ist Grundvoraussetzung für den Datenexport in beispielsweise Excel, CATIA oder Simulink. Bild 3-9 zeigt den Unterschied beim Datenaustausch mit und ohne zentrales Datenmodell. Ohne ein integriertes Datenmodell resultieren insgesamt zehn Schnittstellen zwischen den beispielhaft gewählten Werkzeugen. Durch das in PrEMISE vorgesehene integrierte Datenmodell (IPM) bestehen nur Schnittstellen zwischen dem IPM und den Werkzeugen. Es wird das grundsätzliche Ziel verfolgt, durch eine Reduzierung der Schnittstellen, Informationsverluste bei den notwendigen Transformationen zu minimieren [SBE11].

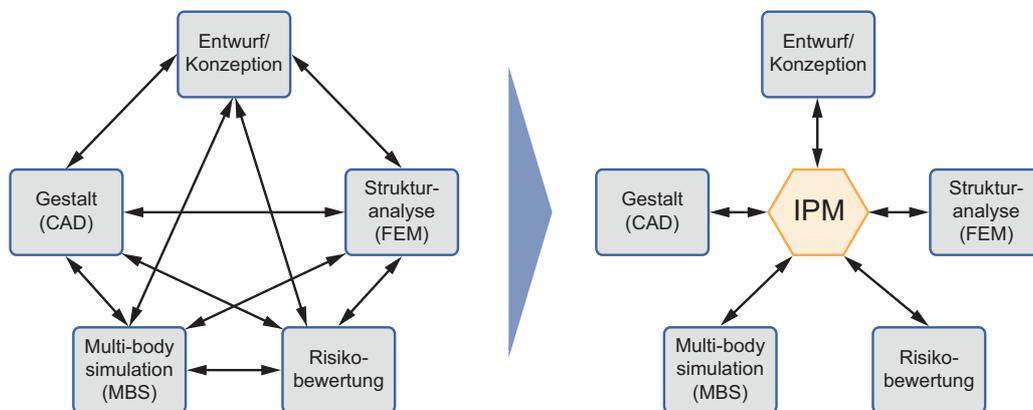


Bild 3-9: Unterschied beim Datenaustausch ohne (links) und mit (rechts) integriertem Produktmodell (IPM) [vgl. SBE11]

**Bewertung:** Das in PrEMISE angestrebte integrierte Produktmodell bietet lediglich die Abbildung der logischen Architektur des Systems. Die Problemanalyse hat ergeben, dass dies ein wichtiger Aspekt des für den Systementwurf relevanten Lösungswissens ist. Neben der Architektur bedarf es jedoch auch Darstellungsmöglichkeiten für informationstechnische Abläufe sowie Anforderungen und Funktionen. Das untersuchte Modell bietet jedoch nicht die Möglichkeit z.B. Funktionen und ihre Hierarchisierung ausreichend zu spezifizieren. Im Hinblick auf die angestrebte Aggregation von Lösungsmustern in elementare und komplexe Muster ist dies jedoch eine Grundvoraussetzung.

### 3.2.1.5 Modelica®

Die objektorientierte Sprache Modelica® wurde im Rahmen des ESPRIT-Projekts „Simulation in Europe Basic Research Working Group“ zur Modellierung physikalischer heterogener Systeme entwickelt. Sie wird durch die Modelica Association weiterentwickelt und vermarktet. Mit Modelica® wird das Ziel verfolgt, Modellierungssprachen wie VHDL-AMS<sup>38</sup> und ObjectMath<sup>39</sup> zu vereinheitlichen. Die Sprache wird in unterschiedlichen Softwarewerkzeugen eingesetzt, z.B. Dymola<sup>40</sup>.

Die Sprache erlaubt die Erstellung und Verknüpfung komplexer physikalischer Modelle aus verschiedenen Bereichen [Fri04, S. 71ff.]. Bild 3-10 zeigt beispielhaft eine Geschwindigkeitsregelung mit Modelica®. Die mathematischen Modelle sind dabei sowohl durch Differentialgleichungen als auch durch algebraische und diskrete Gleichungen beschrieben. Diese beruhen auf physikalischen Gesetzen und ermöglichen die Abbildung von kontinuierlichen und zeitdiskreten Variablen. Bei der Komponentenmodellierung werden die Komponenten des Systems durch Symbole dargestellt, durch Beziehungen miteinander verbunden und durch Parameter konkretisiert. Das Symbol repräsentiert dabei das mathematische Modell der Komponente. Modelica® unterstützt den disziplin- und werkzeugübergreifenden Austausch von Modelldaten und die Wiederverwendung von Modellen sowie Teilen davon. Die Modellierung mit Modelica® erfolgt auf Komponenten- oder Gleichungsebene [Mod10].

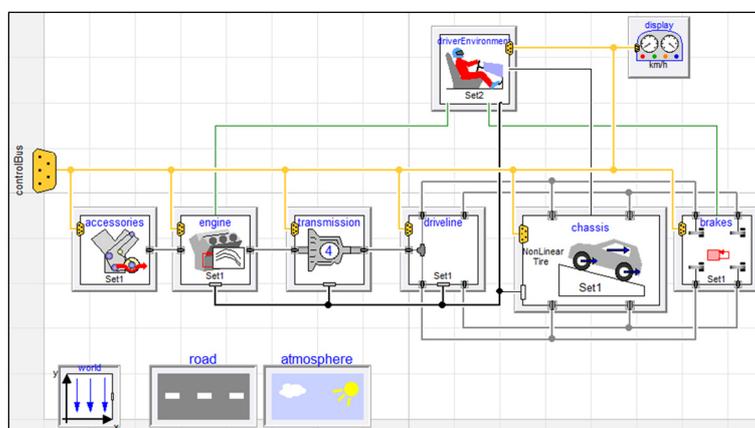


Bild 3-10: Beispiel einer Geschwindigkeitsregelung mit Modelica®

<sup>38</sup>VHDL-AMS (VHSIC Hardware Description Language – Analog and Mixed Signal Extensions) ist eine Hardware-Beschreibungssprache zur formalen und textbasierten Spezifikation digitaler Schaltungen [IEEE1076.1].

<sup>39</sup>ObjectMath ist eine objektorientierte Erweiterung der Computer-Algebra-Sprache „Mathematica“. Sie ermöglicht die Strukturierung mathematischer Modelle durch Gruppierung mathematischer Funktionen und Gleichungen [FHV92].

<sup>40</sup>Dymola (Dynamik Modeling Language) ist ein rechnergestütztes Werkzeug zur Modellierung und Simulation komplexer Systeme, zusammengesetzt aus beispielsweise mechanischen, elektronischen, thermodynamischen oder pneumatischen Komponenten.

Vor dem Hintergrund der durch die Lösungsmuster angestrebten Wiederverwendung von Lösungswissen sind die bereits veröffentlichten Modellbibliotheken zu berücksichtigen. Die Modelica Association stellt in Form der sogenannten Modelica Standard Library eine Open-Source Bibliothek zur Verfügung, die Modelle aus Disziplinen wie der Elektronik, der Thermodynamik oder der Mechanik beinhaltet [Mod10].

**Bewertung:** Modelica<sup>®</sup> ist eine Sprache zur Beschreibung physikalischer Systeme. Sie eignet sich in erster Linie zur Anwendung in den Disziplinen Maschinenbau, Elektrotechnik und Regelungstechnik sowie in ihrem Zusammenspiel. Durch Modellierung des Verhaltens mit mathematischen Gleichungen ermöglicht die Sprache eine Auslegung der physikalischen Eigenschaften sowie deren rechnerunterstützte Simulation. Die Spezifikation der Informationsverarbeitung war zunächst nicht Ziel der Entwicklungen, sie beschränkte sich auf die Regelungs- und Steuerungstechnik. Erste Bibliotheken liefern aber z.T. bereits wiederverwendbare Bausteine für die Spezifikation von Echtzeitkoordination auf System oder Subsystemebene [PDS+12]. Durch den hohen Formalitätsgrad der Sprache birgt die Anwendung im Systementwurf jedoch gewisse Gefahren wie z.B. die Fokussierung auf bestimmte Teilaspekte in einer sehr hohen Detaillierungstiefe. Auch im Hinblick auf die Anforderung beim Managen des Entwicklungsgeschehens zu unterstützen, ist die Darstellung des Systems durch mathematische Gleichungen nicht zielführend.

### 3.2.2 Fachdisziplinübergreifende Lösungsmuster

Der Ursprung fachdisziplinübergreifender Lösungsmusteransätze liegt im Systems Engineering. Die Analyse zweier exemplarisch ausgewählter Ansätze zeigt Kapitel 3.2.2.1 mit den Systemarchitekturmustern nach CLOUTIER sowie Kapitel 3.2.2.2 mit den Lösungsmustern mit SysML nach WEILKINS. Einen Ansatz mit Schwerpunkt Entwicklungsmethodik Mechatronik behandelt Kapitel 3.2.2.3, mit den Lösungsmustern für selbstoptimierende Systeme nach DUMITRESCU.

#### 3.2.2.1 Systemarchitekturmuster nach CLOUTIER

CLOUTIER beruft sich in seiner Definition von einem Lösungsmuster auf die Definition von GAMMA ET AL. und folglich auf den Architekten ALEXANDER. Ein Lösungsmuster benennt, motiviert und beschreibt dabei ein grundsätzliches Lösungsprinzip für wiederkehrende Entwurfsprobleme. Anwendungsschwerpunkt der Systemarchitekturmuster ist der objektorientierte Systementwurf. Ein Muster enthält Angaben zur Implementierung, d.h. die Übertragung auf ähnliche Problemstellungen, und Anwendungsbeispiele. Die Lösung wird repräsentiert durch generalisierte Objekte und Klassen. Diese Art der Aufbereitung ist notwendig, um das Lösungsmuster auf weitere Probleme adaptieren zu können [Clo06].

Als Einsatzzweck nennt CLOUTIER die Erstellung der Systemarchitektur. Gemeint ist die Kunst und Wissenschaft komplexe technische Systeme sowie deren Entstehungsprozesse zu strukturieren und zu beherrschen. Aufgrund der steigenden Komplexität der Systeme kann dies nicht durch einen einzelnen Systemarchitekten erfolgen. Es ist vielmehr ein Team von Systemarchitekten notwendig, die sich mit ihrer spezifischen Expertise komplementär ergänzen, um den gesamten Entstehungsprozess gemeinsam zu managen. In diesem Zusammenhang sieht CLOUTIER Lösungsmuster als ein wesentliches Werkzeug zur Förderung der Kommunikation. Voraussetzung ist die in Bild 3-11 abgebildete einheitliche Strukturierung von Mustern (Ein Beispielmuster befindet sich im Anhang A1.2).

### Einheitliche Strukturierung der Lösungsmuster für das Systems Engineering

Kategorie	Erklärung zu den Kategorien
Name	Der Name soll dem Anwender den Nutzen des Musters prägnant vermitteln
Aliasnamen	Weitere Namen, durch die das Muster identifiziert werden kann
Schlüsselwörter	Schlüsselwörter, die die Lokalisierung des Lösungsmusters in einer Datenbank verbessern
Problem-Kontext	Eine prägnante Beschreibung des Problemumfeldes: Hierzu sind Anwendungsszenarien über Situationen zu spezifizieren, die das Problem charakterisieren
Problem-beschreibung	Was ist das Problem, für das dieses Lösungsmuster genutzt werden kann?
Herausforderungen	Mögliche Probleme und/oder Einschränkungen, die bei der Implementierung bzw. Anwendung von dem Lösungsmuster auftreten können
Lösung	Eine Beschreibung über den Kern der Lösung bzw. das Lösungsprinzip
Diagramme	Ein oder mehrere Diagramme bzw. Modelle, die das Lösungsmuster repräsentieren
Schnittstellen	Beschreibung der Schnittstellen im Hinblick auf Veränderbarkeit von Parametern (z.B. Informations, Stoff- oder Energiefluss)
Resultierender Kontext	Erläuterung zum Kontext, in dem das Lösungsmuster eingesetzt werden kann
Beispiel(e)	Ein oder mehrere Beispiele, in denen das Lösungsmuster eingesetzt wurde
Funktions-nachweis	Begründung über den erfolgreichen Einsatz des Lösungsmusters
Übertragungsmöglichkeiten	Hinweise zur Übertragbarkeit des Lösungsmusters auf weitere Problemstellungen
Verwandte Muster	Lösungsmuster, mit denen eine Kombination möglich ist oder ausgeschlossen werden kann
Referenzen	Weitere Informationen oder Literatur, die das Verständnis des Lösungsmusters fördern
Autoren	Wer hat das Lösungsmuster dokumentiert?

Bild 3-11: Einheitliche Strukturierung von Systemarchitekturmustern nach CLOUTIER [vgl. Clo06, S. 42]

Die Aufgabenfelder und Fähigkeiten der Systemarchitekten sind dabei sehr vielseitig. Neben der Modellierung eines fachdisziplinübergreifenden Systemmodells stehen vor allem kommunikative Fähigkeiten im Vordergrund. Ein Systemarchitekt als Vermittler verantwortet die Abstimmung zwischen den Fachdisziplinen entlang des Entstehungsprozesses. Es gilt eine gemeinsame Plattform zu schaffen, über die sich alle am Produktentstehungsprozess beteiligten Akteure gleichermaßen austauschen können. Kernelement sind Systemarchitekturmuster, die die Verständlichkeit über die Fachdisziplinen hinweg sicherstellen, indem sie die spezifischen Lösungsansätze miteinander verbinden bzw. vereinen.

Als wesentliches Alleinstellungsmerkmal der Systemarchitekturmuster beschreibt CLOUTIER die Dokumentation von Schnittstellen. Schnittstellen zu weiteren Architekturmustern, aber auch Schnittstellen zu fachdisziplinspezifischen Lösungsmustern. Hinzu kommen die Aggregation von Lösungsmustern und die Dokumentation der daraus resultierenden Beziehungen untereinander.

**Bewertung:** CLOUTIER beschäftigt sich mit Lösungsmustern für die Systemarchitekturerstellung. Dies umfasst die Modellierung eines fachdisziplinübergreifenden Systemmodells sowie dessen fortlaufende Pflege über den Entstehungsprozess. CLOUTIER sieht den Nutzen im Lösungsmusteransatz in der Förderung der Kommunikation zwischen den Fachdisziplinen. Im Hinblick auf eine gemeinsame Plattform zum Austausch über Problem- und Lösungsprinzipien präsentiert er eine einheitliche Strukturierung von Systemarchitekturmustern. Diese sehr umfangreiche Art der Informationsdokumentation umfasst neben generalisierten Modellen auch Zusatzinformationen für den Anwender wie z.B. Nutzungs- und Implementierungshinweise. Ein wesentlicher Nachteil dieser Art der Repräsentation ist die Informationsflut und intransparente Darstellung lösungsmusterübergreifender Beziehungen. Gemeint sind Schnittstellen und wechselseitige Abhängigkeiten zu weiteren Lösungsmustern anderer Fachdisziplinen als auch Aggregationsbeziehungen.

### 3.2.2.2 Lösungsmuster mit SysML nach WEILKINS

Nach WEILKINS ist ein Lösungsmuster ein Hilfsmittel, um erfolgreich eingesetztes Lösungswissen langfristig zu speichern und explizit darzustellen. Wie die Vielzahl der in der Literatur veröffentlichten Definitionen bezieht sich auch WEILKINS auf den Architekturtheoretiker ALEXANDER. Jeder Entwickler, besonders wenn er langjährige Berufserfahrung hat, verfügt über ein umfangreiches Repertoire an Lösungsmustern. Er setzt es tagtäglich bei der Bewältigung von Problemstellungen ein. Um ein Lösungsmuster zu definieren, ist es erforderlich, die erarbeitete Lösung generalisiert zu beschreiben und somit ihre Wiederverwendbarkeit und Übertragbarkeit zu ermöglichen [Wei14-ol].

Im Gegensatz zu zahlreichen Ansätzen für Entwurfsmuster fokussiert WEILKINS mit seinen Arbeiten die grafische Anordnung von Elementen. Er beschreibt die grafische Repräsentation des zu entwickelnden Systems bzw. seiner Subsysteme als Muster.

WEILKINS nutzt dabei die Spezifikationstechnik SysML für die Systemmodellierung. Im Hinblick auf die Verständlichkeit spielt dabei die Art und Weise der Anordnung eine entscheidende Rolle. Nach WEILKINS trägt das Layout der Diagramme wesentlich zur Effizienzsteigerung in der Systemmodellierung bei [Wei14-ol]. Bild 3-12 verdeutlicht drei identifizierte Layoutmuster von WEILKINS.

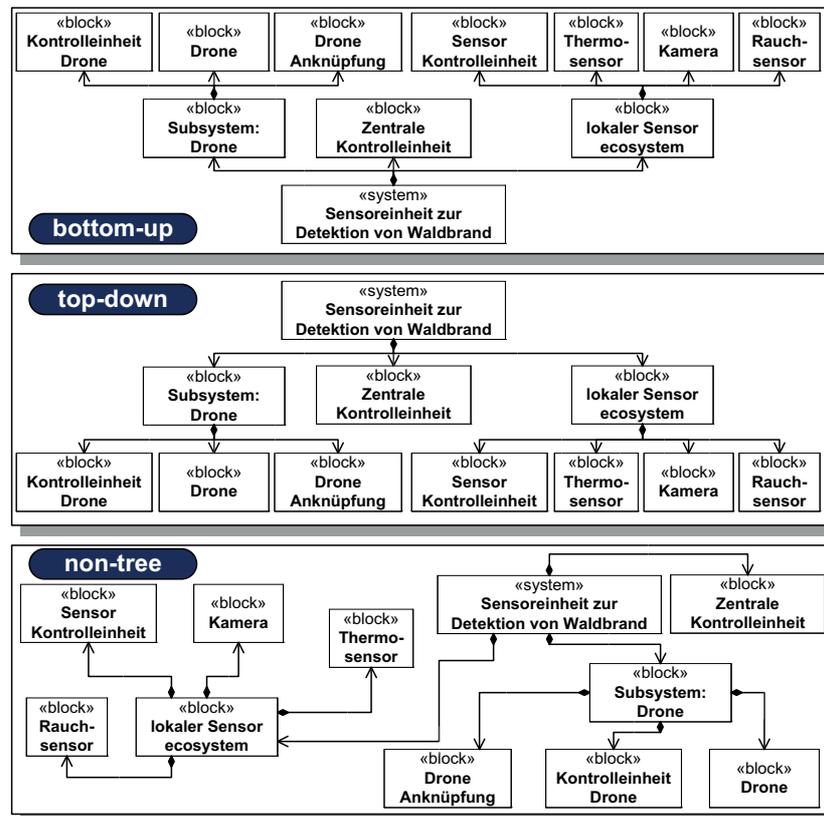


Bild 3-12: Muster „bottom-up“, „top-down“ und „non-tree-style“ nach WEILKINS [vgl. Wei14-ol]

Das Anwendungsbeispiel besteht aus einer „komplexen Sensoreinheit zur Identifizierung von Waldbränden“, die mit SysML abgebildet ist. Es sind unterschiedliche Abbildungsformen eines Produktbaumes dargestellt. Hierbei wird ein Gesamtsystem in seine Teilsysteme zerlegt und mit logischen Beziehungen verbunden. WEILKINS unterscheidet zwischen „bottom-up“, „top-down“ und „non-tree-style“.

**Bewertung:** Der Ansatz nach WEILKINS verdeutlicht, wie facettenreich der Lösungsmusterbegriff im Kontext Systems Engineering interpretiert wird. Im Gegensatz zu der Vielzahl der Ansätze handelt es sich um Layoutmuster, die Hinweise zur Anordnung von Modellen geben. In Anlehnung an die Unterteilung nach GRABOWSKI (Kapitel 3.1.1) handelt es sich hierbei um eine Art Prozessmuster. Sie unterstützen die Entwickler nicht bei der Frage, was eine mögliche Lösung für das Entwurfsproblem (fachdisziplinübergreifend) ist. Es sind vielmehr sehr generische und abstrakte Hinweise, die sprachübergreifend die Abbildung von Systemen mit Hilfe von Modellen betrifft.

### 3.2.2.3 Lösungsmuster für selbstoptimierende Systeme nach DUMITRESCU

DUMITRESCU schlägt Lösungsmuster für die Wissensexternalisierung und Wiederverwendung zur Integration kognitiver Funktionen in mechatronische Systeme vor. Die erarbeitete Spezifikation der Lösungsmuster eignet sich sowohl zur Aufnahme von Lösungen mit physikalischem als auch informationstechnischem Schwerpunkt. Hauptaugenmerk liegt jedoch auf letztgenannten Lösungsansätzen, vorrangig aus den Bereichen der künstlichen Intelligenz und der höheren Mathematik. Bild 3-13 gibt einen Überblick über die *einheitliche Spezifikation eines Lösungsmusters für Selbstoptimierung* nach DUMITRESCU. Die Lösungsmuster werden in sechs Aspekte gegliedert, die nachfolgend erläutert werden. Die Wissensinhalte werden mit Hilfe der Spezifikationstechnik CONSENS dargestellt [Dum10, S. 129ff.].

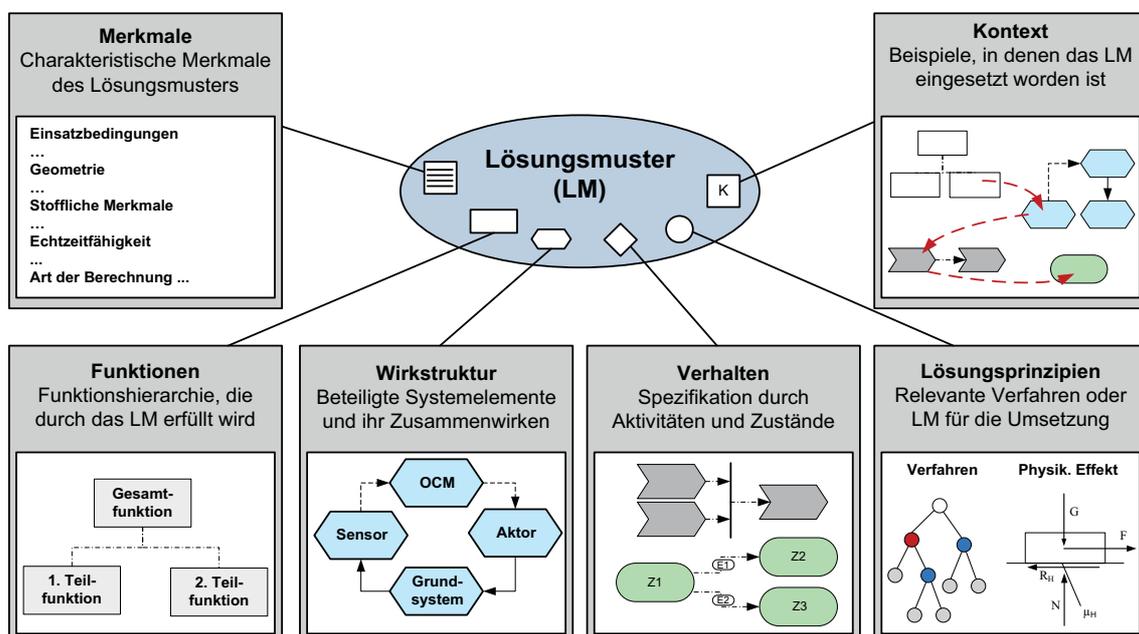


Bild 3-13: Einheitliche Spezifikation eines Lösungsmusters nach DUMITRESCU [Dum10, S. 130]

#### Merkmale

Als „Merkmale“ werden charakteristische Eigenschaften des Lösungsmusters festgehalten, die u.a. zum Abgleich mit den Anforderungen in einer Neuentwicklung genutzt werden können. Es wird zwischen Merkmalen für die Informationsverarbeitung und Merkmalen für das (physikalische) Grundsystem eines mechatronischen Systems unterschieden. DUMITRESCU schlägt nachfolgende Liste an Eigenschaften für das Grundsystem vor: *Einsatzbedingungen, Geometrie, Stoffliche Merkmale, Kinematik, Montage, Normung*. Es wird betont, dass diese Liste keinen Anspruch auf Vollständigkeit besitzt. Vielmehr ist diese Liste fortwährend zu erweitern, sobald die Lösungsmuster als Wissensmanagementansatz im Unternehmen implementiert sind. Zu den charakteristischen Merkmalen für Lösungen der Informationsverarbeitung zählt DUMITRESCU: *Einsatzbe-*

dingungen; Zeitpunkt der Aktivität, Echtzeitfähigkeit; Modellierbarkeit, Art der Berechnung, Entität, Signalart, Modellierungs-/Programmiersprache, Datenstruktur.<sup>41</sup>

### **Funktionen**

Dieser Aspekt beschreibt die Aufgabe eines Lösungsmusters in Form von „Funktionen“. Im einfachsten Fall ist dies genau eine Funktion. Analog z.B. zur Aggregation von Bauteilen zu einer Baugruppe können komplexere Lösungen mehrere Funktionen umfassen. Die Funktionen und die entsprechenden Zusammenhänge werden in einer Funktionshierarchie dargestellt. DUMITRESCU unterscheidet, in Anlehnung an die drei prinzipiellen Flussarten mechatronischer Systeme zwischen stoffbestimmten (z.B. Stoff transportieren), energiebestimmten (z.B. Energie umwandeln) und informationsbestimmten Funktionen (z.B. Informationen erfassen). Aufbauend auf dem zuvor erläuterten Aspekt *Merkmale*, komplettiert der Aspekt Funktionen die Problembeschreibung eines Lösungsmusters. DUMITRESCU konzentriert sich hierbei auf kognitive Funktionen wie z.B. „planen“ oder „lernen“.

### **Wirkstruktur**

In Anlehnung an ALEXANDER umfasst jedes Lösungsmuster neben einer Problem- auch eine Lösungsbeschreibung. In der Spezifikation von DUMITRESCU bildet die „Wirkstruktur“ den Kern der Lösungsbeschreibung. Es werden Systemelemente dargestellt, die in ihrem Zusammenwirken die definierten Funktionen erfüllen<sup>42</sup>.

### **Verhalten**

Der Aspekt „Verhalten“ komplettiert die Lösungsbeschreibung eines Musters. DUMITRESCU betont die Relevanz dieses Aspekts vor allem in Bezug auf Lösungswissen für die Selbstoptimierung. Er unterteilt diesen Aspekt in die beiden Bereiche *Verhalten – Aktivitäten* und *Verhalten – Zustände*.

- Das Partialmodell *Verhalten – Aktivitäten* stellt Ablaufdiagramme dar. Diese repräsentieren in welcher Reihenfolge ein Systemelement seine Funktionen ausführt. Insbesondere bei kognitiven Prozessen sollte, unter Berücksichtigung des Selbstoptimierungsprozesses, Bezug auf die Hauptaktivitäten *Situationsanalyse*, *Zielbestimmung* und *Verhaltensanpassung* genommen werden<sup>43</sup>.

---

<sup>41</sup>Für eine detaillierte Erläuterung sei auf [Dum10, S. 131f.] verwiesen.

<sup>42</sup>Für weitere Informationen bzgl. einer Wirkstruktur sei auf Kapitel 3.2.1.1 verwiesen.

<sup>43</sup>Die Situationsanalyse umfasst neben der Aufzeichnung relevanter Zustandsgrößen des Systems auch die Wahrnehmung äußerer Einflüsse. Eine Voraussetzung für einen Lerneffekt ist das Speichern sämtlicher Systemzustände, um einen Abruf zurückliegender Aufzeichnungen zu ermöglichen. Ziel der Zielbestimmung ist die Definition neuer z.T. konfliktär zueinander stehenden Systemziele (z.B. bei einer Bahn: max. Leistung, min. Verbrauch). Als Entscheidungsgrundlage dienen die zuvor gewonnenen Informationen. Unterschieden wird zwischen der Auswahl, Anpassung und Generierung von Zielen. Im Rahmen der Verhaltensanpassung werden Veränderungen bzw. Neueinstellung der Systemparameter vorgenommen [ADG+09, S. 6ff.]

- Neben Aktivitäten eignen sich vor allem Zustandsdiagramme, um informationstechnische Abläufe darzustellen. Ferner können mit diesen Diagrammen auch sich ändernde Zustände von passiven Elementen abgebildet werden. DUMITRESCU hebt die Interdisziplinarität dieser Diagramme hervor, da die Dokumentation von Zuständen und Zustandsübergängen für jedes Lösungsmuster aller beteiligten Fachdisziplinen relevant sein kann.

### **Lösungsprinzipien**

Jedes Lösungsmuster beruht auf mindestens einem „Lösungsprinzip“ bzw. einem Verfahren. Diese sind die Grundlage für die Implementieren eines Lösungsmusters im Rahmen der Produktentwicklung. Verfahren umfassen nach DUMITRESCU<sup>44</sup> die Abfolge von physikalisch-technischen, chemischen, biologischen oder informationstechnischen Wirkungsabläufen, die zur Realisierung einer gewünschten Funktion notwendig sind. Ein Verfahren konkretisiert eine Funktion. Bei der Definition von Lösungsmustern für die Selbstoptimierung werden zwei Verfahrensarten unterschieden: *Physikalische Effekte* und *informationsverarbeitende Verfahren*.

### **Kontext**

Die Lösungsmusterspezifikation wird komplettiert durch den „Kontext“. Dieser beschreibt, in welchen zurückliegenden Projekten das betrachtete Lösungsmuster erfolgreich eingesetzt wurde. Somit muss jede vollständige Spezifikation eines Lösungsmusters mindestens ein Beispiel beinhalten. Die Kontextbeschreibung setzt sich zusammen aus einer Bezeichnung, einer kurzen Beschreibung in Textform sowie einer aussagekräftigen Skizze. Im Idealfall liegt die Einbettung des Lösungsmusters in das Gesamtsystem in den von DUMITRESCU gewählten Partialmodellen vor: Funktionen, Wirkstruktur und Verhalten. Ferner spielen die Querverweise eine wichtige Rolle. Sie geben den Entwicklern wichtige Hinweise für die Anwendung des Lösungsmusters in einer Neuentwicklung.

**Bewertung:** DUMITRESCU präsentiert eine einheitliche Spezifikation von Lösungsmustern, mit deren Hilfe kognitive Funktionen in mechatronische Systeme integriert werden können. Es werden unterschiedliche Kategorien definiert. Das Hinterlegen von notwendigen Informationen erfolgt vorrangig modellbasiert durch Nutzung der Spezifikationstechnik CONSENS. Die Strukturierung ist gut geeignet, um multidisziplinäre Lösungsansätze darzustellen. Die Navigation über den Aspekt Kontext und eine ergänzende Erklärung im Aspekt Lösungsprinzipien ist für einen kleinen Anwendungsbereich wie z.B. künstliche Intelligenz ausreichend. In einer umfangreichen Übertragung auf alle am Systementwurf beteiligten Fachdisziplinen, wie sie im Rahmen der vorliegenden Arbeit angestrebt wird, ist mit einem deutlichen Komplexitätsanstieg in den wechselseitigen Abhängigkeiten zu rechnen. In diesem Zusammenhang ist es notwendig, die Strukturie-

---

<sup>44</sup> Die Definition von DUMITRESCU beruht auf den Arbeiten von SAUER. Für weitere Informationen sei an dieser Stelle auf [Sau06, S. 74f.] verwiesen.

nung zu vereinfachen und Darstellungsformen zu erarbeiten, die die Beziehungen untereinander und zu den Fachdisziplinen transparent abbilden.

### 3.2.3 Fachdisziplinspezifische Lösungsmuster

Für die Wiederverwendung von Lösungsprinzipien existieren zahlreiche fachdisziplinspezifische Ansätze. Aus diesem Grund untersucht Kapitel 3.2.3.1 mit den Lösungsmustern nach SUHM sowie Kapitel 3.2.3.2 mit den Entwurfsmustern nach SALUSTRI Ansätze für den klassischen Maschinenbau. Ein Ansatz zur Spezifikation von Koordinationsmustern der Softwaretechnik wird in Kapitel 3.2.3.3 beschrieben. Den Abschluss bildet die Analyse von Ansätzen aus der Regelungstechnik. Hierzu werden in Kapitel 3.2.3.4 und 3.2.3.5 die Arbeiten von FÖLLINGER und SANZ/ZALEWSKI untersucht.

#### 3.2.3.1 Lösungsmuster im Maschinenbau nach SUHM

Lösungsmuster für den klassischen Maschinenbau, die über die etablierten Wirkprinzipien<sup>45</sup> oder Lösungsprinzipien aus der VDI-Richtlinie 2727 hinausgehen, liefert SUHM [VDI2727]. Die Muster repräsentieren Expertise, z.B. über Gestaltungsrichtlinien oder den Einsatz typischer Lösungselemente (z.B. Radiallager). Der Aufbau eines Lösungsmusters ist in Bild 3-14 beispielhaft dargestellt. SUHM unterteilt ein Lösungsmuster in die Kategorien Voraussetzungen, Lösung sowie Umgebung [Suh93, S. 78ff.].

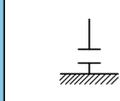
Lösungsmuster Radiallager		
Radiallagerung		Bezeichnung
DIN 615, DIN 617,...		Quelle
		
		Symbol
Voraussetzungen	Funktionsoperation	(Kraft) leiten
	Funktionsgröße	Radialkraft, Axialkraft
	Wartung	wartungsfrei
Lösung	Relativbewegung	wälzen
	Tragzahl	0,3 ... 5000 kN
	Drehzahl	< 10000 min <sup>-1</sup>
Umgebung	sitzt auf	Welle
	sitzt in	Lagerblock, Gehäuse
	Sicherung innen	keine, Sprengring, Mutter

Bild 3-14: Steckbrief eines Lösungsmusters [Suh93, S. 79]

- **Voraussetzungen:** Diese Kategorie enthält Angaben aus vorangegangenen Konstruktionsphasen, die als Voraussetzungen für das vorliegende Muster gelten.

<sup>45</sup> Vgl. Kapitel 2.4.4

- **Lösung:** Es wird eine Lösungsbeschreibung in parametrisierter Form angegeben.
- **Umgebung:** Beschreibung der Lösungsumgebung innerhalb der gleichen Konstruktionsphase, die bereits vorhanden ist oder noch ausgearbeitet werden muss.

SUHM betont in seiner Arbeit die Notwendigkeit, komplexe Problemstellungen und somit auch die zugehörigen Lösungen bzw. Lösungsmuster einem Strukturierungsmodell unterzuordnen. Dazu definiert er drei generische Hierarchietypen, entlang derer ein spezifisches Lösungsmuster für den Maschinenbau charakterisiert werden kann (Bild 3-15) [Suh93, S 80ff.].

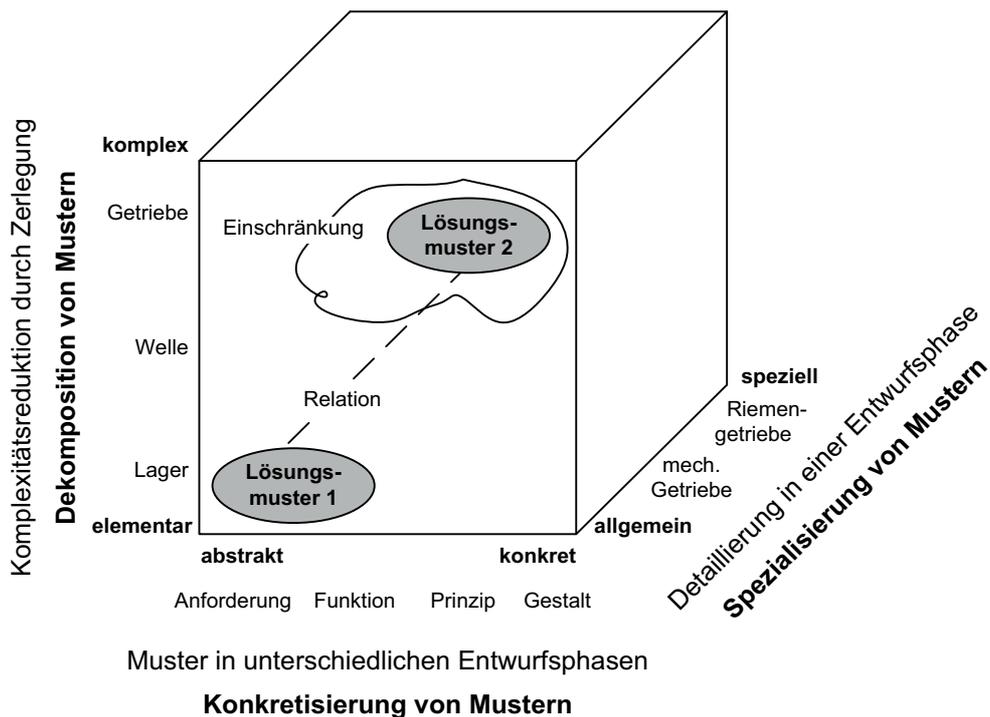


Bild 3-15: Strukturierung der Lösungsmuster [Sch06, S. 41] in Anlehnung an [Suh93, S. 80ff.]

- **Konkretisierungshierarchie:** Diese Ebene beschreibt den Einsatz der Muster in unterschiedlichen Entwurfsphasen. Grundgedanke ist, bei zunehmender Konkretisierung zunächst mit abstrakten Lösungsmustern zu arbeiten und diese später zu verfeinern und zu detaillieren. Diese Herangehensweise und Hierarchieebene soll dem Arbeiten in unterschiedlich abstrakten Konstruktionsphasen gerecht werden.
- **Spezialisierungshierarchie:** Diese Hierarchie steht für die Konkretisierung eines Musters in einer speziellen Entwurfsphase. So wird bei der Lösung einer bestimmten Problemstellung mit einem sehr allgemeinen Lösungsmuster begonnen. Unter Berücksichtigung weiterer Merkmale, die anschließend festgelegt

werden, wird die Lösung entlang einer Spezialisierungshierarchie sukzessive verfeinert.

- **Dekompositionshierarchie:** Diese Ebene bildet die Zerlegung eines komplexen Musters in elementare Muster ab. Die Dekompositionshierarchie soll es ermöglichen, die Komplexität des anstehenden Problems zu zerlegen und somit beherrschbar zu machen. Analog zum Komplexitätsgrad des Problems soll der Lösungsraum mit den entsprechend komplexen und elementaren Lösungsmustern eingeschränkt werden.

Im Rahmen der vorliegenden Arbeit stellt insbesondere die Komplexitätsreduktion durch die Unterteilung in elementare und komplexe Lösungsmuster einen bedeutenden Ansatz dar. Nach SUHM lassen sich komplexe Lösungsmuster grundsätzlich in elementare Lösungsmuster zergliedern. Aufgrund der damit einhergehenden hohen Granularität, lässt sich jede Konstruktionsaufgabe durch Kombination von elementaren Lösungsmustern lösen. Der Zusammenhang zwischen komplexen und elementaren Lösungsmustern nach SUHM ist anhand eines Lagersitzes auf einer Welle in Bild 3-16 dargestellt.

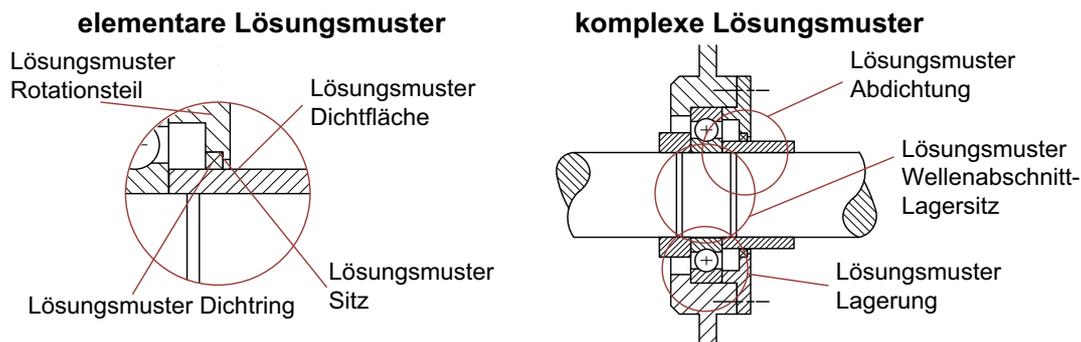


Bild 3-16: Zusammenhang zwischen komplexen und elementaren Lösungsmustern  
[Suh93, S. 81]

**Bewertung:** Die Lösungsmuster nach SUHM konzentrieren sich auf die Beschreibung rein maschinenbaulicher Problemstellungen. Die Darstellung von Wissensinhalten erfolgt textuell in tabellarischer Form. Dies ermöglicht zwar eine grundsätzliche Übertragung auf weitere Disziplinen, stellt aber einen bedeutenden Mehraufwand bei der Erstellung der Lösungsmuster dar. Vorteilhafter wäre die Darstellung durch die in der Entwicklung verwendeten Modelle oder Beschreibungssprachen. Positiv hervorzuheben sind die Überlegungen zu den unterschiedlichen Dimensionen, mit denen sich Lösungsmuster charakterisieren und voneinander abgrenzen lassen. Der dadurch entstehende Lösungsraum beschränkt sich jedoch, wie alle Überlegungen von SUHM, auf den klassischen Maschinenbau. Folglich sind im Rahmen der vorliegenden Arbeit einige positive Aspekte aufzugreifen und entsprechend der Interdisziplinarität des Systementwurfs zu erweitern.

### 3.2.3.2 Entwurfsmuster nach SALUSTRI

SALUSTRI beschäftigt sich mit Entwurfsmustern für den Maschinenbau, den sog. „Engineering Design Pattern“. Er bezieht sich bei seiner Interpretation des Begriffes Muster auf die Definition von ALEXANDER<sup>46</sup>. Demnach versteht er unter einem Muster eine Art Sprache, mit der eine kontextgebundene Lösung eines Problems, bzw. einer Klasse von Problemen, beschrieben werden kann. Die Lösungsmuster sollen helfen, Wissen zu bewahren und wiederabrufbar zu machen. Ferner tragen sie dazu bei, erfolgreich eingesetztes Lösungswissen zu verbreiten. Muster sind eine Art beschreibende Anleitung für den Wissenstransfer zwischen einem Individuum und einer Gruppe. SALUSTRI betont die Notwendigkeit der Verbreitung und Erweiterung des externalisierten Wissens via Internet [Sal01], [Sal05].

Grundvoraussetzung dafür ist der einheitliche Aufbau eines Lösungsmusters sowie eine allgemeinverständliche Darstellung der Lösung. SALUSTRI definiert hierfür die sechs Aspekte **Name**, **Problembeschreibung**, **Anforderungsliste**, **Kontextbeschreibung**, **Lösungsweg** sowie **Konsequenzen**, die jedes Muster enthalten muss. Diese Aspekte ordnet er fünf übergeordneten Kategorien zu, die nachfolgend erläutert werden [Sal05].

- **Muster-Name (Descriptive Pattern Name):** Diese Kategorie beinhaltet den Namen, den Autor sowie das Bearbeitungsdatum des Musters.
- **Problem:** Im Rahmen der Problembeschreibung ist darauf zu achten, präzise und detailliert den Kern des Problems zu schildern. Es ist auf den Kontext des Problems einzugehen. Dieser Abschnitt des Musters enthält zudem eine Auflistung der wesentlichen Anforderungen an die zu erarbeitende Lösung.
- **Deshalb (Therefore):** In einem kurzen Absatz wird zunächst in Form einer kurzen Anleitung die Art des Lösungsweges beschrieben. Es folgt eine detaillierte Unterteilung sowie generische Ausarbeitung der Problemlösung. Zusätzlich ist im Rahmen dieser Kategorie auf Verbindungen zu ähnlichen Mustern, Lösungen oder Methoden hinzuweisen.
- **Aber (But):** In diesem Abschnitt erfolgt eine Beschreibung der Konsequenzen bei der Benutzung des betrachteten Musters. Hierzu ist es notwendig, explizite Anwendungsbeispiele aufzuzeigen. Idealerweise werden Beispiele hinterlegt, die variierende Konsequenzen beinhalten.
- **Siehe auch (See Also):** Diese Kategorie des Musters ist optional. Sie kann beispielsweise ein verwandtes Muster und dessen Kurzbeschreibung beinhalten.

**Bewertung:** Die vorgestellte Herangehensweise nach SALUSTRI ist in weiten Teilen sehr generisch. Die Kategorisierung entspricht zwar der verfolgten Definition nach ALEXANDER, jedoch ist die Beschreibungstiefe innerhalb eines Lösungsmusters sehr

---

<sup>46</sup>Vgl. Kapitel 2.4.4

gering. Die Repräsentation relevanten Wissens in Form von Modellen bleibt aus. SALUSTRI schlägt zwar eine disziplinunabhängige Anwendung vor, ein konkretes Beispiel mit zugehöriger Strukturierung der Lösungsmuster wird nicht vorgestellt. Ferner mangelt es an einer Erklärung über Relationen oder Gemeinsamkeiten zwischen fachdisziplinspezifischen Lösungsmustern. Dies ist jedoch erforderlich, um entsprechendes Wissen zusammenzuführen oder, falls vorhanden, zu erweitern.

### 3.2.3.3 Koordinationsmuster der Softwaretechnik nach DZIWOK

Die Wiederverwendung von Wissen in Form von Lösungsmustern ist vor allem in der Informatik sehr populär. Die Basis der Mehrzahl dieser Ansätze sind die Entwurfsmuster der „Gang of Four“. Eine Adaption auf die Produktentwicklung liefern DZIWOK ET AL. Sie definieren sog. Koordinationsmuster der Softwaretechnik für den Entwurf echtzeitkritischer Kommunikation in und zwischen intelligenten mechatronischen Systemen [DHT12].

Zur Wissensrepräsentation wird die Modellierungssprache MechatronikUML<sup>47</sup> verwendet. Für die Modellierungssprache gibt es einen systematischen Entwicklungsprozess, der die Entwickler bei der modelgetriebenen Softwareerstellung für mechatronische Systeme, unterstützt. Die Sprache eignet sich in erster Linie zur Abbildung von Struktur- bzw. Zustandsdiagrammen, wobei auf die Elemente der UML-Notation, wie beispielsweise Komponenten, zurückgegriffen wird. Komponenten besitzen fest definierte Schnittstellen, über die eine Verbindung zu weiteren Komponenten hergestellt werden kann. So genannte „Realtime-Statecharts“ spezifizieren das Verhalten dieser Schnittstellen (vgl. Bild 3-17). Eine wesentliche Eigenschaft ist ihre Übertragbarkeit auf weitere Komponenten. Dies ist Grundvoraussetzung zur Erfüllung der Mustertheorie und der damit verbundenen effizienten Wiederverwendbarkeit.

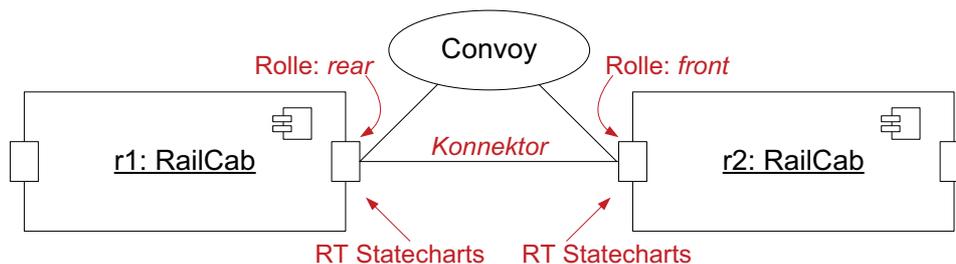


Bild 3-17: Realtime-Statecharts für die Konvoibildung zweier RailCabs

#### Beispielmuster Kooperative Konvoifahrt:

Im Folgenden wird exemplarisch ein Lösungsmuster für die Softwaretechnik an einem Anwendungsbeispiel erklärt. Der Demonstrator ist das sog. RailCab<sup>48</sup>, ein schienenge-

<sup>47</sup>Weitere Informationen sind zu finden in [BBB+12]

<sup>48</sup>Für eine nähere Beschreibung sei auf <http://www.railcab.de> verwiesen.

bundenes Verkehrssystem, das als Versuchsanlage im Maßstab 1:2,5 an der Universität Paderborn realisiert ist. In Anlehnung an das RailCab und die damit einhergehende Verbindung zur Struktur des OCM<sup>49</sup> finden die Statecharts im reflektorischen Operator Anwendung. Der Name des in Bild 3-18 dargestellten Lösungsmusters ist **Kooperative Konvoifahrt**. Es beschreibt die notwendige Koordination, um einen Konvoi von Fahrzeugen zu bilden und aufzulösen [GTS14, S. 114ff.].

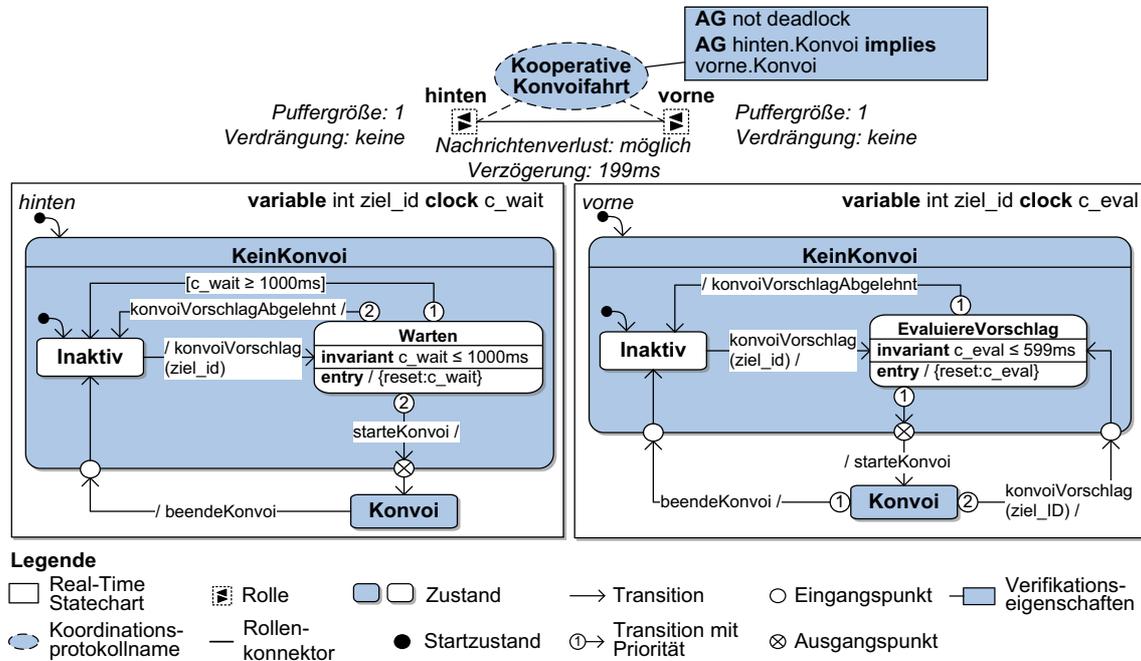


Bild 3-18: Zustandsdiagramme des Lösungsmusters „Kooperative Konvoifahrt“ [GTS14, S. 120]

Befinden sich zwei RailCabs im Zustand Konvoi, fahren diese in einem Abstand von ca. einem Meter bei 160 km/h hintereinander her. Eine mechanische Kopplung liegt nicht vor. Ziel ist es, einen möglichst geringen Energieverbrauch für das hintere RailCab zu erzielen. Für eine kontinuierliche Gewährleistung der Sicherheit ist es essentiell, dass das hintere RailCab vom vorderen RailCab über einen bevorstehenden Bremsvorgang in Echtzeit informiert wird. Das gewählte Lösungsmuster stellt genau dieses sicher.

Das Lösungsmuster setzt sich zusammen aus den beiden Rollen *rear* (Beschreibung für das hintere RailCab) und *front* (Beschreibung für das vordere RailCab). Diese beiden sog. Ports kommunizieren miteinander. Für jede Rolle ist das Verhalten formal durch je ein Real-Time Statechart spezifiziert. Zudem charakterisieren zwei weitere Verifikationseigenschaften das Lösungsmuster: 1. es darf kein Deadlock auftreten; 2. es muss die Bedingung erfüllt sein, dass das vordere RailCab immer im Konvoi-Modus ist, wenn das hintere RailCab im Konvoi-Modus ist. Letzteres bezieht sich dabei auf die Rollen und deren Zustände.

<sup>49</sup>Vgl. Kapitel 2.2.3

**Bewertung:** Die Koordinationsmuster werden für die Spezifikation ereignisdiskreter Echtzeitkoordination von Softwarekomponenten in mechatronischen Systemen genutzt. Die Darstellung der Inhalte in MechatronikUML unterstreicht ihren fachdisziplinspezifischen Charakter und ihren Einsatz unmittelbar im Anschluss an den Systementwurf. Aufgrund der zunehmenden Durchdringung technischer Systeme mit Software, sollte ein besonderer Fokus auf der Verzahnung der Koordinationsmuster mit den zu erarbeitenden Lösungsmustern für den Systementwurf liegen. Dieser existiert aktuell noch nicht.

#### 3.2.3.4 Ansätze der Regelungstechnik nach FÖLLINGER

Lösungsmuster in der Regelungstechnik sind weitaus weniger populär als beispielsweise Muster in der Informatik oder Softwaretechnik. Die Basis potentieller Lösungsmuster der Regelungstechnik geht zurück auf die Arbeiten von FÖLLINGER. Um diese nachvollziehen zu können, ist es zunächst notwendig, ein einheitliches Verständnis der Regelungstechnik bzw. einer Regelung herzustellen. Ziel einer Regelung ist, das Verhalten eines dynamischen Systems durch gezielte Eingriffe mittels Hilfseinrichtungen in gewünschter Weise zu beeinflussen. FÖLLINGER charakterisiert eine Regelung wie folgt:

*„Ausgangspunkt ist die Forderung nach selbsttätiger gezielter Beeinflussung eines dynamischen Systems, das nur unvollständig bekannt ist und das sein Verhalten in nicht genau vorhersagbarer Weise ändert, während des Betriebs. Die Erfüllung dieser Forderung erfolgt durch eine Rückführungsstruktur, bestehend aus einer Beobachtungseinrichtung, die laufend Informationen über das veränderliche Systemverhalten sammelt, und einer Beeinflussungseinrichtung, welche diese Informationen verarbeitet und dadurch so auf das System einwirkt, dass dessen Verhalten dem gewünschten Verhalten angeglichen wird. Eine solche Rückführungsstruktur heißt Regelung“ [Foe13, S. 1f.]*

Ein Schwerpunkt der Regelungstechnik liegt dabei auf der Informationsverarbeitung. Technisch realisiert wird diese jedoch durch Komponenten der Elektrotechnik und des Maschinenbaus. Um das Vorgehen und sämtliche Zusammenhänge im Rahmen der Entwicklung grafisch zu modellieren werden sog. Strukturbilder genutzt. Ein Strukturbild ist die Basis für die weitere Ausarbeitung einer Regelung, bzw. eines Regelkreises.

Eine detailliertere, jedoch sehr allgemein gehaltene Darstellung einer Regelung zeigt das Bild 3-19. Es enthält sowohl die wesentlichen Komponenten, wie Regelglied oder Stellglied, als auch sämtliche Bezeichnungen der relevanten Ein- und Ausgangsgrößen. Die Ausgangsgröße der sog. Strecke, bzw. des dynamischen Systems, ist die Regelgröße  $x$ . Diese wird fortlaufend von Sensoren (vgl. Kap. 2.2.1) erfasst und an die Messeinrichtung weitergeleitet. Analoge Messgrößen wie Temperatur oder Druck werden dabei in elektrische Größen umgewandelt, die als Rückführungsgröße  $r$  bezeichnet werden.

Die Rückführungsgröße wird mit einer von außen vorgegebenen Führungsgröße  $w$  verglichen. Innerhalb des Vergleichsgliedes erfolgt die Ermittlung der Regeldifferenz  $e$ . Hauptaufgabe des darauf folgenden Regelgliedes ist die Korrektur des dynamischen Verhaltens des Systems. Ohne das Regelglied besteht die Gefahr, dass die Regelung zu instabil, zu ungenau oder zu träge arbeitet. Hat der Vergleich der Führungsgröße  $w$  mit der Rückführungsgröße  $r$  ergeben, dass eine Differenz vorliegt, muss die Regelgröße an den Sollverlauf der Führungsgröße angeglichen werden. Hervorgerufen wird diese Differenz durch die auf die Strecke einwirkende Störgröße  $z$ . Die anschließende Anpassung der Werte wird durch die Stelleinrichtung realisiert, die sich in Steller und Stellglied unterteilt [DIN19226], [Foe13, S. 2f.].

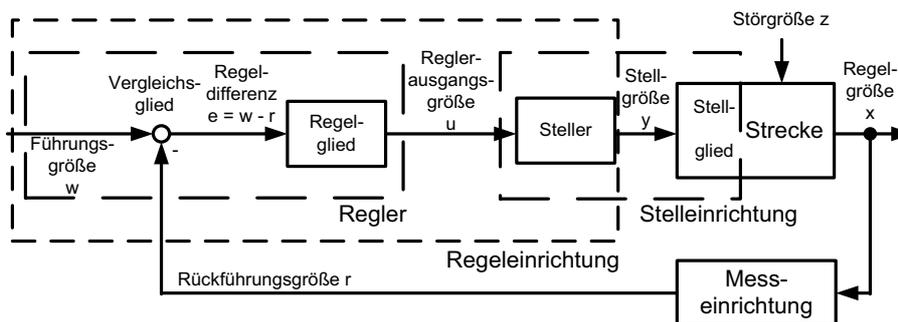


Bild 3-19: Abstraktes Blockschema einer Regelung [Foe13, S. 3]

Bei der Entwicklung einer Regelung geht es daher um Umwandlungen und Anpassungen von skalaren Größen. Diese erfolgen durch die sog. Übertragungsglieder, wie beispielsweise das Vergleichsglied, das durch elektrische Schaltungen realisiert wird. Komplexe anwendungsspezifische Regelungen setzen sich dabei stets aus einer Reihe elementarer Übertragungsglieder zusammen. Aus der vorgenommenen Anpassung der Eingangsgröße im Vergleich zur Ausgangsgröße resultieren sog. Übertragungsfunktionen, die durch mathematische Formeln festgehalten sind. Die anschließende Simulation einer entwickelten Regelung erfolgt durch Softwaretools und basiert auf mathematischen Modellen. Diese Modelle gehen dabei auf die Kombination der Übertragungsglieder mit den zugehörigen Übertragungsfunktionen zurück. Ein Auszug der elementarsten Übertragungsglieder zeigt die Tabelle 3-1 [Foe13, S. 32ff.].

Tabelle 3-1: Zusammenstellung der einfachsten Übertragungsglieder [Foe13, S. 45]

Benennung	Funktionalbezeichnung	Übertragungsfunktion	Sprungantwort (Null für $t < 0$ )	Verlauf der Sprungantwort	Symbol
P-Glied	$y = k u$	$k$	$k$		
I-Glied	$y = \int_0^t u(t) dt$	$\frac{k}{s}$	$k t$		

**Bewertung:** FÖLLINGER liefert mit der Darstellung einer Regelstrecke als Blockschaltbild und der Definition wesentlicher Übertragungsglieder erste Ansatzpunkte zur Ableitung von Lösungsmustern der Regelungstechnik. Sämtliche Ausarbeitungen beziehen sich ausschließlich auf die Anforderungen der Regelungstechnik. Analog zu den Überlegungen nach SUHM schlägt auch FÖLLINGER eine Aggregation der Probleme und zugehörigen Lösungen in Form von Übertragungsgliedern vor. Dieser etablierten Herangehensweise muss bei der Definition eines multidimensionalen Wissensraums Rechnung getragen werden.

### 3.2.3.5 Muster der Regelungstechnik nach SANZ/ZALEWSKI

Einen weiteren Ansatz für Lösungsmuster der Regelungstechnik liefern SANZ/ZALEWSKI. Sie betonen einen einheitlichen Aufbau von Lösungsmustern als Grundvoraussetzung für deren Verwendung. Dies gilt für alle Fachdisziplinen gleichermaßen. Die in der Literatur verwendeten Begriffe für den Aufbau von Lösungsmustern sind z.B. Schemata, Form, Struktur, Spezifikation, Template oder auch Steckbrief. Typischerweise werden alle benötigten Informationen über die Lösung und das zugrundeliegende Problem in unterschiedliche Kategorien gegliedert. Auf diese Weise lassen sich die unterschiedlichen Aspekte eines Musters gut strukturiert charakterisieren. Die Autoren dokumentieren dabei den Inhalt der Lösungsmuster im Hinblick auf den vorgesehenen Einsatzzweck. Die Repräsentation der Informationen kann somit sehr unterschiedlich ausfallen (je nach Einsatzzeitpunkt im Produktentstehungsprozess z.B. mit Hilfe von „natürlicher Sprache“ oder „formaler Modelle“). In der Regelungstechnik bietet es sich an, die Informationen in Form von Blockschaltbildern darzustellen. Besonders im Hinblick auf einen modellbasierten Entwurf ist der Detaillierungsgrad der Modelle äußerst variabel [SZ03].

Eine besondere Herausforderung sehen SANZ/ZALEWSKI in der Abgrenzung der unterschiedlichen Detailstufen. Diese sind nämlich nicht eindeutig voneinander zu unterscheiden, was eine Klassifikation sehr erschwert, wenn nicht sogar verhindert. Es existieren vielmehr kontinuierliche Übergänge, denen bei der Definition der Lösungsmuster und der Abbildung der Beziehungen, Rechnung getragen werden muss [SZ03].

SANZ/ZALEWSKI betonen die Gefahr, in der Entwicklung komplexer technischer Systeme zu früh in detaillierten Lösungen zu denken. Sie sehen in der Verwendung von Lösungsmustern großes Potential, dieser Gefahr entgegenzuwirken. Lösungsmuster sind keine detaillierten Lösungen. Es sind vielmehr generische Prinzipien, die individuell angepasst werden müssen. Die wesentlichen Vorteile bei der Verwendung von Lösungsmustern sind die Steigerung von Kreativität und Effizienz im Entwicklungsprozess. Ein Beispiel für ein Lösungsmuster (Feedback-Regelung) der Regelungstechnik nach SANZ/ZALEWSKI zeigt Bild 3-20. Es handelt sich im Kern um die bereits in Kapitel 3.2.3.4 vorgestellten Ansätze der Regelungstechnik nach FÖLLINGER [SZ03].

Name:	Feedback
Aliasnamen:	Closed-loop control, Feedback control, Output feedback
Problem:	Das System verhält sich nicht wie gewünscht. Die Reaktionsantwort ist z.B.: zu langsam, zu instabil, zu nichtlinear oder zu oszillierend.
Ziel:	Das Verhalten eines Systems (Output) folgt einer vorgegebenen Referenz (Set Point).
Kontext:	Das System hat einen sensorbaren Output und einen variablen Input. Es ist ein Referenzsignal vorhanden.
Herausforderungen:	a) Das Verhalten der Strecke wird durch Störungen beeinflusst. B) Genaue Kenntnisse über die Strecke sind nicht immer in der erforderlichen Detaillierung vorhanden.
Lösung:	Der Eingangswert der Strecke wird determiniert durch den Controller. Der Eingangswert für den Controller ergibt sich aus der Differenz zwischen der Führungsgröße und der Rückführungsgröße.
Diagramm:	
Begründung:	Die unmodellerte Dynamik der Strecke und die Störgrößen machen eine perfekte open-loop Regelung unmöglich. Die Sensierung des tatsächlichen Verhaltens ist unerlässlich.
Varianten:	PI-Regler; PID-Regler; PD-Regler
Bekannte Anwendungen:	.....
Verwandte Muster:	Feedforward

Bild 3-20: Lösungsmuster „Feedback“ nach SANZ/ZALEWSKI [SZ03]

**Bewertung:** SANZ/ZALEWSKI bilden in der Regelungstechnik eine Ausnahme, indem sie typische Problemstellungen der Regelungstechnik und zugehörige Lösungsansätze als Lösungsmuster titulieren und entsprechen dokumentieren. Die aufgezeigten Beispiele sind dabei größtenteils deckungsgleich mit den Arbeiten von FÖLLINGER. Bei der zu erarbeitenden einheitlichen Strukturierung müssen die gezeigten Ansätze der Regelungstechnik entsprechend berücksichtigt werden. Hinzu kommen Anforderungen von SANZ/ZALEWSKI an die Art der Wissensrepräsentation und die Besonderheiten bei dem Versuch, eine Lösungsmusterklassifikation zu erstellen. Die im Rahmen der vorliegenden Arbeit angestrebte multidimensionale Skalierung muss den Übergang von der natürlichen Sprache hin zu formalen Modellen sowie dem kontinuierlich ansteigenden Detaillierungsgrad gerecht werden.

### 3.3 Systematiken für den Einsatz von Lösungsmustern

Ein lösungsmusterbasierter Systementwurf erfordert eine geeignete Systematik. Die Problemanalyse hat gezeigt, dass die Systematik zwei Handlungsfelder abdecken muss. Zum einen bedarf es einer methodischen Unterstützung wie auf Basis bestehender Systeme bzw. Teilsysteme relevantes Lösungswissen identifiziert und in Lösungsmuster überführt werden kann<sup>50</sup>. Zum anderen ist ein Entwicklungsleitfaden notwendig, der die

<sup>50</sup> Der Fokus im Rahmen dieser Arbeit liegt bei der Identifizierung von Lösungsmustern auf dem methodischen Vorgehen. Er ist daher von rechnerunterstützten Ansätzen aus der Softwaretechnik, die unter den Begriffen Data Mining oder Pattern Mining bekannt sind, zu differenzieren [HK01].

Nutzung der im Rahmen dieser Arbeit zu entwickelnden Lösungsmuster allgemeinverständlich widerspiegelt. Aus diesen Gründen beschäftigt sich Kapitel 3.3.1 mit einer Methode zur Identifizierung von Systemarchitekturmustern. Kapitel 3.3.2 und 3.3.3 untersuchen Vorgehen für einen lösungsmusterbasierten Systementwurf.

### 3.3.1 Identifizierung von Systemarchitekturmustern nach KALAWSKY

KALAWSKY beschäftigt sich in seinen Arbeiten mit Systemarchitekturmustern. Er untersucht einerseits die Erstellung von Systemarchitekturen mit Hilfe von Lösungsmustern und andererseits die Identifizierung von Systemarchitekturmustern auf Basis bestehender Systeme. Bild 3-21 präsentiert ein von KALAWSKY erarbeitetes Vorgehensmodell zur Definition von Systemarchitekturmustern. Es ist die komprimierte Abbildung eines hochgradig iterativen Prozesses. Dieser ist geprägt durch die Kommunikation und Kooperation des Systemarchitekten mit den Experten der in der Systementwicklung involvierten Fachdisziplinen. Dem Systemarchitekten wird eine Schlüsselrolle bescheinigt, denn er ist für die fachdisziplinübergreifenden Zusammenhänge zwischen den Teilsystemen verantwortlich [KJT+13].

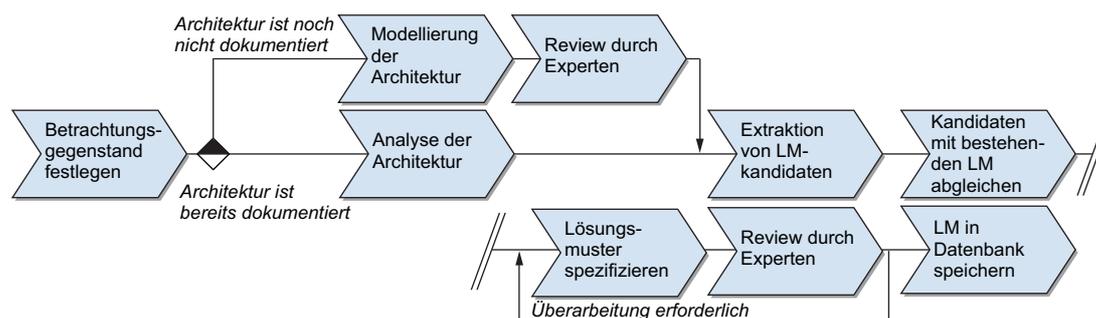


Bild 3-21: Vorgehensmodell zur Identifizierung von Systemarchitekturmustern nach KALAWSKY [KJT+13, S. 289]

Der erste Schritt im Vorgehensmodell zur Identifizierung von Systemarchitekturmustern ist die Festlegung des Betrachtungsgegenstandes. Es bildet die Grundlage aller folgenden Tätigkeiten. Anschließend gilt es zu untersuchen, ob für den Betrachtungsgegenstand (z.B. Gesamtsystem, oder nur bestimmte Teilsysteme) eine modellbasierte Spezifikation vorliegt. Hinweise auf die Verwendung einer bestimmten Spezifikationstechnik werden nicht gegeben. Das weitere Vorgehen wird hinsichtlich zweier unterschiedlicher Ausgangssituationen bestimmt: Liegt eine Spezifikation vor, so kann direkt mit der Analyse begonnen werden; liegt keine Spezifikation vor, so ist die Architektur entsprechend zu modellieren und anschließend mit Experten zu diskutieren [KJT13].

Darauf aufbauend folgt die Extraktion von sog. Lösungsmusterkandidaten. Diese geben erste Anhaltspunkte für ein potentielles Lösungsmuster. Es handelt sich um eine detaillierte Lösung, die noch nicht auf den Problem- und Lösungskern abstrahiert ist. Um Dopplungen zu vermeiden, wird daher zunächst ein Abgleich mit existierenden Lösungsmustern vorgenommen. Handelt es sich um ein neuartiges Lösungsmuster, so wird

dieses durch den Systemarchitekten entsprechend spezifiziert. Da an einem Systemarchitekturmuster unterschiedliche Fachdisziplinen beteiligt sein können, ist ein Review des erarbeiteten Lösungsmusters mit den Experten unumgänglich. Die Folge können mehrere Iterationen sein, bis alle erforderlichen Informationen im Lösungsmuster spezifiziert sind. Den Abschluss bildet die langfristige Sicherung des erstellten Lösungsmusters in einer Datenbank [KJT13].

**Bewertung:** Das Vorgehensmodell zur Identifizierung von Systemarchitekturmustern ist sehr generisch. Es beschreibt die durchzuführenden Tätigkeiten und die Vorgehensweise des Systemarchitekten, der aufgrund seines fachdisziplinübergreifenden Know-Hows die Schlüsselperson ist. Dem Ansatz von KALAWSKY mangelt es jedoch an nachvollziehbaren Beispielen in Bezug auf das Handlungsfeld komplexer technischer Systeme. Dies betrifft sowohl die zu verwendende Spezifikationstechnik, eine Lösungsmusterstruktur als auch die Systemarchitekturmuster. Es wird nicht ersichtlich, wie der iterative Prozess von einer Systemspezifikation über Lösungsmusterkandidaten hin zu Lösungsmustern durchzuführen ist. Das Vorgehensmodell liefert lediglich einen groben Rahmen.

### 3.3.2 Musterbasierter Entwurf mechatronischer Systeme nach GAUSEMEIER ET AL.

Eine generische Vorgehensweise für den lösungsmusterbasierten Entwurf mechatronischer Systeme liefern GAUSEMEIER ET AL. Dabei wird zwischen Lösungsmustern und Lösungselementen unterschieden. GAUSEMEIER unterteilt Lösungsmuster je nach Disziplin in Wirkprinzipien und Wirkmuster<sup>51</sup> [ADG+09, S. 173ff.]. Diese werden mit zunehmender Konkretisierung durch Lösungselemente, also Katalog- oder Normteile, ersetzt. Für einen effizienten Zugriff und eine langfristige Sicherung der Lösungsmuster sowie -elemente wird ein Semantic Web erarbeitet. Bild 3-22 zeigt schematisch den Entwurf mechatronischer Systeme unter Zuhilfenahme von Lösungswissen aus dem Semantic Web [GAC+13].

Der fachdisziplinübergreifende Entwurf mechatronischer Systeme beginnt mit der Zielbestimmung. Im Rahmen der Aufgabenanalyse wird die Aufgabenstellung abstrahiert, um den Kern der Entwicklungsaufgabe zu identifizieren. Anschließend wird eine Umfeldanalyse durchgeführt, in der die Randbedingungen und Einflüsse auf das System ermittelt werden. Die gesammelten Erkenntnisse werden nach der Definition aller möglichen Anwendungsszenarien in einer Anforderungsliste festgehalten. Diese ist Basis für die Definition der erforderlichen Funktionalität des Gesamtsystems in Form einer Funktionshierarchie [GAC+13].

---

<sup>51</sup>Einige ausgewählte Beispiele befinden sich im Anhang A1.3.

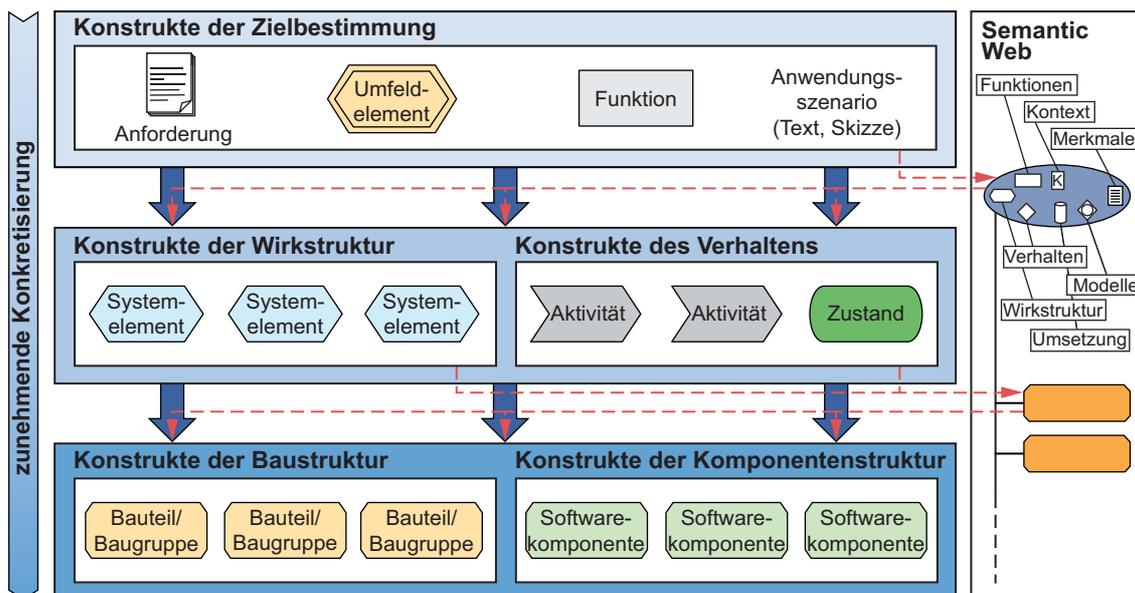


Bild 3-22: Systematik für die Nutzung von Lösungsmustern und -elementen in der Entwicklung mechatronischer Systeme [GAC+13]

Auf Grundlage der Zielbestimmung erfolgt die Auswahl von Lösungsmustern. Eingangsinformationen sind die spezifizierten Funktionen und Anforderungen. Die anschließende Auswahl und Kombination potentieller Lösungsmuster erfordert kreatives Denken und Know-how der beteiligten Ingenieure. Auf Basis der ausgewählten Lösungsmuster werden die Wirkstruktur, Verhaltensmodelle und eine erste Grobgestalt entwickelt. Dies geschieht unter Zuhilfenahme der in den Lösungsmustern bereitgestellten Informationen. Diese werden genutzt und je nach zugrundeliegendem Entwurfsproblem individuell konkretisiert. Erfüllen die gefundenen Lösungsmuster nicht alle im Rahmen der Zielbestimmung spezifizierten Funktionen, so müssen zusätzliche Lösungskonzepte erarbeitet werden. Je nach Größe des Gesamtsystems wird dieses bereits während des Systementwurfs in Module unterteilt. Die einzelnen Module werden daraufhin verfeinert und anschließend wieder zusammengeführt. Das Ergebnis ist die Prinziplösung [GAC+13].

Es folgt der fachdisziplinspezifische Entwurf, der mit der Auswahl der entsprechenden Lösungselemente beginnt. Hierzu werden die in der Prinziplösung enthaltenen Informationen genutzt, in die Terminologien der beteiligten Disziplinen übersetzt und anschließend detailliert. So werden beispielsweise detaillierte Parameterbereiche (Bauraum, max. Drehzahl etc.) festgelegt. Diese Informationen sind notwendig, um im Semantic Web geeignete Lösungselemente auszuwählen, die den verwendeten Lösungsmustern durch semantische Annotationen zugeordnet sind. Mit Auswahl der Lösungselemente endet der Zugriff auf Lösungswissen im Semantic Web. Es folgt der fachdisziplinspezifische Entwurf, dessen Ergebnis durch die Bau- und Komponentestruktur repräsentiert wird [GAC+13].

**Bewertung:** GAUSEMEIER ET AL. liefern ein allgemeingültiges Vorgehen für einen musterbasierten Systementwurf. Ausgangspunkt sind Informationen aus der strategischen Produktplanung mit ersten Anforderungen an das zu entwickelnde System. Das Ergebnis ist das entwickelte System, repräsentiert durch Bau- und Komponentenstruktur. Dabei wird auf Lösungswissen – Lösungsmuster und -elemente – aus dem Semantic Web zurückgegriffen. Vor dem Hintergrund der zu erfüllenden Anforderungen an das Instrumentarium ist zu klären, ob die Unterteilung in Lösungsmuster und -elemente ausreichend ist. Es werden beispielhaft Lösungsmuster aus den verschiedenen Disziplinen beschrieben, ihre systematische Herleitung, einheitliche Strukturierung sowie Anordnung im Wissensraum fehlt jedoch. Einer detaillierten Beschreibung des fortlaufenden Wechselspiels zwischen einem Entwurfsraum und möglichen Wissensraum wird der analysierte Ansatz nicht gerecht.

### 3.3.3 Musterbasierter Entwurf der selbstoptimierenden Informationsverarbeitung nach DUMITRESCU

Zusätzlich zu der einheitlichen Strukturierung der Lösungsmuster für Selbstoptimierung beschreibt DUMITRESCU mit Hilfe eines Vorgehensmodells (vgl. Bild 3-23) den musterbasierten Entwurf der s.o. Informationsverarbeitung. Übergeordnetes Ziel ist die Integration kognitiver Funktionen in mechatronische Systeme unter Berücksichtigung der in der VDI2206 festgehaltenen Vorgehensweise für den Systementwurf. Der Fokus liegt dabei auf dem Entwurf der intelligenten Informationsverarbeitung. Weitere Disziplinen wie z.B. der Maschinenbau finden nur partiell Berücksichtigung [Dum10].

Das Vorgehensmodell, gegliedert in Phasen und Meilensteine, gibt einen Überblick über die wesentlichen Tätigkeiten und zu erarbeitenden Resultate. Die Darstellung ist idealtypisch. Die Anwendung ist geprägt durch Iterationen und Rücksprünge in vorangegangene Phasen. Die einzelnen Phasen und die durchzuführenden Tätigkeiten werden im Folgenden erläutert. Ausgangspunkt für das Vorgehensmodell ist ein nicht vollständig konzipiertes mechatronisches System. Das zu entwickelnde System enthält noch keine spezifizierte Informationsverarbeitung [Dum10].

**Systemanalyse:** In der ersten Phase wird der aktuelle Entwicklungsstand des mechatronischen Systems analysiert. Ziel ist die Aufdeckung von Nutzenpotentialen für kognitive Funktionalitäten im System. Hierzu schlägt DUMITRESCU im Kern eine detaillierte Ausarbeitung des Zielsystems (vgl. Kapitel 2.2.3) unter Berücksichtigung von Einflüssen aus dem Systemumfeld vor. Liegen als Ergebnis Potentiale für kognitive Funktionen vor, folgen die Phasen 2 bis 4. Hat die Analyse ergeben, dass keine Potentiale vorliegen, ist das Vorgehensmodell beendet und es wird mit klassischen Entwurfsmethoden (z.B. VDI 2206) fortgefahren.

**Funktionssynthese:** Kern der zweiten Phase ist die Spezifikation notwendiger kognitiver Funktionen. Mit Hilfe eines Funktionsverbenkatalogs ermöglicht es DUMITRESCU Entwicklern aus technischen Disziplinen die zu erarbeitende Informationsverarbeitung

funktional zu beschreiben. Die Funktionen werden in einer Funktionshierarchie abgebildet. Eine Entwurfsschablone erleichtert die Spezifikation und ist auf höchster Ebene gegliedert in die wesentlichen Phasen des Selbstoptimierungsprozesses (vgl. Kapitel 2.2.3).

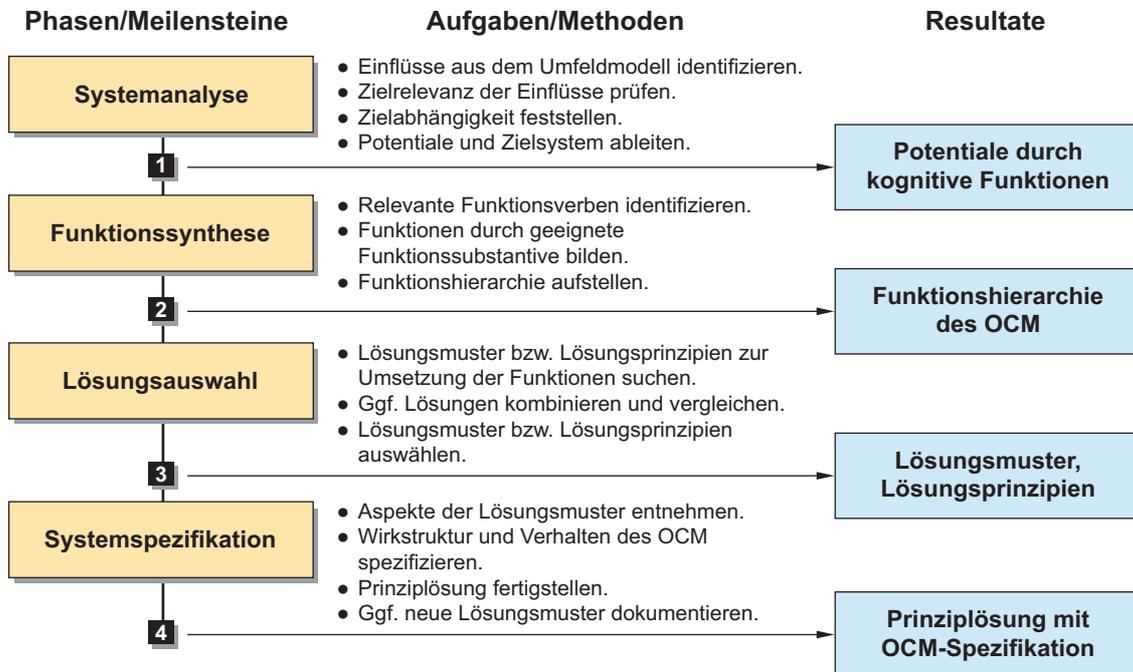


Bild 3-23: Vorgehensmodell für den lösungsmusterbasierten Entwurf nach DUMITRESCU [Dum10, S. 100]

**Lösungsauswahl:** Im Hinblick auf das zu entwickelnde Instrumentarium ist die dritte Phase des Vorgehensmodells wesentlich. Auf Basis der zuvor erarbeitenden Funktionshierarchie werden geeignete Lösungsmuster gesucht und ausgewählt. Der Abgleich erfolgt dabei über die Funktionshierarchie und den gleichnamigen Aspekt in der Strukturierung der Lösungsmuster für Selbstoptimierung (vgl. Kapitel 3.2.2.3). Im Falle eines Misserfolgs bei der Suche über die Funktionen schlägt DUMITRESCU eine weitere Suchmöglichkeit vor. Demnach ist es auch möglich, direkt nach Verfahren zu suchen, auf denen die Lösungsmuster für Selbstoptimierung beruhen.

**Systemspezifikation:** Ziel der letzten Phase ist die Komplettierung der Prinziplösung des Gesamtsystems. Hierzu wird mit Hilfe der in den Lösungsmustern enthaltenen Informationen die Wirkstruktur, die Aktivitäten sowie die Zustände spezifiziert. Der Schwerpunkt liegt dabei auf der Konzipierung der Informationsverarbeitung, die in geeigneter Weise mit dem jeweiligen Entwicklungsstand des Grundsystems zusammenzuführen ist. Die entwickelte Spezifikation in Anlehnung an die Strukturierung des OCM (vgl. Kapitel 2.2.3) und die ausgewählten Lösungsprinzipien bilden die Grundlage für die Implementierung der kognitiven Informationsverarbeitung durch die entsprechenden Experten im späteren Verlauf der Entwicklung.

**Bewertung:** Das Vorgehensmodell für einen musterbasierten Entwurf der Informationsverarbeitung fortgeschrittener mechatronischer Systeme ist gut auf die zugehörige Strukturierung der Lösungsmuster abgestimmt (vgl. Kapitel 3.2.2.3). Es wird dem Anspruch gerecht, Dritte in die Lage zu versetzen, eine kognitive Informationsverarbeitung zu spezifizieren. Dem Vorgehensmodell mangelt es jedoch an einer allgemeinverständlichen Darstellung zur Identifizierung und Definition von Lösungsmustern. Ferner ist der Betrachtungshorizont auf Ansätze zur Integration von Kognition im Hinblick auf einen ganzheitlichen Systementwurf zu eng gefasst.

### 3.4 Bewertung und Handlungsbedarf

Ein Vergleich des Stands der Technik mit den abgeleiteten Anforderungen aus Kapitel 2 führt zu folgender Bewertung, die Bild 3-24 zusammenfasst:

**A1) Ganzheitlichkeit:** Eine sehr umfangreiche Übertragung des Paradigmas der Lösungsmuster auf die Produktentstehung bzw. das Systems Engineering liefert CLOUTIER. Diese gilt es auf etablierte Strukturierungskonzepte, wie etwa das 3-Zyklen-Modell der Produktentstehung nach GAUSEMEIER, zu übertragen und zu erweitern.

**A2) Charakterisierung von Lösungsmustern:** Einen Anhaltspunkt für die Charakterisierung von Lösungsmustern liefert GRABOWSKI mit der UDT. Weitere Arbeiten, die im Hinblick auf diese Anforderung relevant sind, beschreiben Lösungsmuster und ihre Wirkbeziehungen auf einer Meta-Ebene. Dazu zählen die Arbeiten von PFISTER und SIMPSON. Die vorliegenden Erkenntnisse sind bei der Charakterisierung von Lösungsmustern und ihrer Abgrenzung untereinander (Objekt-/ Prozessmuster, verträgliche Kombination von LM etc.) zu berücksichtigen und zu konkretisieren.

**A3) Einheitliche Strukturierung nach ALEXANDER:** Der Großteil der untersuchten Lösungsmusteransätze orientiert sich an der Definition eines Lösungsmusters nach ALEXANDER. Ferner wird die Notwendigkeit der Interdisziplinarität und des Zusammenwirkens unterschiedlicher Disziplinen betont. In diesem Zusammenhang präsentiert ausschließlich DUMITRESCU eine einheitliche Strukturierung von Lösungsmustern. Diese ist jedoch auf wenige Lösungsansätze zur Entwicklung einer kognitiven Informationsverarbeitung abgestimmt. Der Informationsumfang der präsentierten Lösungsmuster ist im Hinblick auf einen ganzheitlichen Systementwurf zu hoch. Es ist daher eine Strukturierung erforderlich, die sich auf die wesentlichen Informationen konzentriert. Weitere Informationen wie z.B. über Wirkbeziehungen sind gesondert darzustellen.

**A4) Interdisziplinarität und Verständlichkeit:** Als Spezifikationstechniken zur Darstellung des relevanten Lösungswissens und zur Modellierung komplexer technischer Systeme eignet sich in erster Linie CONSENS. Neben der Repräsentation einzelner Aspekte werden auch bestehende Querbeziehungen zwischen den Aspekten und Beziehungen zu fachdisziplinspezifischen Modellen beschrieben.

**A5) Durchgängige Dokumentation:** Erste Überlegungen für eine durchgängige Dokumentation liefert SUHM. Er definiert einen Wissensraum mit unterschiedlichen Dimensionen, jedoch abgestimmt auf die Bedürfnisse des Maschinenbaus. Diese Dimensionen sind aufzugreifen und auf die Besonderheiten im fachdisziplinübergreifenden Systementwurf anzupassen. Gefordert ist ein multidimensionaler Wissensraum, in dem Lösungsmuster für den Systementwurf durchgängig dokumentiert werden können.

Bewertungsskala: ○ = nicht erfüllt ◐ = teilweise erfüllt ● = voll erfüllt  Legende: LM = Lösungsmuster PE = Produktentstehung s.o. = selbstoptimierend SE = Systems Engineering IV = Informationsverarbeitung		Anforderungen									
		Ganzheitlichkeit	Charakterisierung	Einheitliche Strukturierung	Interdisziplinarität und Verständlichkeit	Durchgängige Dokumentation	Systematische Vorgehensweise	Geschlossener SECI-Kreislauf	Abstraktion und Generalisierung	Orientierung an etablierten Standards	
		A1	A2	A3	A4	A5	A6	A7	A8	A9	
Überblick über Lösungsmuster in der PE	Universal Design Theory nach GRABOWSKI	◐	◐	◐	●	◐	○	○	○	●	
	Musterhierarchie nach CLOUTIER	●	◐	◐	○	○	○	○	○	○	
	SE Entwurfsmuster-Metamodell nach PFISTER	●	●	○	◐	○	●	◐	○	●	
	SE Mustersprache nach SIMPSON	●	◐	○	◐	○	○	○	○	○	
Lösungswissen für den Systementwurf	Techniken für die Wissensrepräsentation	CONSENS	●	◐	○	●	○	●	○	◐	●
		UML	○	◐	○	○	○	○	○	○	○
		SysML	◐	○	○	◐	○	○	○	◐	○
		PREMISE	○	○	○	◐	○	○	○	○	○
		Modelica®	○	○	○	◐	○	○	○	◐	○
	Disziplinübergreifende LM	Systemarchitekturmuster nach CLOUTIER	○	◐	●	◐	○	○	○	○	○
		Lösungsmuster mit SysML nach WEILKINS	○	○	○	◐	○	○	○	○	○
		Lösungsmuster für selbstoptimierende Systeme nach DUMITRESCU	◐	◐	●	●	○	●	◐	○	●
	Disziplinspezifische LM	Lösungsmuster im Maschinenbau nach SUHM	○	◐	●	○	◐	○	◐	○	●
		Entwurfsmuster nach SALUSTRI	○	○	●	○	○	○	○	○	○
		Koordinationsmuster der Softwaretechnik nach DZIWOJ	○	◐	●	○	○	◐	○	○	○
		Ansätze der Regelungstechnik nach FÖLLINGER	○	◐	○	○	○	◐	○	○	○
		Muster der Regelungstechnik nach SANZ/ZALEWSKI	○	○	●	○	○	○	○	○	○
Systematiken für den Einsatz von LM	Identifizierung von Systemarchitekturmustern nach KAWALSKI	●	○	○	●	○	●	◐	●	○	
	Musterbasierter Entwurf mechatronischer Systeme nach GAUSEMEIER	◐	◐	○	●	○	●	◐	○	●	
	Musterbasierter Entwurf s.o. IV nach DUMITRESCU	◐	◐	●	●	○	●	◐	○	●	

Bild 3-24: Bewertung des untersuchten Stands der Technik anhand der Anforderungen

**A6) Systematische Vorgehensweise:** Die Systematik für den Einsatz von Lösungsmustern muss eine zielgerichtete und systematische Vorgehensweise darstellen. Dieses gilt für die Identifizierung von Lösungsmustern und einen lösungsmusterbasierten Systementwurf gleichermaßen. Hierzu bedarf es Vorgehensmodelle, die notwendige Phasen, Hilfsmittel und Resultate beschreiben. Keiner der untersuchten Ansätze erfüllt in Bezug auf den Einsatz von Lösungsmustern eine derart umfangreiche Vorgehensweise.

**A7) Geschlossener SECI-Kreislauf:** Da sich etablierte Systematiken im Kontext Wissensmanagement an dem SECI-Kreislauf orientieren, muss das angestrebte Instrumentarium dieser Anforderung gerecht werden. Keiner der untersuchten Ansätze erfüllt isoliert betrachtet diese Anforderung in Gänze. Es ist ein holistischer Ansatz zu erarbeiten, der die Identifizierung, Dokumentation und Anwendung von Lösungsmustern zyklisch durchläuft.

**A8) Abstraktion und Generalisierung:** Ein Vorgehen zur Identifizierung und Ableitung von Lösungsmustern, das dieser Anforderung gerecht wird, liefert KALAWSKY. Dieses ist auf die Entwicklung fortgeschrittener mechatronischer Systeme zu übertragen und zu detaillieren. Ferner sind die Zusammenhänge zum multidimensionalen Wissensraum und der damit verbundenen langfristigen Dokumentation der identifizierten Lösungsmuster herauszustellen.

**A9) Orientierung an etablierten Standards im Systementwurf:** Von den untersuchten Ansätzen zu den Lösungsmustern orientieren sich vorrangig DUMITRESCU und GAUSEMEIER ET AL. an etablierten Standards für den Systementwurf. Die erzielten Ergebnisse sind aufzugreifen und im Vorgehensmodell für einen lösungsmusterbasierten Systementwurf zu berücksichtigen. Der Fokus muss dabei auf dem systematischen Zusammenspiel zwischen den Tätigkeiten im Entwurf und dem zu definierenden Wissensraum liegen.

Keiner der untersuchten Ansätze, als auch keine Kombination bestehender Ansätze erfüllt alle Anforderungen in vollem Umfang. Ein entscheidender Mangel ist die unzureichende Verzahnung zwischen Lösungsmustern für den Systementwurf und deren Einbettung in den Gesamtkontext Produktentstehung. Ferner fokussieren viele Ansätze nur Teilaspekte des Gesamtproblems wie z.B. die Typisierung von Lösungsmustern auf sehr hohem Abstraktionsniveau (Objektmuster, Prozessmuster etc.). Eine einheitliche Strukturierung von Lösungsmustern für den Systementwurf und eine zugehörige multidimensionale Skalierung zur durchgängigen Dokumentation, fehlen. Zudem existiert keine Systematik, die Angaben darüber enthält, wie Lösungsmuster für den Systementwurf identifiziert werden und wie mit Hilfe von Lösungsmustern ein fortgeschrittenes mechatronisches System fachdisziplinübergreifend entworfen wird. Es besteht demnach dringender Handlungsbedarf nach einem *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*.

## 4 Instrumentarium für einen Lösungsmusterbasierten Entwurf

Der aus Kapitel 3 resultierende Handlungsbedarf verdeutlicht, dass es derzeit kein *Instrumentarium für einen Lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* entsprechend den in Kapitel 2 definierten Anforderungen gibt. Ein solches Instrumentarium bildet den Kern der vorliegenden Arbeit. Es wird in diesem Kapitel auf theoretischer Basis erläutert. Kapitel 4.1 gibt einen Überblick über das Instrumentarium und seine wesentlichen Bestandteile. Die Basis bildet eine Beschreibung des Paradigmas der Lösungsmuster in der Produktentstehung. Dieses ist Gegenstand von Kapitel 4.2. Anschließend findet eine Fokussierung auf den Systementwurf statt. Kapitel 4.3 stellt eine einheitliche Strukturierung von Lösungsmustern vor. Zudem werden die Lösungsmuster anhand mehrerer Dimensionen charakterisiert, um darauf aufbauend die Zusammenhänge in dem entwickelten multidimensionalen Wissensraum allgemeinverständlich abzubilden. Es folgt eine Systematik für den Einsatz von Lösungsmustern. Kapitel 4.4 beinhaltet ein Vorgehensmodell für die Identifizierung von Lösungsmustern auf der Basis etablierter Systeme. In Kapitel 4.5 wird die Vorgehensweise für einen Lösungsmusterbasierten Systementwurf beschrieben. Die Verifikation des Instrumentariums, insbesondere der Systematik, ist anschließend Gegenstand von Kapitel 5.

### 4.1 Das Instrumentarium im Überblick

Das *Instrumentarium für einen Lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* besteht aus vier Bestandteilen, die in diesem Kapitel erläutert werden. Es bildet den Kern der vorliegenden Arbeit. Das Instrumentarium hat den Anspruch, dem aufgezeigten Handlungsbedarf aus der Problemanalyse gerecht zu werden und setzt sich daher wie folgt zusammen (Bild 4-1):

- Eine umfangreiche Übertragung **des Paradigmas der Lösungsmuster auf die Produktentstehung** bildet die Grundlage. Hierzu werden die unterschiedlichen Lösungsmuster beschrieben und in einem Wirkgefüge vereint.
- **Lösungswissen für den Systementwurf** wird im Hinblick auf eine geeignete **Dokumentation** einheitlich strukturiert. Relevante Artefakte für den Systementwurf lassen sich so fachdisziplinübergreifend in unterschiedlichen Spezialisierungs- und Aggregationsniveaus repräsentieren. Etablierte Techniken für die Modellierung von Systembestandteilen werden genutzt und geeignet erweitert.
- Ein **Vorgehensmodell zur Identifizierung von Lösungsmustern**, das durchzuführende Tätigkeiten, deren Ergebnisse sowie notwendige Hilfsmittel definiert, um auf Basis von realisierten Systemen, Lösungsmuster zu identifizieren und zu dokumentieren.

- Ein weiteres **Vorgehensmodell für einen lösungsmusterbasierten Systementwurf**, versetzt zudem Dritte in die Lage, sukzessive unter Verwendung etablierter Lösungswissens ein Systemkonzept zu erstellen.

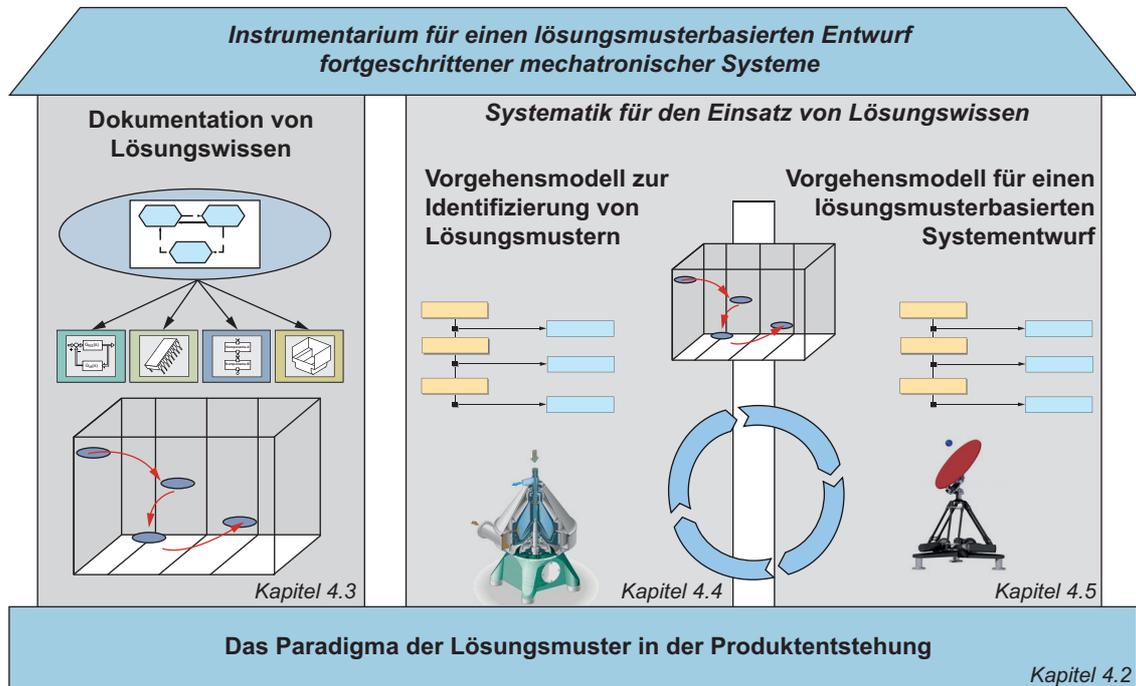


Bild 4-1: Aufbau des Instrumentariums

Angesichts der Erkenntnisse aus der Problemanalyse ist das Fundament des angestrebten Instrumentariums eine einheitliche Definition des Lösungsmusterbegriffs, die alle Bereiche der Produktentstehung umfasst. Des Weiteren sind die Fragen zu klären, ob Lösungsmuster anhand etablierter Strukturierungskonzepte für die Produktentstehung gegliedert werden können und welche Beziehungen zwischen den Lösungsmustern entstehen. Besonders die Prinziplösung eines Systems im Sinne der VDI2206 als Ergebnis der Konzipierung bzw. des Systementwurfs steht im Fokus dieses Instrumentariums. Ziel ist die frühzeitige Spezifikation des Systems als abstraktes und fachdisziplinübergreifendes Abbild des zu entwickelnden Systems. Angesichts der Analyse des Stands der Technik wird zu diesem Zweck auf die Spezifikationstechnik CONSENS zurückgegriffen. Diese wird zur Erfüllung der gestellten Anforderungen um spezifische Aspekte ergänzt. Die erarbeiteten Vorgehensmodelle ergeben in Kombination eine Systematik für den Einsatz von Lösungswissen im Systementwurf.

## 4.2 Das Paradigma der Lösungsmuster in der Produktentstehung

Eine konsequente Übertragung der Lösungsmuster auf die Produktentstehung erfordert eine einheitliche Begriffsdefinition, die alle partizipierenden Bereiche einschließt. Diese ist Gegenstand von Kapitel 4.2.1. Darauf aufbauend wird in Kapitel 4.2.2 eine Klassifikation der Lösungsmuster in der Produktentstehung vorgestellt. Hierzu werden bei-

spielhaft unterschiedliche Lösungsmuster präsentiert. Abschließend stellt Kapitel 4.2.3 das Wirkgefüge der Lösungsmuster auf oberster Gliederungsebene vor.

#### 4.2.1 Lösungsmuster – Begriffsklärung

Der Begriff Lösungsmuster setzt sich zusammen aus den Begriffen **Lösung** und **Muster**. Beide Begriffe haben jeweils mehrere Bedeutungen, was zwangsläufig zu diversen Interpretationen führt. Für die vorliegende Arbeit ist folgende Bedeutung des Begriffs Lösung relevant: Eine **Lösung** ist ein durch die Anwendung *spezieller Kenntnisse* und Methoden ermitteltes *Ergebnis, Verfahren, Vorgehen* zur Bewältigung einer *Schwierigkeit*. Der Begriff **Muster** kommt vom lateinischen Wort „*mostra*“ und bedeutet übersetzt so viel wie Schau oder Probe. Die Bedeutungen sind sehr vielseitig: *Warenprobe* bzw. *Warenmuster*; aus einzelnen Motiven bestehende *flächige Verzierung, Stoff* etc.; *Vorlage* nach der *etwas hergestellt, gemacht*, wird. Vor dem Hintergrund des angestrebten Instrumentariums ist die letztgenannte Interpretation wesentlich. In einer Verfeinerung heißt es zudem: Ein **Muster** ist eine *regelmäßige* Anordnung oder Darstellung elementarer psychophysischer Abläufe oder auch bewusster Handlungen, v.a. als Struktur und Erregungs- und Empfindungsabläufen oder Verhaltensfolgen; dem engl. Wort „*pattern*“ nachgebildet [Dud14-01].

Aufbauend auf diesen grundlegenden Bedeutungen und den Interpretationen von ALEXANDER, RISING, CLOUTIER etc. (vgl. Kapitel 2.4.4 und 2.4.5) lässt sich für die in der Produktentstehung betrachteten Lösungsmuster folgende Definition formulieren:

**Lösungsmuster** resultieren aus durch den Menschen hervorgerufenen *Regelmäßigkeiten* oder *regelmäßigen Wiederholungen* von Lösungen (Vorgehen, Methode, Verfahren, spezielle Kenntnis, Ergebnis) für bestimmte Problemstellungen. Ein Lösungsmuster umfasst eine explizite und generalisierte Beschreibung eines *Problems* sowie *der zugehörigen Lösung* (Problem-Lösungs-Paar). Lösungsmuster unterstützen den Menschen bei der *Erzeugung von Artefakten*. Artefakte in der Produktentstehung sind u.a. *Objekte* (Organisationsstruktur, CAD-Modell, Systemmodell, Softwarecode, Arbeitsplan etc.) oder *Prozesse* (Ablauforganisation, Vertriebsprozess, Entwicklungsprozess etc.).

#### 4.2.2 Klassifikation von Lösungsmustern in der Produktentstehung

Eine durchgängige Anwendung von Lösungsmustern in der Produktentstehung erfordert eine umfassende Hierarchisierung bzw. Klassifizierung. Diese muss über die bisher vielfach beschriebenen Anwendungsgebiete von Lösungsmustern – Einsatz im Rahmen der Entwicklung – hinausgehen und sich an etablierten Ordnungsschemata orientieren. Einen ersten Anhaltspunkt für eine Klassifikation liefern die in der PE durchzuführenden

den Geschäftsprozesse. In Bezug auf die in Kapitel 4.2.1 formulierte Definition von Lösungsmustern liefern die (Teil-)Ergebnisse von Geschäftsprozessen Anhaltspunkte für relevante Artefakte und somit Einsatzpotential für Lösungsmuster.

SCHMELZER ET AL. unterscheiden auf oberster Ebene *primäre* und *sekundäre* Geschäftsprozesse [SS10]. Primäre Geschäftsprozesse, auch Hauptgeschäftsprozesse genannt, sind der Innovations-, Produktplanungs-, Produktentwicklungs-, Vertriebs-, Auftragsabwicklungs- und Serviceprozess. Sekundäre Geschäftsprozesse wirken unterstützend und umfassen die Aktivitäten zur Führung einer Organisation.

Bei den primären Geschäftsprozessen kann beispielsweise die Gestaltung einer Organisationsstruktur auf **Organisationsstrukturmustern** beruhen. Beispiele sind eine funktionale oder divisionale Struktur<sup>52</sup> [Bue04]. Darauf aufbauend lassen sich noch weitere Strukturmuster identifizieren. So entstehen z.B. unter zusätzlicher Betrachtung des Projektmanagements die Lösungsmuster *Projektmanagement in der Linie*, die *Einflussprojektorganisation*, die *Matrixorganisation* und die *reine Projektorganisation*<sup>53</sup> [Sch12].

Eine speziell auf die Produktentstehung mechatronischer Systeme angepasste Strukturierung der übrigen Geschäftsprozesse von SCHMELZER ET AL. ist das 3-Zyklen-Modell nach GAUSEMEIER (vgl. Kapitel 2.3.1). Die nachfolgende Gliederung dieses Kapitel orientiert sich an diesem Modell. Die vorgestellten Lösungsmuster fungieren dabei als Basis bei der Erstellung eines übergeordneten Wirkgefüges. Es handelt sich um einen beispielhaften Auszug, wohlwissend, dass es in den unterschiedlichen Bereichen der PE eine Vielzahl weiterer Artefakte und somit auch Lösungsmuster gibt. Es besteht daher über die gesamte Produktentstehung *kein Anspruch auf Vollständigkeit*. Ferner wird der Teil des Produktlebenszyklus im Anschluss an den Start der Produktion nicht betrachtet.

#### 4.2.2.1 Lösungsmuster in der Strategischen Produktplanung

Eine Übersicht über identifizierte Artefakte der strategischen Produktplanung nach GAUSEMEIER und zugehörige Lösungsmuster zeigt Bild 4-2. Dies sind in erster Linie Erfolgspotentiale, Produkt- und Technologiestrategien sowie Geschäftsmodelle.

##### **Erfolgspotential**

Bei der Identifizierung von Erfolgspotential zukünftiger Produkte kann eine besondere Form von Lösungsmustern dienlich sein. Die Rede ist von **Verhaltensmustern**, speziell von Verhaltensmustern potentieller Kunden. Die Erforschung von Verhaltensmustern findet größtenteils in der Psychologie statt [Kas03]. Die Industrie hat zunehmend ein gesteigertes Interesse daran, die Verhaltensmuster der Kunden zu analysieren, um zu-

---

<sup>52</sup>S. Anhang A2.1

<sup>53</sup>S. Anhang A2.1

künftige Trends oder Produktverbesserungen ableiten zu können. Mit modernen IT-Technologien wie bibliometrischen Analysen oder semantischen Technologien lässt sich das Verhalten der Kunden z.B. durch Foreneinträge im Internet umfangreich auswerten [GP14]. Die dadurch erhobenen Erkenntnisse fließen systematisch in die Planung neuer Produkte – Systeme oder Dienstleistungen – mit ein.

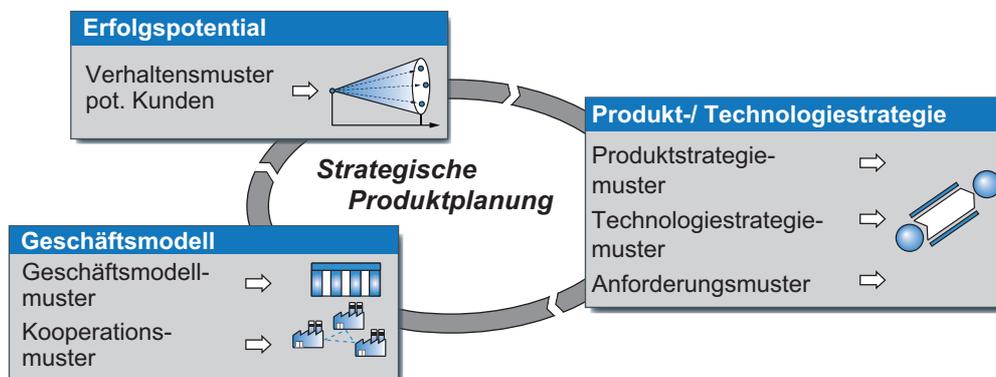


Bild 4-2: Lösungsmuster in der strategischen Produktplanung

## Produkt- und Technologiestrategie

Ein weiterer Schwerpunkt der strategischen Produktplanung ist die Entwicklung von Produkt- und Technologiestrategien. GAUSEMEIER zählt zu einer Produktstrategie in erster Linie Aussagen zur Produktprogrammgestaltung, zur Bewältigung der vom Markt geforderten Variantenvielfalt sowie zur Gestaltung des Entwicklungs- und Markteinführungsprozesses bis hin zur Produktprogrammpflege über den Produktlebenszyklus [GEK01], [GP14]. Es werden u.a. einzuhaltende Standards festgelegt, um produktübergreifend Skaleneffekte und somit Kostenvorteile erschließen zu können. Beispiele für **Produktstrategiemuster** sind Plattformstrategie, 100%-Engineering Sortimentsstrategie, Job #1-Engineering Sortimentsstrategie, Release-Engineering-Strategie, Add-on-Engineering Strategie, Non-variant-Engineering Strategie [Sch12].

Eng verknüpft mit der Produktstrategie ist die Technologiestrategie eines Unternehmens [Bri10]. Nach BUCHER ist „Eine Technologiestrategie ein Plan oder ein Muster, welches die Technologieziele, die Technologiepolitik und technologische Handlungsprogramme eines Unternehmens in ein geschlossenes Ganzes integriert“ [Buc03]. BULLINGER unterscheidet vier Strategien, die folglich als Beispiele für **Technologiestrategiemuster** interpretiert werden können: Pionierstrategie, Imitationsstrategie, Nischenstrategie und Kooperationsstrategie [Bul94].

Ein weiterer Typ von Lösungsmustern, der innerhalb der strategischen Produktplanung relevant ist, sind **Anforderungsmuster**. Der Ursprung von Anforderungsmustern liegt in der Informatik, speziell im Requirements Engineering. Einen Überblick über Anforderungsmuster und Zusammenhänge mit verwandten Hilfsmitteln wie z.B. Satzschablonen geben LEIMEISTER ET AL. [LJH+13]. Anforderungsmuster umfassen Angaben zu

Anforderungen, die wiederkehrend in Zusammenhang zueinander stehen und mit deren Hilfe die vollständige Spezifizierung von Anforderungen beschleunigt werden kann. Als verwandte Begriffe sind Anforderungsbündel oder auch Feature zu nennen. Im Rahmen der strategischen Produktplanung beinhalten die Anforderungsmuster, Anforderungen bzgl. des Marktes (resultieren oftmals aus Verhaltensmustern potentieller Kunden) oder auch des Gesetzes. Der Konkretisierungsgrad der Anforderungsspezifikation steigt mit der Dauer eines Entwicklungsprojekts [Hoo05]. Folglich verändern sich die inhaltlichen Schwerpunkte der Anforderungsmuster, wohingegen der grundsätzliche Aufbau und das Anwendungsziel identisch bleiben. Im Folgenden werden daher die Anforderungsmuster, wie z.B. Systemanforderungsmuster (Angaben auf Systemebene z.B. Entwicklung, Montage, Zulieferer etc.) lediglich referenziert und nicht mehr im Detail beschrieben.

### **Geschäftsmodell**

Nach GASSMAN ET AL. ist „*ein Geschäftsmodell ein aggregiertes Abbild der Geschäftslogik eines Unternehmens. Es beschreibt, wie ein Unternehmen Werte schafft, die seinen Kunden Nutzen stiften und dazu motivieren, dafür Geld zu zahlen*“ [GFC13]. Lösungsmuster und die methodische Entwicklung von neuen **Geschäftsmodellen** auf Basis dieser Muster sind bereits sehr etabliert. So präsentieren GASSMAN ET AL. insgesamt **55 Geschäftsmodellmuster**. Beispiele sind u.a. die Lösungsmuster *razor-and-blade*, *flatrate* oder auch *rent-instead-of-buy* [GFC13]. Geschäftsmodelle enthalten u.a. auch Angaben über Kooperationsformen. Auch in diesem Bereich lassen sich Beweggründe der Unternehmen für eine Kooperation in einem übergeordneten Wertschöpfungsnetzwerk als sog. **Kooperationsmuster** verstehen. Diese Muster liefern dem Management Informationen darüber, wann z.B. strategische Kooperationen im Bereich Forschung und Entwicklung sinnvoll sind, oder wann auf Lieferanten zurückgegriffen wird (System-/Modullieferant, Komponentenlieferant oder Teilelieferant) [Wag00].

#### 4.2.2.2 Lösungsmuster in der Produktentwicklung

Mit einer gewissen Überschneidung zur strategischen Produktplanung beginnt die Produktentwicklung. Die Entwicklung mechatronischer Systeme orientiert sich dabei nach wie vor am V-Modell der VDI 2206 [VDI2206]. Wie eingangs bereits geschildert, vollzieht der moderne Maschinenbau derzeit einen Wandel hin zu fortgeschrittenen mechatronischen Systemen wie beispielsweise einem CPS. Dies hat zur Folge, dass neben den klassischen Disziplinen der Mechatronik weitere z.T. nicht technische Disziplinen hinzukommen. Folglich steigt die Komplexität auf Gesamtsystemebene – sowohl bezogen auf das System als auch auf die Vernetzung und somit Kooperation zwischen den Disziplinen. Dieser Trend erfordert im Hinblick auf die VDI 2206 ein Umdenken. In der VDI 2206 bildet die sog. Prinziplösung das Ergebnis des fachdisziplinübergreifenden Systementwurfs. Diese wird in die einzelnen Disziplinen übergeben, wo die weitere

Konkretisierung stattfindet. Abstimmungen zwischen den Disziplinen sorgen für die fortlaufende Konsistenzsicherung.

In Anbetracht der Arbeiten des Systems Engineering und den Herausforderungen in der Entwicklung von CPS wird nur die Übergabe eines gewissen Entwurfsstands an die Disziplinen nicht mehr zielführend sein. Es wird vielmehr ein Systemmodell benötigt, das über die gesamte Entwicklung mitgeführt wird. Die charakteristischen Abläufe und Tätigkeiten in der Entwicklung von fortgeschrittenen mechatronischen Systemen sind schematisch in Bild 4-3 visualisiert. Die dargestellten Aktionen verteilen sich über insgesamt drei Ebenen mit zunehmender Konkretisierung von links nach rechts. Bei den Ebenen handelt es sich um die „Disziplinübergreifende System-Ebene“, die „Disziplinorientierte Mechatronik-Ebene“ und die „Disziplinintegrierende Aspektspezifische-Ebene“.

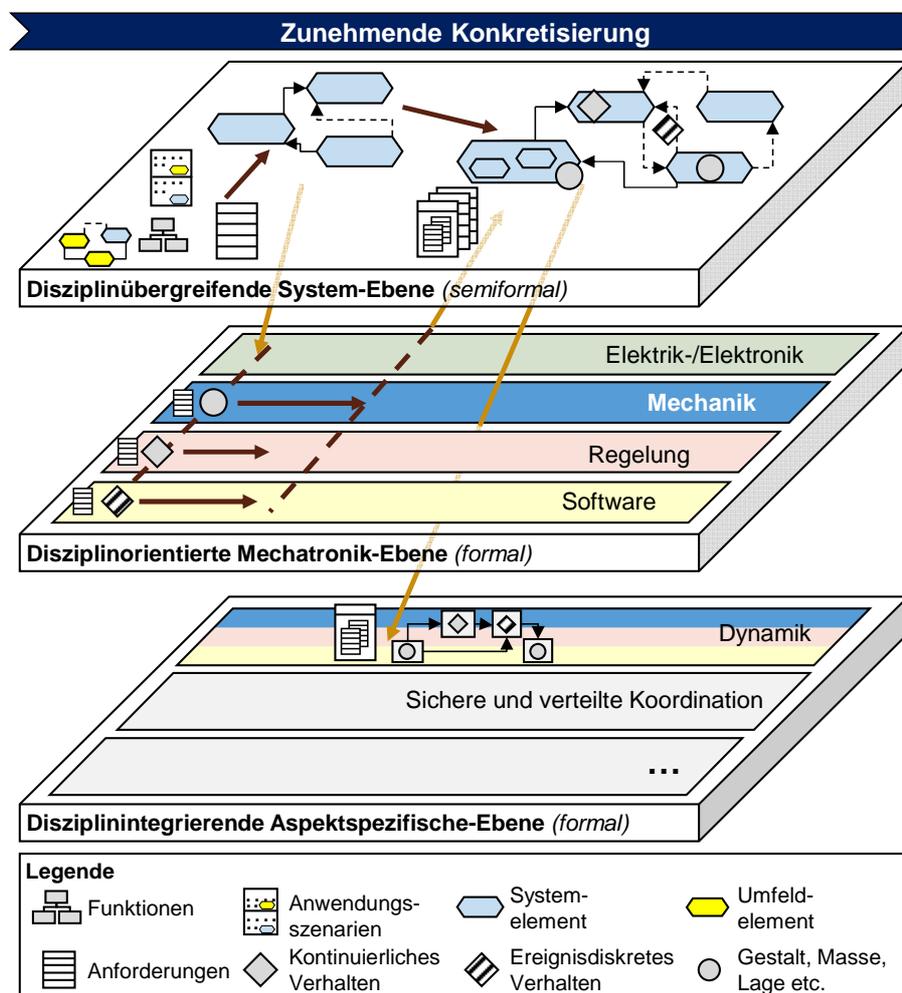


Bild 4-3: Ablauf von Tätigkeiten in der Entwicklung fortgeschrittener mechatronischer Systeme (Auszug)

Die *disziplinübergreifende System-Ebene* beinhaltet das Systemmodell. Es handelt sich um eine semiformale Abbildung des Gesamtsystems. Diese Ebene hat den Anspruch,

alle fachdisziplinübergreifend relevanten Informationen abzubilden. Es ist als eine Art Koordinationsschicht zu verstehen, über die alle Aktivitäten in der Entwicklung gesteuert werden.

Die Modelle auf der fachdisziplinübergreifenden Ebene werden bis zu jenem Zeitpunkt konkretisiert, ab dem disziplinspezifische Modelle hinsichtlich eines zu analysierenden Aspekts erforderlich werden. Zu analysierende *Aspekte* bei der Entwicklung von CPS sind beispielsweise die *Dynamik eines Mehrkörpersystems* oder die *sichere und verteilte Kommunikation oder Koordination*. Es handelt sich um Synthese- und Analyseschritte, die formale Modelle erfordern und größtenteils *fachdisziplinintegrierend* sind. Dies bedeutet, dass fachdisziplinspezifische Artefakte im Zusammenspiel betrachtet werden.

Folglich existiert eine Zwischenschicht, die sog. *Disziplinorientierte Mechatronik-Ebene*. Diese Ebene repräsentiert die zuvor erwähnten Artefakte, die losgelöst von den zu analysierenden Aspekten erarbeitet und betrachtet werden müssen. So besitzt jedes mechatronische System beispielsweise eine Regelung oder mechanische Komponenten.

Das Zusammenspiel der Entwurfsschritte in den einzelnen Ebenen kann unterschiedlich ausfallen. Dennoch sind alle Ebenen essentiell. Dies verdeutlichen die beiden folgenden Szenarien von Entwicklungstätigkeiten über die drei Ebenen hinweg. Das dynamische Verhalten dient dabei als zu analysierender Aspekt. Es wird davon ausgegangen, dass das System in der Entwicklung hinsichtlich weiterer Aspekte analysiert wird.

*Szenario 1:* Die abgebildeten Piktogramme (in diesem Fall bezogen auf die Spezifikationstechnik CONSENS) auf der disziplinübergreifenden Ebene symbolisieren die relevanten Aspekte beim Systementwurf. Aufbauend auf den Anforderungen und einer funktionalen Beschreibung des Systems folgt die Modellierung von Struktur- und Verhaltensmodellen (Bild 4-3, 1. dunkelroter Pfeil). Ab einem bestimmten Zeitpunkt sind disziplinspezifische Erkenntnisse für die weitere Konkretisierung erforderlich. Es erfolgt eine Überführung der Informationen z.B. in die Fachdisziplinen Regelungstechnik, Maschinenbau und Softwaretechnik (Bild 4-3, orangener Pfeil-links). Über einen gewissen Zeitraum wird beispielsweise die Gestalt oder eine Regelung formal in entsprechenden Softwarewerkzeugen erarbeitet. Hierbei kommt es in der Regel zu detaillierten Anforderungen, die z.T. Auswirkungen auf die anderen beteiligten Disziplinen haben können (Bild 4-3, rote parallele Pfeile). Anschließend folgt eine Synchronisation der Entwicklungsergebnisse auf der disziplinübergreifenden Ebene (Bild 4-3, orangener Pfeil-Mitte). Die Modelle werden entsprechend konkretisiert. Ferner werden Verweise auf die disziplinspezifischen Informationen vorgenommen. Von der fachdisziplinübergreifenden Ebene aus wird eine initiale Untersuchung der Dynamik auf der disziplinintegrierenden Ebene angestoßen (Bild 4-3, orangener Pfeil-rechts). Hierzu werden die formalen Modelle der Zwischenschicht zusammen mit den Schnittstelleninformationen der fachdisziplinübergreifenden Schicht an die disziplinintegrierende Ebene übermittelt. Die Informationen werden zusammengeführt, ggf. detailliert und anschließend simuliert und

analysiert. Die Ergebnisse werden zur weiteren Koordination des Entwicklungsgeschehens an die übergreifende Ebene übergeben. Dieses Zusammenspiel und die Absicherung der unterschiedlichen Eigenschaften geschehen fortlaufend unter Berücksichtigung aller relevanten Aspekte der unteren Ebene bis zum fertig entwickelten Produkt.

*Szenario 2:* Dieses Szenario unterscheidet sich von Szenario 1 in der erstmaligen Erstellung formaler Modelle (Bild 4-3, orangener Pfeil-links). Im Hinblick auf z.B. die frühestmögliche Analyse der Dynamik wird nicht als Erstes ein disziplinentorientierter Entwurf angestoßen. Stattdessen findet beispielsweise ein integrierter Software- und Regelungsentwurf auf der aspektspezifischen Ebene statt [GTS14]. In diesem Fall werden kontinuierliche Verhaltensanteile der Regelungstechnik mit ereignisdiskreten Verhaltensanteilen der Softwaretechnik im direkten Zusammenspiel entworfen, simuliert und analysiert. Die mittlere Ebene wird dabei gedanklich ausgelassen. Trotzdem werden die disziplinspezifischen Artefakte auf dieser Ebene indirekt erstellt und sind folglich (als eine Art Speicher für mechatronische Artefakte) mitzuführen. Dies ist notwendig, um der steigenden Anzahl zu analysierender Aspekte eines CPS adäquat zu begegnen. Dies verdeutlicht die Initiierung einer Analyse der verteilten Koordination im Anschluss an die der Dynamik. Hierzu sind z.B. von denen bis dato erstellten formalen Modellen nur die über die ereignisdiskrete Software relevant. Diese sind im gewählten Beispiel um spezifische Informationen bzgl. Datenverwaltung oder Dynamik in Netzwerken durch Experten weiterer Disziplinen zu ergänzen. Bei der Initiierung dieser disziplinintegrierenden Entwurfsschritte wird ausgehend von der obersten Ebene auf die separierten Informationen der Mechatronik-Ebene zurückgegriffen. Andernfalls würde sich eine nur sehr schwer beherrschbare Komplexität in der Vernetzung zwischen den CPS-spezifischen Aspekten auf der untersten Ebene ergeben.

Die fortlaufende Berücksichtigung der einzelnen Mechatronik-Artefakte ist daher ein entscheidender Stellhebel zur Bewältigung der steigenden Anzahl CPS-spezifischer Aspekte und dessen Zusammenwirken. Grundvoraussetzung dieser Vorgehensweise ist die abstrakte Darstellung aller fachdisziplinübergreifenden Informationen durch semi-formale Modelle auf einer Art Koordinationsebene. Auf den geschilderten Zusammenhängen fußt die nachfolgende Erläuterung der an der Produktentwicklung beteiligten Lösungsmuster. Einen Überblick über die identifizierten Lösungsmuster zeigt Bild 4-4.

## **Systemmodell**

Wie eingangs beschrieben, ist das zentrale Element in der Produktentwicklung das sog. Systemmodell. Eng mit dem Systemmodell verknüpft sind Anforderungen auf Systemebene. Der erste Schritt bei der Erstellung eines Systemmodells und somit der Konzipierung des Gesamtsystems ist die Anforderungsspezifikation. Hierbei werden, wie auch bereits im Rahmen der strategischen Produktplanung, **Anforderungsmuster** genutzt. Ebenfalls in Absprache mit der strategischen Produktplanung folgt die Erarbeitung einer Systemarchitektur. Bei der Gestaltung der Architektur muss unterschiedlichen Anforderungsmustern Rechnung getragen werden.

rungen Rechnung getragen werden – Kundenanforderungen, gesetzlichen Anforderungen, Anforderungen aus der Entwicklung, der Fertigung, Montage etc. **Systemarchitekturmuster** für mechatronische Systeme haben ihren Ursprung im Maschinenbau. Zu nennen sind z.B. die Integral- oder Differentialbauweise nach PAHL/BEITZ, sowie die Unterteilung nach SCHUH in Baukästen, Module, Baureihen und Pakete [PBF+07], [Sch12].

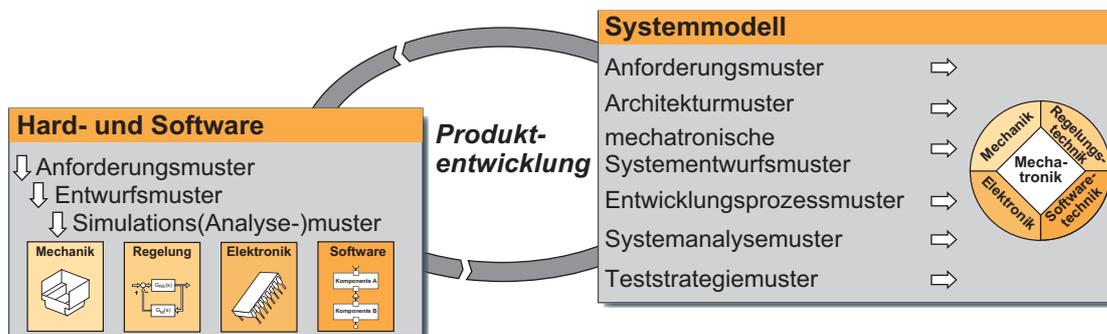


Bild 4-4: Lösungsmuster in der Produktentwicklung

Auf Basis der Architektur folgt der Systementwurf. Bei der Suche nach erfolgversprechenden Lösungsmustern hat sich die Funktionsbeschreibung als zentraler Meilenstein etabliert. Die Gliederung der gewünschten Funktionalität erfolgt in der Regel in einer Funktionshierarchie. Die Gesamtfunktion wird solange in Teilfunktionen unterteilt, bis sich für die Funktionen sinnvolle Lösungen finden lassen [VDI2221]. Dieser Vorgehensweise wurde in der Vergangenheit stark im Maschinenbau gefolgt, was die etablierten in der Regel fachdisziplinspezifischen **Entwurfsmuster**<sup>54</sup> wie Wirkprinzipien oder Funktions- und Prinzipfeature<sup>55</sup> etc. verdeutlichen [PBF+07], [Rot01], [KK98]. Aus der Analyse des Stands der Technik geht hervor, dass insbesondere im Systems Engineering (vgl. CLOUTIER, Kapitel 3.1.2) eine enge Verknüpfung zwischen Entwurfsmustern und sog. **Entwicklungsprozessmustern** gesehen wird. Es zeigt sich, dass z.B. generische Ablauffolgen zum Lösen technischer Probleme, wie „*Functions, Requirements, Architecture, Tests*“ als sog. FRAT-Muster interpretiert und dokumentiert sind [SS06]. Dieser Interpretationsweise folgend, lassen sich auch Referenzprozesse zur Entwicklung mechatronischer Systeme sowohl auf Makro- (Systementwurf, fachdisziplinspezifischer Entwurf und Ausarbeitung, Systemintegration) als auch auf Mikro-Ebene (Zielbestimmung, Synthese, Analyse) als Prozessmuster interpretieren [VDI2206]. Weitere Beispiele für feingranulare wiederverwendbare Prozessbausteine liefern beispielsweise KAHL, BICHLMAIER, MURR [Kah13] [Bic00], [Mur99]. Bei der Planung von Entwick-

<sup>54</sup>Der Schwerpunkt dieser Dissertation liegt auf **z.T. mechatronischen Systementwurfsmustern** für fortgeschrittene mechatronische Systeme. Diese sind im Hinblick auf die gestellten Anforderungen an dieses Instrumentarium zu entwickeln. Eine detaillierte Erläuterung findet daher in den nachfolgenden Kapiteln, insb. Kapitel 4.3, statt.

<sup>55</sup>S. Anhang A2.2

lungsprozessen erhält die Rollen- und Aufgabenverteilung eine zunehmend größere Bedeutung. Diese werden in einigen Ansätzen aus dem Stand der Technik ebenfalls als Muster interpretiert und in direkten Zusammenhang zu Entwicklungsprozessmustern gesetzt<sup>56</sup> [CD06].

In Bezug zu den in Bild 4-3 schematisch dargestellten Ebenen in der Entwicklung fortgeschrittener mechatronischer Systeme, lassen sich auch auf der fachdisziplinübergreifenden Ebene Analysen durchführen und somit **Systemanalysemuster** ableiten. Diese Muster enthalten z.B. Angaben darüber, welche Analysen zur Absicherung bestimmter Eigenschaften eingesetzt werden können. So existiert z.B. bereits eine Wissensbasis, die Erfahrungswissen zur frühzeitigen Analyse der Zuverlässigkeit mechatronischer Systeme bereitstellt [Gau10]. Ein weiteres Handlungsfeld des Systemmodells ist die Abstimmung mit der Produktionssystementwicklung im Hinblick auf eine Teststrategie. Je nach Komplexität und Architektur des Systems sind unterschiedliche **Teststrategiemuster** erforderlich. Diese beinhalten Angaben über Zusammenhänge, welche Eigenschaften und Funktionalitäten des Systems beispielsweise mit einem Komponententest, Modultest oder Systemtest getestet werden. Dieses Zusammenspiel mit der Produktionssystementwicklung, speziell der Montage, muss im Systemmodell frühzeitig berücksichtigt und konzipiert werden.

### Hard- und Software

Wie bereits beschrieben und in Bild 4-3 visualisiert, werden zu einem bestimmten Zeitpunkt in der Entwicklung erarbeitete Informationen im Systemmodell in den fachdisziplinspezifischen Entwurf der Hard- und Software übergeben. In einem ersten Schritt werden in der Regel detaillierte disziplinspezifische Anforderungen definiert, die auf der fachdisziplinübergreifenden Ebene keine Relevanz haben (z.B. Anforderungen an echtzeitkritisches Verhalten). In diesem Zusammenhang lassen sich erneut **Anforderungsmuster** spezifizieren. Diese, sowie die im Folgenden erläuterten Lösungsmuster, betreffen alle Disziplinen gleichermaßen (vgl. Bild 4-4).

Aufbauend auf den Anforderungsmustern werden mit Hilfe von **Entwurfsmustern** die erforderlichen Artefakte in den einzelnen Disziplinen synthetisiert. Diesbezüglich ist in Industrie und Forschung eine Vielzahl von Ansätzen vorhanden, um den in der Regel rechnerinternen Prozess zu beschleunigen (Koordinationsmuster der Software [DHT12] oder Gestaltungsfeature in CAD-Systemen). Damit einher geht auch die Anwendung von **Simulations- bzw. Analysemustern**. Dies verdeutlicht das nachfolgende Beispiel aus dem Maschinenbau.

Im *Maschinenbau* werden beispielsweise nach Festlegung der groben Gestalt auf der fachdisziplinübergreifenden Ebene, erste Gestaltmodelle am CAD-System erarbeitet.

---

<sup>56</sup>Einen ausführlichen Überblick über Rollen und Aufgaben im Hinblick auf die Erstellung eines fachdisziplinübergreifenden Systemmodells liefert KAISER [Kai13, S. 33ff.].

Unter Berücksichtigung der Restriktionen im Systemmodell werden Anforderungen an das Material, die Geometrie und z.B. die Belastung (Belastungsformen) spezifiziert. Bei dem anschließenden Entwurf werden beispielsweise Phasen, Nuten oder auch Bohrungen am Bauteil platziert. In diesem Zusammenhang existieren gestalterische Richtlinien, die auf langjähriger Expertise beruhen und als **Entwurfsmuster** interpretiert werden können. Diese enthalten z.B. Angaben über die Anzahl, Durchmesser und Abstände von Bohrungen je nach zugrundeliegendem Belastungsprofil. In modernen CAD-Systemen sind diese Zusammenhänge bereits hinterlegt und unterstützen den Entwickler beim Entwurf. Unmittelbar im Anschluss wird auf Basis der formalen Modelle rechnerintern eine Simulation und somit eine Analyse der erarbeiteten Ergebnisse durchgeführt. Hinsichtlich der Festigkeit lassen sich Bauteile beispielsweise mit Hilfe der Finite-Element-Methode<sup>57</sup> und zugehörigen Softwarewerkzeugen bewerten. Bei dieser Methode wird das zu untersuchende Bauteil in eine endliche Anzahl nicht unendlich kleiner Elemente unterteilt. Die auf diese Weise entstehende Gitterstruktur kann über den Verlauf eines Bauteils variieren. Je nach z.B. Form, Material und geforderter Güte der Analyse, lässt sich die Granularität der aufzubauenden Gitterstrukturen bestimmen. Wissen zur Beschleunigung dieser Analysen ist z.T. bereits Softwarewerkzeugen implementiert und lässt sich folglich als eine Art **Simulations- oder Analysemuster** interpretieren.

Aufgrund des anhaltenden Trends zu intelligenten technischen Systemen zufolge wird es zukünftig vermutlich vermehrt zu einem fachdisziplinintegrierten Entwurf und einer Analyse hinsichtlich unterschiedlicher Aspekte wie beispielsweise der Dynamik kommen. Damit einher geht das Potential, weitere Lösungsmuster zu identifizieren.

#### 4.2.2.3 Lösungsmuster in der Produktionssystementwicklung

Die Produktionssystemkonzipierung, als erster Schritt der Produktionssystementwicklung, beginnt bereits im Wechselspiel mit der Produktkonzipierung. Hierbei werden alle für das Produktionssystem relevanten Aspekte fachdisziplinübergreifend in einem Produktionssystemmodell abgebildet [GLL12]. Auf Basis dieses Modells erfolgt anschließend die Erstellung eines Arbeitsplans bestehend aus einem Arbeitsablauf-, Arbeitsstätten-, Materialfluss- und Arbeitsmittelplan.

#### **Produktionssystemmodell**

Notwendige Aspekte des Produktionssystemmodells sind u.a. Anforderungen, Gestalt des Produktionssystems, Prozesse oder auch Ressourcen. Anforderungen an das Produktionssystem werden maßgeblich vom Produktkonzept determiniert. Dieser Kausalkette folgend lassen sich **Anforderungsmuster** beschreiben (vgl. Bild 4-5). Diese bein-

---

<sup>57</sup>Die gewählte Methode dient der Erklärung von Simulations- und Analysemustern und wird daher nicht in Gänze beschrieben. Für detaillierte Informationen sei an dieser Stelle auf einschlägige Literatur verwiesen. Vgl. z.B. [Kle10]

halten als zugrundeliegendes Problem charakteristische Eigenschaften des Produktkonzepts und als Lösungsbeschreibung zugehörige Anforderungsbündel an das zu konzipierende Produktionssystem. Charakteristische Eigenschaften könnten z.B. sein: verwendete Technologie (z.B. MID<sup>58</sup>) oder geplante Stückzahl.

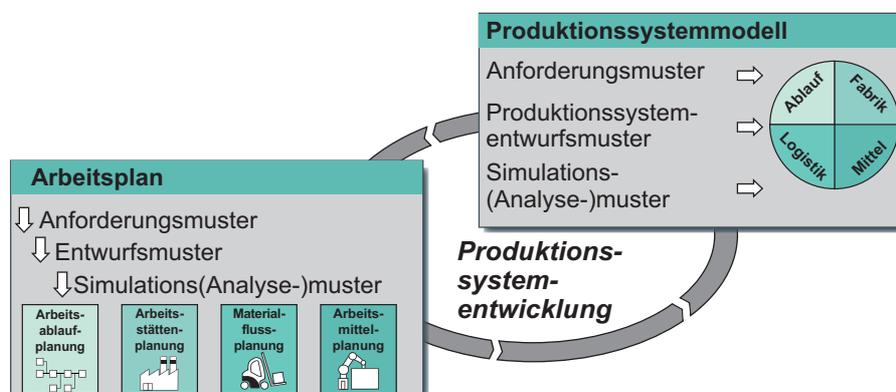


Bild 4-5: Lösungsmuster in der Produktionssystementwicklung

Auf Basis der spezifizierten Anforderungen folgt die Produktionssystemkonzipierung unter Zuhilfenahme von **Entwurfsmustern**. Als Eingangsinformationen dient dabei neben den Anforderungen der bis dato erarbeitete Stand des Produktkonzepts. Ist beispielsweise die Technologie zur Umsetzung des Produkts oder einer Teilkomponente ausgewählt, lassen sich das Produktionssystemmodell und die darin enthaltenen Aspekte sukzessive synthetisieren. Ein Beispiel für eine mögliche Strukturierung von Lösungsmustern für die Produktionssystemkonzipierung zeigt Bild 4-6. In Anlehnung an die Kategorien nach dem Architekturtheoretiker ALEXANDER unterteilen sich die Lösungsmuster in die Kategorien Name, Problem, Lösung und Kontext. Das zugrundeliegende Problem wird symbolisiert durch einen dreidimensionalen spritzgegossenen Schaltungsträger. Als Kern der Lösung lassen sich beispielsweise unterschiedliche Prozessketten zur Herstellung eines solchen Schaltungsträgers aufführen [Fra13]. In Bild 4-6 werden diese verdeutlicht durch die Partialmodelle Ressourcen und Prozesse. Zusätzlich ist es sinnvoll, spezifische Merkmale, Expertise über einzelne Feature – als eine Art spezialisiertes Muster<sup>59</sup> – oder den Kontext, zu beschreiben. Unmittelbar im Zusammenspiel mit den Entwurfsmustern lässt sich mit Hilfe von **Simulationen** das synthetisierte Produktionssystemkonzept untersuchen. Diese Zusammenhänge verdeutlicht das skizzierte Anwendungsbeispiel der Herstellung eines 3D-Schaltungsträgers im Anhang A2.3. **Analysemuster** in Bezug auf das Produktionssystemmodell existieren beispielsweise bzgl. Herstellungskosten [GLL12].

<sup>58</sup>Vgl. [Fra13]

<sup>59</sup>Ein Beispiel für ein spezialisiertes Muster der Technologie MID befindet sich im Anhang A2.3.

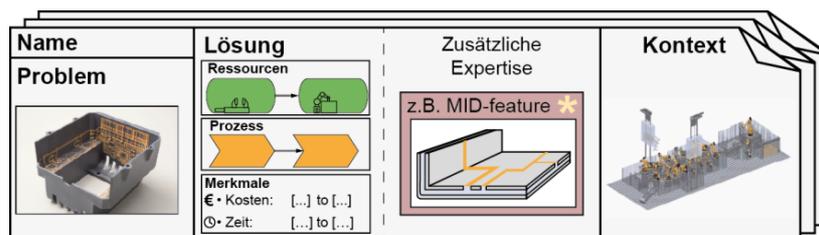


Bild 4-6: Strukturierung für Entwurfsmuster für die Produktionssystemkonzipierung

Analog zu den bereits beschriebenen Artefakten in der Produktentstehung lassen sich, auch im Hinblick auf die Teilbereiche des Arbeitsplans, **Anforderungsmuster** identifizieren. Die darauf aufbauende Planung und Umsetzung der zugehörigen notwendigen Artefakte wie ein Arbeitsmittelplan oder ein Arbeitsstättenplan beruhen auf langjähriger Expertise. Diese ist umfangreich dokumentiert und steht der in Kapitel 4.2.1 formulierten Definition zur Folge, als Lösungsmuster bzw. Entwurfsmuster zur Verfügung. Als Beispiele sind u.a. die Zusammenhänge hinsichtlich erforderlicher Werkzeuge zum Umformen, Urformen etc. aus der VDI3320 zu nennen [VDI3320-2]. Als Entwurfsmuster für den Aufbau von Arbeitsstätten existieren heutzutage zahlreiche Normungen zur Arbeitsplatzgestaltung in jeglicher Form. Diese enthalten u.a. Informationen zu den Bereichen Ergonomie oder Anordnung. Im Bereich der Materialflussplanung lassen sich z.B. die Entwurfsmuster „Just-in-Time“ oder „Just-in-Sequence“ aufführen.

Zur Validierung bestimmter Zusammenhänge werden auch im Bereich der konkreten Arbeitsplangestaltung **Simulations- bzw. Analysemuster** eingesetzt. So existieren beispielsweise Arbeiten für die simulative Unterstützung der Arbeitsplanung und Arbeitssteuerung durch eine Cloud-Anwendung. Ein virtuelles Abbild einer Werkzeugmaschine soll ein automatisiertes Einrichten der Maschinen und eine effiziente Auftragsdisposition ermöglichen. Die dafür erforderliche Expertise der Fertigungsplaner wird rechnerintern in Form von Mustern in die Cloud-Anwendung integriert [BGP+14].

#### 4.2.3 Wirkgefüge der Lösungsmuster in der Produktentstehung

Wie die vorangegangenen Kapitel anhand ausgewählter Beispiele verdeutlichen, lassen sich Lösungsmuster in allen Bereichen der PE zur Unterstützung der wiederkehrenden Tätigkeiten – Zielbestimmung, Synthese, Analyse – identifizieren. Ferner existieren Zusammenhänge zwischen den einzelnen Lösungsmustern. Dies betrifft die Lösungsmuster innerhalb eines spezifischen Bereichs der Produktentstehung, wie z.B. der Produktentwicklung (s. Kapitel 4.3.3.4), als auch bereichsübergreifend.

Eine Übersicht der im Vorfeld beschriebenen Lösungsmuster und ihre Relationen über die PE hinweg zeigt Bild 4-7. Die Strukturierung orientiert sich dabei an den wesentlichen Handlungsfeldern in der Produktentstehung: *Strategische Produktplanung*, *Produktentwicklung*, *Produktionssystementwicklung*. Dieser Struktur folgend, lassen sich

auf einer übergeordneten Ebene grundsätzliche Zusammenhänge herstellen. Diese werden nachfolgend auszugsweise erläutert.

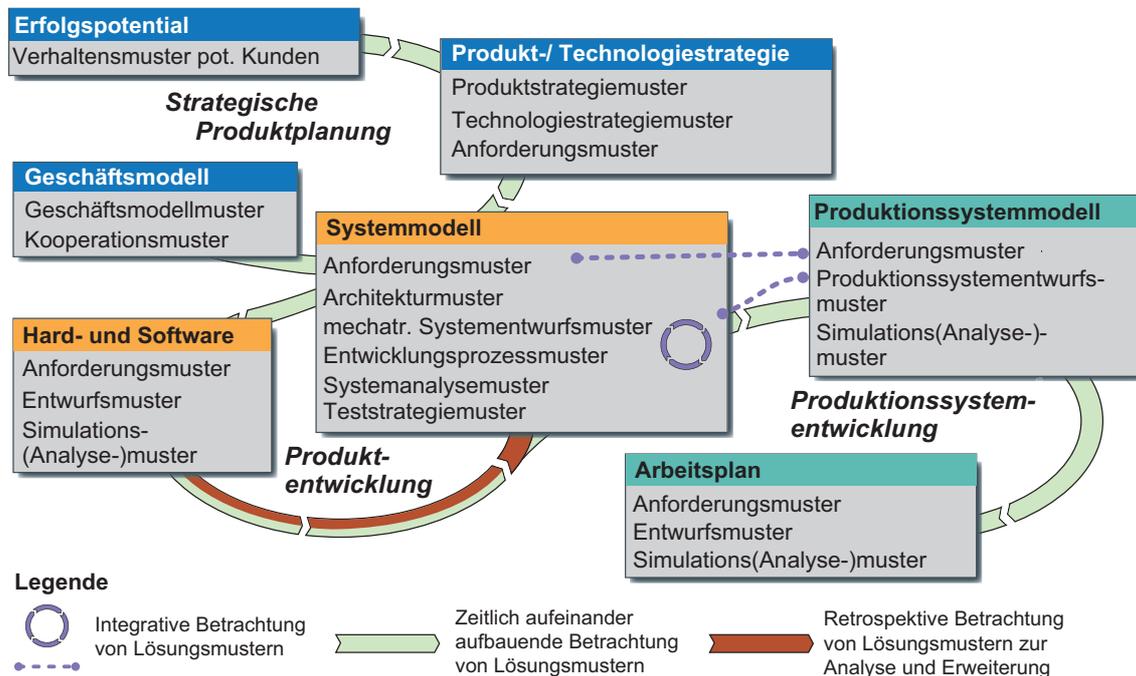


Bild 4-7: Wirkgefüge von Lösungsmustern in der Produktentstehung

### Integrative Betrachtung von Lösungsmustern

Die integrative Betrachtung von Lösungsmustern ist genau dann erforderlich, wenn bestimmte bereichsübergreifende Aktivitäten integrativ ablaufen müssen. Zu erwähnen sind u.a. der integrative Entwurf von Produkt- und Produktionssystemkonzept. Daraus resultiert der direkte Zusammenhang zwischen den Systementwurfsmustern und den Entwurfsmustern für das Produktionssystemmodell. Ein weiteres Beispiel lässt sich aus den Forderungen an die Weiterentwicklung des Systems Engineering für die Entwicklung intelligenter technischer Systeme ableiten. Gemeint ist die integrative Betrachtung von Produkt und Projekt. Hieraus resultiert die bilaterale Verbindung von Systementwurfsmustern und Entwicklungsprozessmustern als ein wesentlicher Baustein für die Projektplanung.

### Zeitlich aufeinander aufbauende Betrachtung von Lösungsmustern

Die Problemanalyse hat verdeutlicht, dass aus einer beliebigen Menge von Lösungsmustern eine sog. *Mustersprache* wird, wenn die Anwendung eines Musters zu weiteren Mustern führt, die *anschließend* betrachtet werden sollten [BC87]. Diese Beziehungsform zieht sich in Bild 4-7 konsequent durch die gesamte Produktentstehung. Als Beispiel sei der Zusammenhang zwischen den *Produktstrategiemustern* und den *Systementwurfsmustern* zu nennen. Die Festlegung auf ein Produktstrategiemuster oder auf eine Kombination von Produktstrategiemustern (Beispiel modularer Querbaukasten in

der Automobilindustrie: Plattformstrategie, Modulstrategie und Baukastenstrategie) determinieren die Auswahl und somit die für die Entwicklung zur Verfügung stehenden Lösungsmuster signifikant. Dies ist insbesondere der Fall, wenn ein Neuprodukt gemäß der festgelegten Strategie entwickelt werden muss.

### **Retrospektive Betrachtung von Lösungsmustern zur Analyse und Erweiterung**

Eine weitere Relation ist die retrospektive Betrachtung von Lösungsmustern. Im Nachgang einer erfolgreichen Produkt- und Produktionssystementwicklung ist zu analysieren, aus welchen angewendeten detaillierten Lösungsmustern abstrakte Lösungsmuster für zeitlich vorgelagerte Entwicklungstätigkeiten identifiziert werden können. Als erklärendes Beispiel sind die im Rahmen der Produktentwicklung verwendeten Entwurfsmuster der Fachdisziplinen Maschinenbau, Regelungstechnik und Softwaretechnik zu nennen. Auf Basis einzelner Muster oder einer bestimmten Kombination dieser Muster lassen sich z.T. mechatronische Systementwurfsmuster identifizieren und definieren. Im Hinblick auf ihren Einsatzzweck unterscheiden sich die Lösungsmuster und die darin enthaltenen Informationen beispielsweise in ihrer Detailtiefe<sup>60</sup>. In Ergänzung enthalten sie relevante Informationen über die Schnittstellen zwischen den ehemals disziplinspezifischen Lösungsansätzen.

### **Weitere grundsätzliche Zusammenhänge und Besonderheiten**

Es ist darauf hinzuweisen, dass neben den Beziehungen über die gesamte Produktentstehung weitere Wechselwirkungen und Besonderheiten bestehen. Ein sehr allgemeiner Zusammenhang bezieht sich dabei auf Lösungsmuster zur Erstellung von **Prozessen und Objekten**. Ferner existieren **Wechselwirkungen zwischen Lösungsmustern in der Entwicklung einzelner Artefakte** (z.B. Systemmodell). Darüber hinaus gibt es Besonderheiten in der **Verträglichkeit von Lösungsmustern gleichen Typs**. Abschließend wird ersichtlich, dass **wiederholte Nennungen von Lösungsmustern gleichen Typs** vorkommen, diese jedoch deutliche Unterschiede in ihrem Detaillierungsgrad und der Repräsentationsart der bereitgestellten Informationen aufweisen.

- *Prozess- und Objektmuster*: Bei der Erstellung von Prozessabläufen in unterschiedlichen Bereichen der Produktentstehung können Lösungsmuster unterstützend eingesetzt werden. Dabei kann es zu Verweisen auf Lösungsmuster kommen, die der Synthese von Objekten dienen (vgl. Definition in Kapitel 4.2.1). Dies muss nicht zwangsläufig der Fall sein.
- *Wechselwirkungen zwischen Lösungsmustern in der Entwicklung einzelner Artefakte*: Analog zu der zeitlichen Reihenfolge in der Betrachtung von Lösungs-

---

<sup>60</sup>Die Art der Zusammenhänge zwischen Systementwurfsmustern und z.B. den Entwurfsmustern der Softwaretechnik oder der Regelungstechnik ist ein Schwerpunkt der vorliegenden Dissertation. Eine detaillierte Beschreibung folgt in Kapitel 4.3 speziell in Kapitel 4.3.3.

mustern auf höchster Ebene in der Produktentstehung, besteht ein zeitlicher Zusammenhang während der Entwicklung einzelner Artefakte. So determinieren beispielsweise ausgeprägte Anforderungsmuster die Auswahl von Systementwurfsmustern.

- *Verträglichkeit von Lösungsmustern gleichen Typs:* In der Regel kommt es im Rahmen einer Lösungsmusterbasierten Produktentstehung bei der Erstellung der einzelnen Artefakte zur Kombination von Lösungsmustern. Hierbei ist darauf zu achten, dass die einzelnen Lösungsmuster in Kombination *verträglich* sind. Die wechselseitige Beziehung von unverträglichen Lösungsmustern ist als sog. *Anti-Pattern-Beziehung* festzuhalten.
- *Wiederholte Nennung von Lösungsmustern gleichen Typs:* Im Wirkgefüge der Lösungsmuster in Bild 4-7 wird ersichtlich, dass sich z.B. Anforderungsmuster wiederholen. Hierbei gilt es zu beachten, dass zwar der Aufbau und die Zielstellung von Lösungsmustern gleich sein kann, sie sich jedoch signifikant im Konkretisierungsgrad des bereitgestellten Wissens unterscheiden. Anforderungsmuster in der strategischen Produktplanung beziehen sich in der Regel auf sehr allgemeine Themen wie gesetzliche Rahmenbedingungen und Marktanforderungen im Hinblick auf das Gesamtsystem. Im Gegensatz dazu beziehen sich z.B. Anforderungsmuster in der Produktentwicklung, etwa in der Softwaretechnik, auf detaillierte Anforderungen z.T. auf Subsysteme. Für das aufgeführte Beispiel der Software können dies z.B. Deadlines für Synchronisationsvorgänge einzelner Softwarekomponenten sein.

### 4.3 Lösungswissen für den Systementwurf

Die vorangegangenen Kapitel zeigen, dass für den Entwurf fortgeschrittener mechatronischer Systeme Lösungswissen von allen beteiligten Disziplinen und deren Experten notwendig ist. In erster Linie handelt es sich dabei um implizites Wissen. Als implizites Wissen wird im Rahmen dieser Arbeit jenes Wissen verstanden, das zunächst stillschweigend verfügbar ist. Es ist entweder personengebunden oder liegt unerkannt in weiteren Wissensquellen vor wie z.B. Entwicklungsdokumenten oder Computermodellen. Eine Besonderheit im Systementwurf liegt darin, dass das Lösungswissen aus den unterschiedlichen Disziplinen gleichwertig behandelt und miteinander kombiniert werden muss. Dieses Kapitel nimmt sich daher den Herausforderungen an, Lösungswissen zu externalisieren und eine geeignete Technik für den Umgang mit diesem Lösungswissen zu erarbeiten. Primäres Ziel ist es, Lösungswissen fachdisziplinübergreifend und allgemeinverständlich zu repräsentieren (Kapitel 4.3.1). Die dafür notwendige Strukturierung gilt es anschließend anhand ausgewählter Beispielmuster zu validieren (Kapitel 4.3.2). Durch die Kombination von ehemals i.d.R. fachdisziplinspezifisch geprägten Lösungsmustern entsteht neuartiges und für den Systementwurf charakteristisches Lösungswissen; Wissen über Schnittstellen und somit über wechselseitige (z.T. multidis-

ziplinäre) Beziehungen. Vor diesem Hintergrund wird ein multidimensionaler Wissensraum vorgestellt (Kap. 4.3.3). Er beschreibt die charakteristischen Dimensionen der Lösungsmuster für den Systementwurf und veranschaulicht die Zusammenhänge zur Produktentwicklung anhand des Zusammenspiels mit den in Kapitel 4.2.2.2 vorgestellten Ebenen.

### 4.3.1 Einheitliche Strukturierung von Lösungsmustern

Um der steigenden Interdisziplinarität im Rahmen des Systementwurfs gerecht zu werden, ist eine einheitliche Strukturierung von Lösungsmustern erforderlich. In Anlehnung an die Definition von ALEXANDER sind die Lösungsmuster in vier Kategorien gegliedert: Name, Problem, Lösung, Kontext [AIS+77]. Relevantes Lösungswissen der einzelnen Kategorien wird durch Aspekte der Spezifikationstechnik CONSENS präsentiert (vgl. Bild 4-8). Die Aspekte sowie deren Bedeutung und Zuordnung zu den Kategorien werden im Folgenden erläutert.

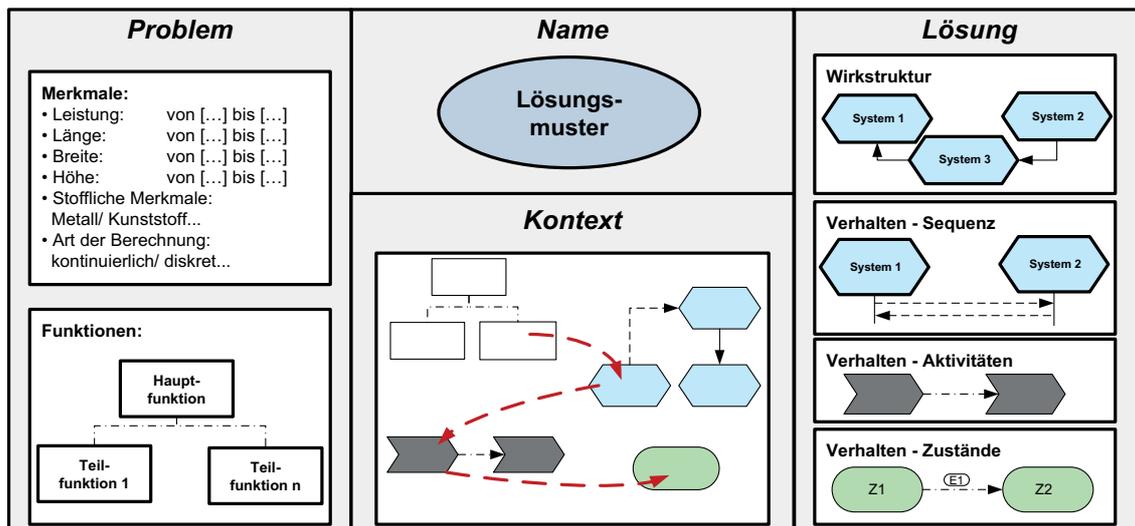


Bild 4-8: Einheitliche Strukturierung von Lösungsmustern für den Systementwurf

#### Name:

Jedes Muster ist eindeutig zu benennen. Der Name gibt dem Entwickler bei der Suche nach einem passenden Muster bereits erste Anhaltspunkte für den Inhalt, sofern sich der Entwickler in der entsprechenden Materie auskennt. Um jedoch die Kreativität des Entwicklers zu fördern, ist es empfehlenswert, die Problembeschreibung als Ausgangspunkt für die Suche zu nutzen.

#### Problem:

Die Aspekte Merkmale und Funktionen sind hinsichtlich der Definition nach ALEXANDER gleichbedeutend mit dem zu lösenden Problem. Beide Aspekte sollten so lösungsneutral wie möglich beschrieben werden. Sie bilden in Kombination mit dem Lösungsmusternamen die Auswahlkriterien im Zuge eines Systementwurfs.

**Merkmale:** Dieser Aspekt umfasst sämtliche Eigenschaften, die für das Lösungsmuster charakteristisch sind. In der Regel werden für die Eigenschaften Parameterbereiche in Intervallen angegeben. Bei einem Elektromotor sind dies z.B. Angaben über die Geometrie oder die Drehzahl der Antriebswelle. Auch Angaben zu stofflichen Merkmalen oder zur Bewegungsart (rotatorisch/translatorisch) werden hinterlegt. Lösungsmuster zur Spezifikation der Informationsverarbeitung besitzen ihrerseits wiederum charakteristische Merkmale. Beispiele sind Echtzeitfähigkeit (weiche/harte Echtzeit) oder Art der Berechnung (diskret/kontinuierlich). Die Ausarbeitung dieses Aspektes basiert auf den zuvor definierten Anforderungen, die im Zuge der Lösungsmusterdefinition generalisiert wurden.

**Funktionen:** Im einfachsten Fall erfüllt ein Lösungsmuster genau eine Funktion (vgl. Wirkprinzipien der Konstruktionslehre) [PBF+07]. Je nach Komplexität des Problems kann ein Lösungsmuster auch mehrere Funktionen erfüllen. Die Abbildung der Funktionen erfolgt mithilfe einer Funktionshierarchie.

### **Lösung:**

Die Beschreibung der Lösung setzt sich zusammen aus den Partialmodellen zur Abbildung des Verhaltens und der Wirkstruktur. Je nach zugrunde liegendem Problem können die genannten Partialmodelle auch einzeln genügen. So reicht die Modellierung der Wirkstruktur aus, wenn keine Softwareanteile vorhanden sind (z.B. bei einem Lösungsmuster über mechanische Zusammenhänge). Das Vorhandensein ausgefüllter Aspekte gibt somit bereits ersten Aufschluss über die Komplexität des Lösungsmusters (vgl. Kapitel 4.3.3)

*Wirkstruktur:* Dieser Aspekt bildet den Kern der Lösungsbeschreibung. Es werden die Systemelemente sowie deren Beziehungen untereinander abgebildet. Die Darstellung der prinzipiellen Wirkungsweise des Lösungsmusters verdeutlicht, wie die zuvor definierten Funktionen erfüllt werden. Die Beziehungen zwischen Lösungsmustern und Systemelementen der Wirkstruktur haben den Typ m:n.

*Verhalten:* Die Verhaltensmodelle komplettieren die Lösungsbeschreibung. Die Spezifikation des Verhaltens ist insbesondere für Lösungsmuster mit Schwerpunkt Softwaretechnik relevant. Je nach fokussiertem Verhalten werden die Informationen durch Aktivitäten, Zustände oder Sequenzdiagramme abgebildet. Mit Hilfe von Verhalten-Aktivitäten lässt sich spezifizieren, in welcher Reihenfolge ein Systemelement seine Funktion/Funktionen ausführt. Im Aspekt Verhalten-Zustände werden Zustände und Zustandübergänge modelliert, die sich in der Regel nach Ausführung einer Aktivität ändern. Hinsichtlich einer Systemsimulation ist die Modellierung von Zuständen auch für passive Systemelemente relevant. Es sind mindestens elementare Zustände wie defekt oder nicht defekt aufzugreifen. In dem Modell Verhalten-Sequenz werden Interaktionen zwischen zwei oder mehreren Systemelementen abgebildet. Der Schwerpunkt liegt auf der zeitlichen Abfolge, in der z.B. Nachrichten untereinander ausgetauscht werden. Die Abbildung von Sequenzen ist eine im Rahmen dieser Arbeit notwendige Erweiterung

der Spezifikationstechnik CONSENS, um die Anknüpfung an Lösungsmuster aus der Fachdisziplin Softwaretechnik zu gewährleisten. Die Erweiterung wird beispielhaft in Kapitel 4.3.2.1 vorgestellt.

**Kontext:**

Der Aspekt Kontext beinhaltet Anwendungen, in denen das Lösungsmuster bereits erfolgreich eingesetzt wurde. Die Spezifikation eines Lösungsmusters verfügt über mindestens ein Anwendungsbeispiel, in dem sowohl das verwendete als auch andere bekannte Muster gekennzeichnet werden. Ferner kann es je nach Lösungsmuster sinnvoll sein, eine aussagekräftige Skizze oder eine präzise textuelle Erklärung zu ergänzen.

### 4.3.2 Ausgewählte Beispiele von Lösungsmustern

In diesem Kapitel werden beispielhaft Lösungsmuster für den Systementwurf präsentiert. Die Auswahl der Beispiele orientiert sich dabei an wesentlichen Herausforderungen im Kontext fortgeschrittener mechatronischer Systeme. Vor diesem Hintergrund adressiert das erste Lösungsmuster die zunehmende Vernetzung der Systeme und die damit verbundene Kommunikation untereinander anhand des softwareintensiven Beispiels „Zusammenarbeit synchronisieren“. Das zweite Beispiel „Servoantrieb“ setzt sich aus regelungstechnischen und maschinenbaulichen Lösungsprinzipien zusammen. Es bezieht sich daher eher auf das physikalische Grundsystem eines mechatronischen Systems und repräsentiert fachdisziplinübergreifendes Lösungswissen der Disziplinen Maschinenbau und Regelungstechnik. Anhand der ausgewählten Beispiele werden ferner charakteristische Dimensionen erläutert, die im späteren Einsatz von Lösungsmustern von entscheidender Bedeutung sind. So werden z.B. die Aggregation von Lösungsmustern für den Systementwurf oder auch die Zusammenhänge in die nachfolgenden fachdisziplinspezifischen Entwurfsphasen geschildert.

#### 4.3.2.1 Softwareintensives Lösungsmuster „Zusammenarbeit synchronisieren“

Bild 4-9 zeigt den exemplarischen Ausschnitt ausgearbeiteter Aspekte eines Lösungsmusters am Beispiel „Zusammenarbeit synchronisieren“. Es sind die Partialmodelle Funktionen, Wirkstruktur, Verhalten-Zustände und Verhalten-Sequenz dargestellt. Das Lösungsmuster „Zusammenarbeit synchronisieren“ basiert auf speziellen Erkenntnissen aus der Softwaretechnik. Um diese Kenntnisse den Entwicklern bereits in der fachdisziplinübergreifenden Entwurfsphase zur Verfügung zu stellen, sind die Informationen zum einen mit der Spezifikationstechnik CONSENS abzubilden. Zum anderen hat das abgebildete Muster einen höheren Generalisierungsgrad und ist weitaus weniger formal

als die Darstellung des Koordinationsmusters „Synchronized Collaboration“<sup>61</sup> aus der Softwaretechnik. Um das Lösungsmuster im Rahmen der fachdisziplinübergreifenden Konzipierung verwenden zu können, sind alle notwendigen Informationen mit der Spezifikationstechnik CONSENS abgebildet. Sehr detaillierte softwarespezifische Informationen wie z.B. Echtzeitbedingungen werden nicht berücksichtigt. Dieser Detaillierungsgrad ist im Rahmen der frühzeitigen Konzipierung nicht praktikabel.

Das gewählte Beispielmuster wird angewendet, um Koordinationsvorgänge zwischen Softwarekomponenten, und somit Systemelementen, vorzunehmen. Es handelt sich um ereignisdiskrete Vorgänge. Diese können gleichermaßen auf Gesamtsystemebene oder auf Subsystemebene stattfinden. Die in Bild 4-9 abgebildeten Aspekte des Lösungsmusters „Zusammenarbeit synchronisieren“ werden im Folgenden erläutert.

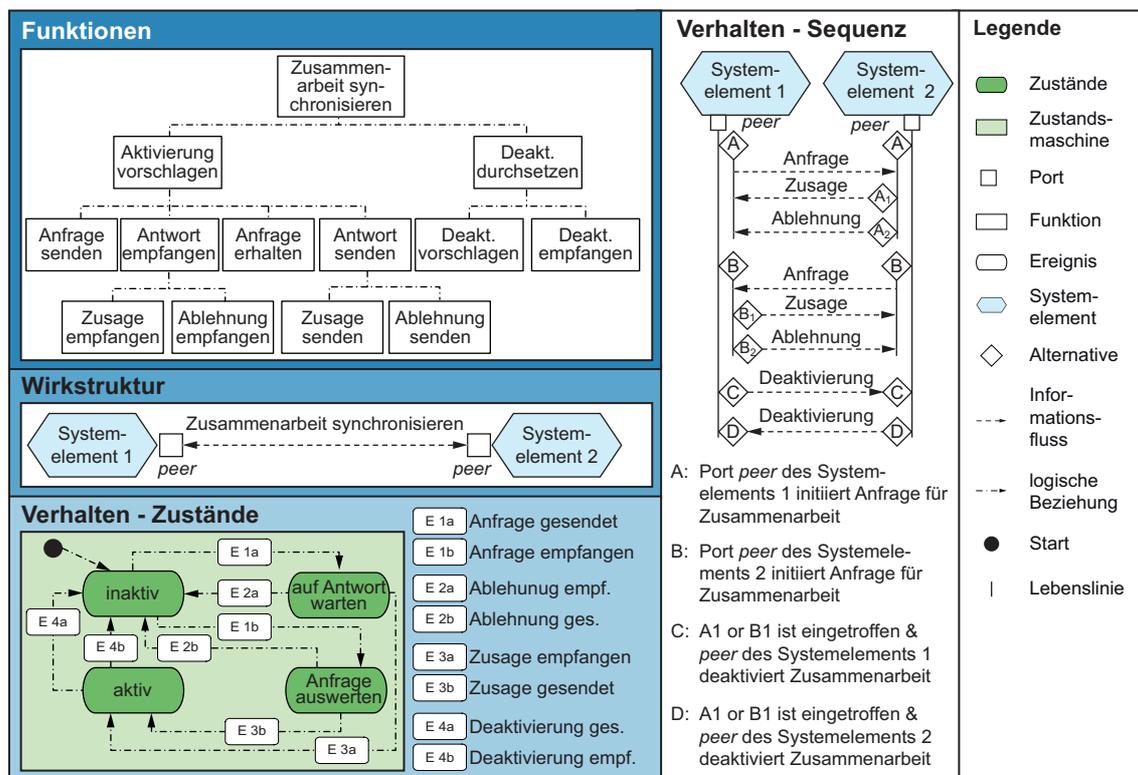


Bild 4-9: Aspekte des Lösungsmusters „Zusammenarbeit synchronisieren“

**Problem** dargestellt durch *Funktionen*: Hauptfunktion des gleichnamigen Lösungsmusters ist „Zusammenarbeit synchronisieren“. Diese unterteilt sich in die beiden Teilfunktionen „Aktivierung vorschlagen“ und „Deaktivierung durchsetzen“. Erstere unterteilt sich weiter in die Funktionen „Anfrage senden/empfangen“ und „Antwort senden/empfangen“. Die Funktion „Deaktivierung durchsetzen“ unterteilt sich in die Funktionen „Deaktivierung vorschlagen“ und „Deaktivierung empfangen“. Wie ein-

<sup>61</sup>Eine Darstellung des Koordinationsmusters „Echtzeit- Zusammenarbeit aushandeln“ befindet sich im Anhang A3.

gangs erwähnt liegt der Schwerpunkt dieses Lösungsmusters auf den Interaktionen zwischen den zwei Systemelementen. Die Berechnungsvorgänge innerhalb eines Systemelements, bzw. im weiteren Verlauf der Entwicklung innerhalb einer Softwarekomponente, sind nicht Gegenstand des Lösungsmusters.

**Lösung** dargestellt durch *Wirkstruktur, Verhalten-Sequenz, Zustände*: Die Spezifikation Wirkstruktur dient im Wesentlichen der Vollständigkeit. Lösungsmuster mit Schwerpunkt Softwaretechnik bestehen in der Regel aus zwei Systemelementen mit einem uni- bzw. bidirektionalen Informationsfluss als Verbindung. Die Kommunikation zwischen den Systemelementen läuft über sog. Ports, die unterschiedliche Berechtigungen bzw. Rollen haben können. In dem dargestellten Fall handelt es sich jeweils um die Rolle „peer“. Weitere Rollen wären z.B. „master“ oder „slave“. Die Bezeichnung der Ports lässt automatisch Rückschlüsse auf ihre Berechtigungen in der Interaktion zu. Ein Master-Port hat somit die Möglichkeit z.B. Anfragen zu initiieren und zu senden, wohingegen der Slave-Port allenfalls reagieren darf. In dem abgebildeten Lösungsmuster haben die Ports „peer“ dieselben Berechtigungen, was in dem Partialmodell Verhalten-Sequenz zum Ausdruck kommt.

Im Kern werden mit Hilfe des neu erarbeiteten Sequenzdiagramms und seiner Anwendung in der Spezifikationstechnik CONSENS Informationsflüsse im Detail spezifiziert. Bei der Integration der Lösungsmuster aus der Software ist es keinesfalls ausreichend, einen Informationsfluss bidirektional mit Beschriftung darzustellen. Die eigentliche Funktionsweise in der Aktion kann auf diese Weise nicht zum Ausdruck gebracht werden. Es ist erforderlich, die Reihenfolge von Nachrichtenvorgängen, die sich hinter einem bidirektionalen Informationsfluss verbergen können, detailliert zu spezifizieren. Um der Anforderung nach einer fachdisziplinübergreifenden Darstellung gerecht zu werden, nutzt das neuentwickelte Sequenzdiagramm vorhandene Konstrukte der Spezifikationstechnik CONSENS.

Das Partialmodell Verhalten-Sequenz beschreibt somit die Funktionsweise eines Informationsflusses. Zu sehen sind zwei Systemelemente mit jeweils einem Port vom Typ „peer“. Unterhalb der Ports startet die sog. Lebenslinie. Die Leserichtung dieses Diagramms ist von oben nach unten. Die Rauten sowie ihre Beschriftung durch Buchstaben beschreiben alternative Startvorgänge für den Informationsaustausch zwischen den Systemelementen. Folglich kann im Falle der „Alternativen A und B“ entweder „Systemelement 1“ oder „Systemelement 2“ eine Anfrage zur Synchronisation senden. Die Art und Weise wie die Anfrage initiiert wird, ist nicht Gegenstand des betrachteten Lösungsmusters. Die Anfrage wird vom jeweils anderen Systemelement positiv oder negativ beantwortet. Das Partialmodell Verhalten-Sequenz steht in enger Verbindung mit dem Partialmodell Verhalten-Zustände. Fällt die Antwort beispielsweise negativ aus, wechseln beide Elemente in den Zustand „inaktiv“ (abgebildet in dem Diagramm Verhalten-Zustände). Fällt die Antwort positiv aus, kommt es zur Zusammenarbeit. Beide Systemelemente wechseln in den Zustand „aktiv“. Dieser Zustand kann von beiden Systemelementen durch eine „Deaktivierung“ – im Partialmodell Verhalten-Sequenz dar-

gestellt durch „Alternative C und D“ – aufgehoben werden. Hierzu wird analog zur Synchronisationsanfrage in der unteren Hälfte der Lebenslinie eines der Systemelemente eine „Deaktivierung“ durchsetzen. Die Durchsetzung unterteilt sich in die aufeinanderfolgenden Schritte „Deaktivierung vorschlagen“ und die Bestätigung „Deaktivierung empfangen“. Ist die Deaktivierung durchgesetzt, kann die Kommunikation, die durch das Lösungsmuster repräsentiert wird, erneut von extern initiiert werden. Der zeitliche Ablauf startet dann erneut mit Beginn der Lebenslinie von oben nach unten. Je nach Anwendungsfall ist das Lösungsmuster im Rahmen des Systementwurfs auf die jeweilige Problemstellung anzupassen – sprich ggf. zu vereinfachen oder zu erweitern.

### Aggregation des Lösungsmusters „Zusammenarbeit synchronisieren“

Das zuvor beschriebene Beispiel stellt ein verhältnismäßig komplexes Zusammenspiel zwischen zwei Softwarekomponenten dar. Je komplexer ein identifiziertes Lösungsmuster ist, desto sinnvoller ist eine Dekomposition in elementarere Lösungsmuster bzw. Lösungsprinzipien. Die resultierenden Zusammenhänge veranschaulicht Bild 4-10. Die in der oberen Hälfte der Grafik befindlichen Lösungsmuster für den Systementwurf haben allesamt ihren Ursprung im fachdisziplinspezifischen Softwareentwurf. Sie beruhen folglich auf Koordinationsmustern für die Softwaretechnik. In Folge der Dekomposition entsteht eine Kaskade von Lösungsmustern für den Systementwurf, die an softwareintensiven Beispielen erläutert wird. Eine Verifikation dieser Gesetzmäßigkeiten und die Übertragung auf weitere Disziplinen folgt in Kapitel 4.3.2.2. Die bereits erwähnte Kaskade beginnt mit dem Lösungsmuster „Aktion befehlen“. Dieses Lösungsmuster ist Teil des übergeordneten Lösungsmusters „Informationsaustausch initiieren“. Dieses ist wiederum Teil des Lösungsmusters „Zusammenarbeit synchronisieren“. Die gezeigten Lösungsmuster werden im Folgenden erläutert.

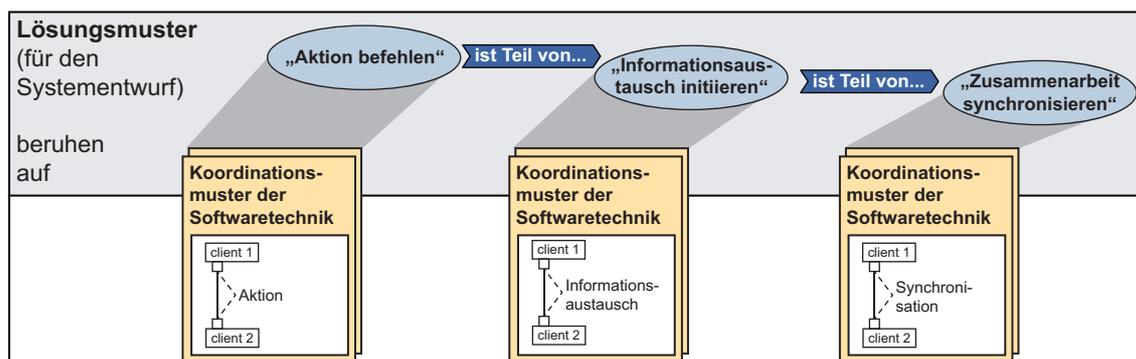


Bild 4-10: Aggregation des Lösungsmusters „Zusammenarbeit synchronisieren“

### Lösungsmuster „Aktion befehlen“

Einen ersten Anhaltspunkt für den Komplexitätsgrad eines Lösungsmusters liefert die Funktionshierarchie. Bei dem gewählten Beispiel (Bild 4-11) handelt es sich um eine Funktionshierarchie über lediglich zwei Ebenen. Hauptfunktion und zugleich Namensgeber ist „Aktion befehlen“. Diese unterteilt sich auf einer Hierarchieebene tiefer in die beiden Subfunktionen „Aufforderung senden“ und „Aufforderung empfangen“.

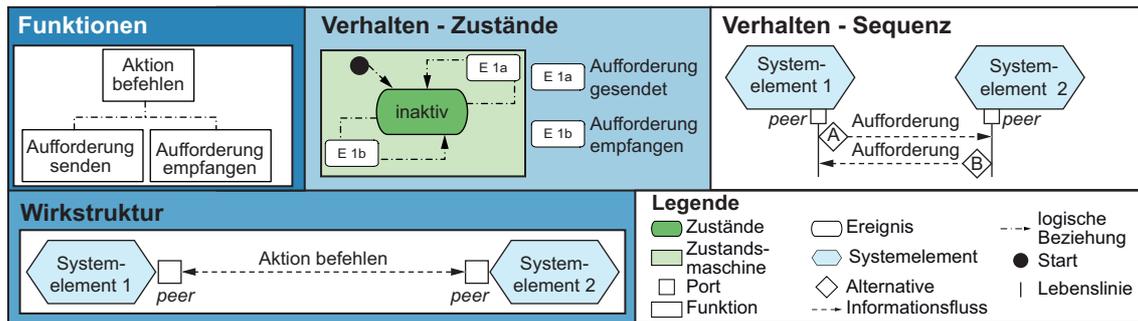


Bild 4-11: Aspekte des Lösungsmusters „Aktion befehlen“

Die Lösungsbeschreibung umfasst die Partialmodelle Wirkstruktur, Verhalten-Zustände und Verhaltens-Sequenz. Beide Systemelemente verfügen über einen Port vom Typ „peer“, was die Gleichberechtigung der Systemelemente widerspiegelt. Beide Systemelemente befinden sich zu Beginn der Kooperation in dem Zustand „inaktiv“. Sie haben jeweils die Möglichkeit, die Interaktion zu starten. Wird ein Sendevorgang von außen initiiert, sendet entweder „Systemelement 1 oder 2“ eine Aufforderung an das jeweils andere Systemelement. Die Reaktion auf die Anfrage ist für das jeweils andere Systemelement nicht mehr von Interesse.

#### Lösungsmuster „Informationsaustausch initiieren“:

Bild 4-12 veranschaulicht die ausgearbeiteten Aspekte des Lösungsmusters „Informationsaustausch initiieren“. Die Funktionshierarchie und somit die Problembeschreibung des Lösungsmusters erstreckt sich über drei Ebenen. Es handelt sich um einen Auszug aus dem Lösungsmuster „Zusammenarbeit synchronisieren“. Die Funktionshierarchie umfasst im Kern die Funktionalitäten rund um das „Senden und Empfangen von Anfrage und Antwort“, wobei die Antworten in „Zusagen“ und „Absagen“ zu unterscheiden sind.

Die prinzipielle Wirkungsweise des Lösungsmusters lässt sich sehr prägnant an dem Partialmodell Verhalten-Sequenz erkennen. Zu Beginn der Kooperation sind beide Systemelemente mit Ports vom Typ „peer“ im Zustand „inaktiv“. Beide Systemelemente haben wie in den Beispielen zuvor jeweils die Möglichkeit, eine Anfrage zu senden, um den Informationsaustausch zu beginnen. Das Systemelement, das die Anfrage gesendet hat, wechselt anschließend in den Zustand „auf Antwort warten“. Das andere Systemelement wechselt seinerseits in den Zustand „Anfrage auswerten“. Ist die Anfrage ausgewertet, positiv oder negativ, folgt als Reaktion wieder ein Informationsfluss zurück. Dem anderen Systemelement wird entweder eine „Zusage“ oder eine „Absage“ für den Austausch erteilt. Im Anschluss an die Interaktion wechseln beide Systemelemente in den Zustand „inaktiv“ und warten erneut auf einen Reiz von außen. Das untere Ende der Lebenslinie im Partialmodell Verhalten-Sequenz ist folglich erreicht.

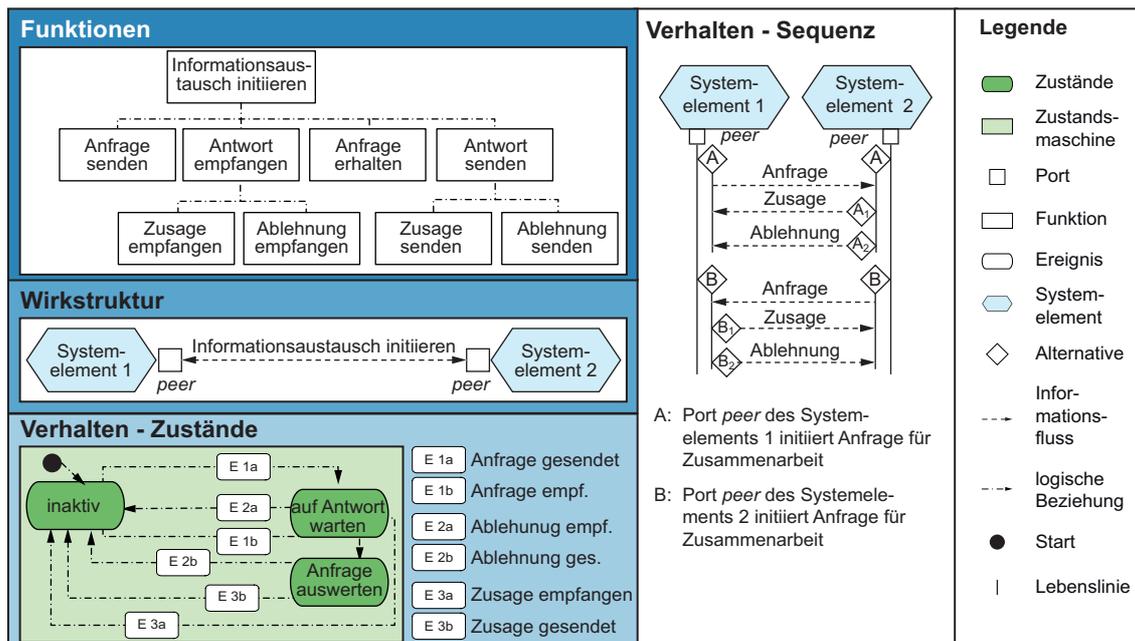


Bild 4-12: Aspekte des Lösungsmusters „Informationsaustausch initiieren“

#### 4.3.2.2 Multidisziplinäres Lösungsmuster „Servoantrieb“

Eine Anforderung an Lösungsmuster für den Systementwurf ist ihre Interdisziplinarität. Dementsprechend sind auch fachdisziplinspezifische Ansätze in die Terminologie der frühzeitigen Konzipierung zu transformieren, um eine Interaktion der Lösungsansätze untereinander zu gewährleisten. Letzteres verdeutlichen die zuvor beschriebenen Beispiele der Softwaretechnik in Kapitel 4.3.2.1. Im vorliegenden Kapitel wird ergänzend ein multidisziplinäres Lösungsmuster beschrieben, das maschinenbauliche und regelungstechnische Anteile enthält. Es handelt sich um das Lösungsmuster „Servoantrieb“. Die wesentlichen Aspekte – Funktion, Wirkstruktur und Merkmale – sind in Bild 4-13 abgebildet und werden im Folgenden kurz erläutert.

#### Problem:

Hauptfunktion eines Servoantriebs ist, wie bei vielen anderen elektrischen Antrieben, die Funktion „Drehmoment erzeugen“. Eine Besonderheit beim Servoantrieb ist im Vergleich zu Elektromotoren die bereits implementierte Regelung. Neben den beiden Funktionen „Magnetfeld erzeugen“ und „Drehmoment abstützen“ hat daher die Funktion „geregelten Stromfluss erzeugen“ eine ganz entscheidende Bedeutung. In der Regel gibt der Entwickler bei der Implementierung eines Servoantriebs lediglich die gewünschten Sollwerte für diejenigen Parameter vor, über die der Servoantrieb eingeregelt wird. Dieses wird physikalisch mit Hilfe des Stromflusses am Antrieb umgesetzt und durch die Funktionen „Stromfluss regeln“ und „elektrische Energie von externer Energiequelle“ in der Funktionshierarchie ausgedrückt. Die Funktion „Stromfluss regeln“ unterteilt sich weiter in Funktionen wie z.B. „Sollwerte zuführen“ oder „Stellgröße berechnen“.

Bild 4-13 zeigt zudem die wesentlichen Merkmale bzw. Parameter. Es wird zwischen zwei Wertarten unterschieden. Es gibt Parameter, dessen Ausprägungen vom Entwickler während des Systementwurfs ermittelt werden müssen. Hierzu zählen das „Nennmoment“, das „Maximalmoment“, die „Nennzahl“ und die „Motorkonstante“. Diese verdeutlichen ein wesentliches Charakteristikum eines Lösungsmusters, nämlich, dass es während des Entwurfs auf die jeweilige Problemstellung adaptiert werden muss. Ferner gibt es Parameter, für die Standardwerte existieren. Es handelt sich hierbei um gängige Werte für die „Netzspannung“ (z.B. 400 V) und die „Verzögerung des Stromregelkreises“ (z.B. 1 ms). In dem beispielhaft dargestellten Lösungsmuster sind die Werte in den Merkmalen daher schon vorgegeben. Selbstredend ist es möglich, die Lösung um diese Werte weiter zu generalisieren, um einen größeren Einsatzbereich abdecken zu können.

**Lösung:**

Die Lösung wird durch das Partialmodell Wirkstruktur abgebildet. Das in Bild 4-13 dargestellte weiße Systemelement im Hintergrund symbolisiert den „Servoantrieb“. Beziehungen nach außen existieren lediglich zu den Umfellelementen „Informationsverarbeitung“, „Energieversorgung“ und „Umwelt“ (mittels einer mechanischen Verbindung). Letztere ist erforderlich, um das erzeugte Drehmoment gegenüber der Umgebung abzustützen. Ausgangsgröße des Musters ist der Energiefluss „Drehmoment“.

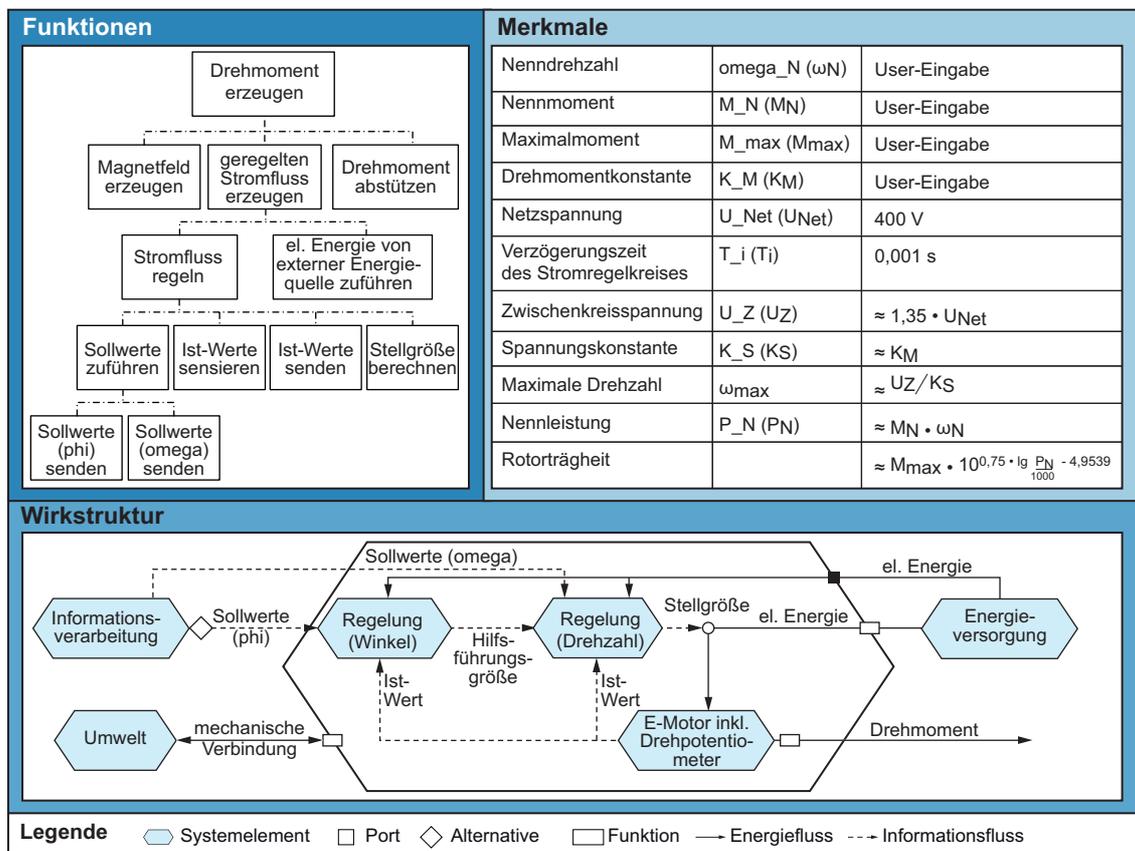


Bild 4-13: Aspekte des Lösungsmusters „Servoantrieb“

Die Besonderheit des Lösungsmusters liegt im Zusammenspiel der Regelung mit dem physikalischen Systemelement „*Elektromotor*“. Dieser beinhaltet eine ganze Reihe weiterer Systemelemente, die der Einfachheit halber in dem Lösungsmuster nicht detailliert abgebildet sind. Es handelt sich dabei in erster Linie um den sog. „*Anker*“ und den sog. „*Stator*“. Auf der „*Ankerwelle*“ befinden sich „*Drahtwicklungen*“, welche von dem „*Kommutator*“ mit einem Stromfluss versorgt werden. Der „*Sollstrom*“, der in den „*Drahtwicklungen*“ eingestellt wird, ist das Ergebnis der „*Regelung*“ und wird mittels sog. „*Bürsten*“ auf den „*Kommutator*“ übertragen. Die „*Ankerwelle*“ ist gegenüber dem „*Motorgehäuse*“ gelagert. Eine Drehbewegung der „*Ankerwelle*“ ist jedoch nur möglich, wenn sich diese in einem „*Magnetfeld*“ befindet. Das benötigte „*Magnetfeld*“ wird vom „*Stator*“ erzeugt, der fest mit dem „*Gehäuse*“ verbunden ist. Der „*Servoantrieb*“ verfügt zusätzlich über ein „*Drehpotentiometer*“ auf der „*Ankerwelle*“. Die „*Regelung*“ ist direkt mit dem „*Antrieb*“ verbunden. Er wird über die beiden Parameter „*Winkel*“ und/oder die „*Drehzahl*“ geregelt (vgl. Wirkstruktur). Die „*Regelung*“ erfolgt nach dem Prinzip einer Kaskadenregelung, die als Lösungsmuster der Regelungstechnik interpretiert werden kann und in Kapitel 3.2.3.4 beschrieben ist.

#### Aggregation des Lösungsmusters „*Servoantrieb*“

Wie bereits beschrieben kann ein Lösungsmuster teilweise oder vollständig aus weiteren Lösungsmustern aggregiert werden. In dem Beispiel „*Servoantrieb*“ handelt es sich um die Lösungsmuster „*Elektromotor*“ und „*Kaskadenregelung*“ (vgl. Bild 4-14). Im Gegensatz zu dem Beispiel aus dem vorangegangenen Kapitel baut die Kaskade somit nicht unmittelbar aufeinander auf. Vielmehr ist der „*Servoantrieb*“ ein Beispiel dafür, wie unabhängige Lösungsmuster miteinander verbunden werden können. Analog zu den beschriebenen Beispielen aus der Softwaretechnik beruhen die Lösungsmuster für den Systementwurf auf Lösungsprinzipien der partizipierenden Fachdisziplinen. Bei dem Elektromotor sind dies in erster Linie Prinzipien aus dem Maschinenbau und der Elektrotechnik. Die „*Kaskadenregelung*“ ist eine spezifische Herangehensweise in der Regelungstechnik. Eine Verfeinerung und Beschreibung weiterer charakteristischer Merkmale für Lösungsmuster für den Systementwurf fortgeschrittener mechatronischer Systeme folgt im anschließenden Kapitel.

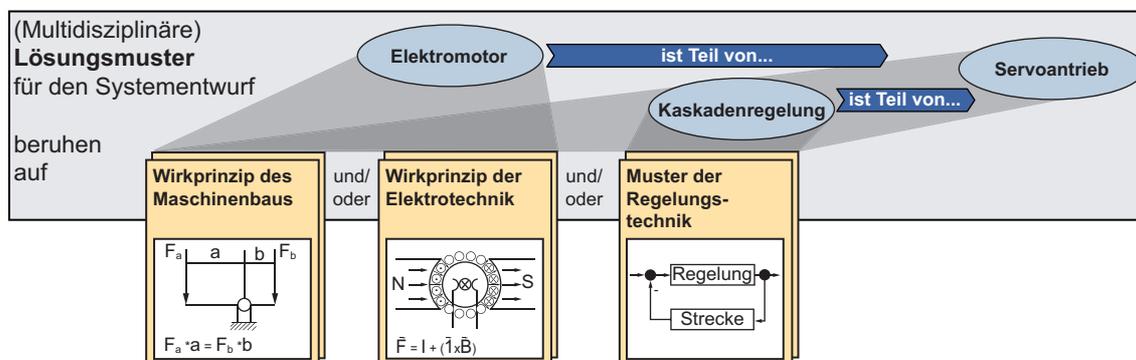


Bild 4-14: Aggregation des Lösungsmusters „*Servoantrieb*“

### 4.3.3 Charakterisierung von Lösungsmustern für den Systementwurf

Die der Arbeit zugrundeliegende Problemanalyse und der analysierte Stand der Technik haben gezeigt, dass Lösungsmuster entlang unterschiedlicher Dimensionen charakterisiert werden. Dies ist auch erforderlich, um den Umgang mit den Lösungsmustern im Rahmen der Produktentwicklung zu ermöglichen (identifizieren, speichern, anwenden, erweitern, in Beziehung zueinander setzen). Die in dieser Arbeit adressierten Lösungsmuster für den Systementwurf, deren Strukturierung Gegenstand des vorangegangenen Kapitels ist, besetzen die Schnittstellen zwischen fachdisziplinspezifischen Ansätzen und vereinen diese in geeigneter Art und Weise. Um den Ansprüchen an einen lösungsmusterbasierten Systementwurf gerecht zu werden, wird in diesem Kapitel ein speziell auf die Eigenschaften der Lösungsmuster abgestimmter multidimensionaler Wissensraum vorgestellt. Demnach lassen sich Lösungsmuster, die für den Systementwurf sowie für vor- und nachgelagerte Entwurfsschritte relevant sind, entlang der drei Dimensionen *Art der Wissensrepräsentation*, *Spezialisierung des Lösungswissens* und *Aggregation des Lösungswissens* beschreiben und voneinander abgrenzen.

#### 4.3.3.1 Dimension 1: Art der Wissensrepräsentation

Lösungsmuster dienen in erster Linie zur **Repräsentation relevanten Lösungswissens**. Ein Alleinstellungsmerkmal der im Rahmen dieser Arbeit betrachteten Lösungsmuster für den Systementwurf ist die Darstellung des Lösungswissens. Grundvoraussetzung für eine effiziente Einbindung relevanten Lösungswissens in den Entwurf ist die Repräsentation des Wissens in einer geeigneten Terminologie. Geeignet bedeutet in diesem Fall, dass zum jeweiligen Einsatzzeitpunkt der Lösungsmuster das Wissen analog zur eingesetzten Sprache bzw. Terminologie des zugehörigen Entwurfsschrittes repräsentiert wird. Diesbezüglich lässt sich auf höchster Ebene zwischen **textbasierter** bzw. dokumentenbasierter und **modellbasierter** Darstellung unterscheiden (vgl. Bild 4-15).

Textbasiert festgehaltene Lösungsmuster haben in der Regel eine längere Historie. Zu erwähnen sind u.a. Wirkprinzipien des Maschinenbaus oder der Elektrotechnik. Auf diese können die Entwickler in Form zahlreicher Konstruktionskataloge zurückgreifen. In der modernen Entwicklung mechatronischer Systeme werden Artefakte des Systems größtenteils durch computerinterne Modelle abgebildet. Beweggründe für die Verwendung von Modellen zum Entwurf und zur Eigenschaftsabsicherung des Systems sind die Kosten- und Zeitersparnisse beim Bau und Test realer Prototypen. Je nach Entwicklungszeitpunkt und Einsatzzweck unterscheiden sich die Modelle hinsichtlich ihres Formalisierungsgrads. Wie die Erläuterung der Entwicklung fortgeschrittener mechatronischer Systeme wie beispielsweise einem CPS in Kapitel 4.2.2.2 verdeutlicht, erfordert eine „*Disziplinübergreifende System-Ebene*“ die Nutzung **semiformaler** Modelle. In den darunterliegenden Ebenen, der „*Disziplinorientierten Mechatronik-Ebene*“ und der „*Disziplinintegrierenden Aspektspezifischen-Ebene*“, werden hingegen **formale**

Modelle genutzt. Diese sind nötig, um spezifische Eigenschaften wie z.B. das dynamische Verhalten von Mehrkörpersystemen zu simulieren und zu analysieren.

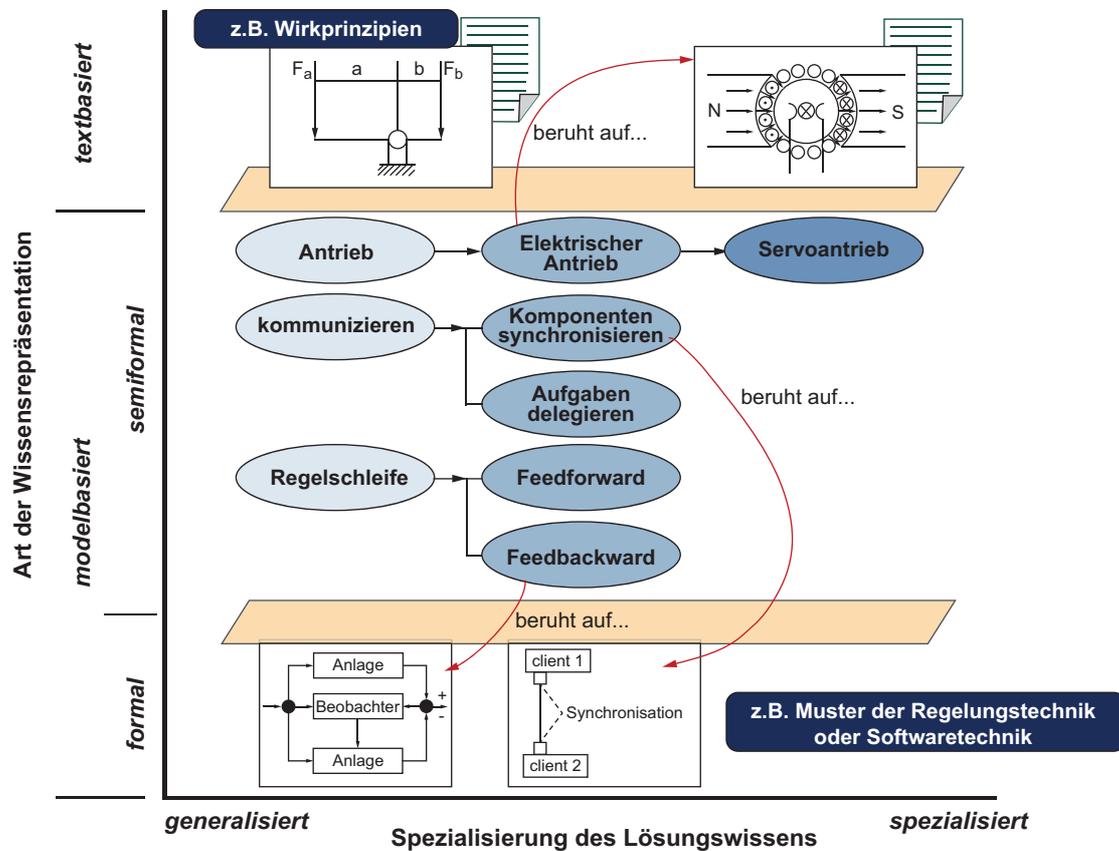


Bild 4-15: Zusammenhänge zwischen Lösungsmustern für den Systementwurf hinsichtlich „Wissensrepräsentation“ und „Spezialisierung“

Analog zum Formalisierungsgrad der Modelle in der Entwicklung verhält es sich mit den Modellen zur Repräsentation des zugehörigen Lösungswissens. So werden die Informationen in einem Lösungsmuster für den Systementwurf mit Hilfe semiformalen Modelle dargestellt während im Regelungs- oder Softwareentwurf formale Modelle in den Lösungsmustern oder in spezifischen Modellbibliotheken verwendet werden.

An dieser Stelle sind die Beziehungen zwischen den unterschiedlichen Ebenen hervorzuheben. Die beispielhafte Erläuterung von Lösungsmustern in den Kapiteln 4.3.2.1 und 4.3.2.2 hat gezeigt, dass Abhängigkeiten untereinander bestehen. So beruht beispielsweise das Lösungsmuster „Elektromotor“ auf dem Wirkprinzip „Lorentz-Kraft“ und das Lösungsmuster „Feedback“ auf regelungstechnischen Prinzipien. Ziel ist es, die Komplexität in den einzelnen Lösungsmustern für den Systementwurf so gering wie möglich zu halten. Hierzu ist es erforderlich, die vorhandenen Beziehungen mehrdimensional abzubilden und die Informationen in einem Lösungsmuster zu fokussieren.

#### 4.3.3.2 Dimension 2: Spezialisierung des Lösungswissens

Die zweite Dimension, entlang derer ein Lösungsmuster gegenüber anderen abgegrenzt werden kann, ist die **Spezialisierung des Lösungswissens**. Dies verdeutlicht das in Bild 4-15 dargestellte Beispiel von Antrieben. So ist ein „*Servoantrieb*“ beispielsweise ein „*elektrischer Antrieb*“, der wiederum ein „*Antrieb*“ ist. Die zweite Dimension ist erforderlich, um die kreative Leistung von Entwicklern bestmöglich zu unterstützen. Je höher der Generalisierungsgrad eines Lösungsmusters, desto weniger verliert sich der Entwickler, insbesondere zu Beginn einer Produktentwicklung, in unnötigen Detailfragen. Die zweite Dimension verläuft daher parallel zur fortschreitenden Konkretisierung im Entwurf.<sup>62</sup>

Es zeigt sich nach wie vor, dass Entwickler große Schwierigkeiten haben, von bekannten Denkmustern abzuweichen. Sie schaffen es oftmals nicht, z.T. fachfremde Lösungsprinzipien bei der Aufgabenbewältigung mit ins Kalkül zu ziehen. Dies führt dazu, dass sie sich auf bereits erfolgreich eingesetzte Lösungen beschränken und das Innovationspotential durch vorhandene Technologien nicht umfassend ausschöpfen. Aufgrund des hohen Zeit- und Kostendrucks verlassen die Entwickler zu früh eine lösungsneutrale Denkweise und wägen unterschiedliche Lösungskonzepte zu wenig ab.

Die im Rahmen dieser Arbeit definierte Dimension zum Spezialisierungsgrad wirkt dem entgegen. Je generalisierter ein Lösungsmuster ist, desto größer ist folglich auch sein Anwendungskontext. Bei der Definition von einem Lösungsmuster bewegt sich der Entwickler i.d.R. vom **Spezialisierten** zum **Generalisierten**. Dies verdeutlicht folgendes Beispiel, das in der zuvor genannten Richtung zu interpretieren ist: Ein 6-Achs-Roboter für das Punktschweißen in einer Fertigungslinie vom Audi A8 *ist ein* 6-Achs-Roboter für das Punktschweißen *ist ein* 6-Achs-Roboter zum Einsatz in einer Fertigung *ist ein* 6-Achs-Roboter *ist ein* Roboter. Ein einheitliches Generalisierungsniveau, das bei der Definition von Lösungsmustern erreicht werden muss, existiert nicht. Die Beziehungen sind vielmehr relativ zueinander zu beschreiben.

Ferner entfällt bei dem erarbeiteten Ansatz eine aufwändige Definition einer Klassifikation von Lösungsmustern. Diese resultiert zwangsläufig durch die festgehaltenen Relationen zwischen den Lösungsmustern. Auf diese Weise wird eine Übertragbarkeit des Ansatzes auf unterschiedliche Unternehmen und Branchen sichergestellt. Es handelt sich bei dem Lösungswissen um unternehmensinternes Know-how, was wiederum zu sehr individuellen Lösungsmustern und Beziehungen führen kann.

Nach einer entsprechenden rechnerinternen Implementierung des hier vorgestellten Wissensraums, haben die Entwickler die Möglichkeit, sich entlang der Beziehungen von den generalisierten zu den spezialisierten Lösungsmustern zu navigieren. Sie besitzen somit in der frühen Phase der Entwicklung ein möglichst facettenreiches Angebot

---

<sup>62</sup>Vgl. hierzu den Entwurfsraum nach GAUSEMEIER in [GP14, S. 27]

an generalisierten Lösungsmustern. Je nach Bedarf haben sie zudem die Möglichkeit, spezifisches Lösungswissen hinzuzuziehen.

#### 4.3.3.3 Dimension 3: Aggregation des Lösungswissens

Die dritte Dimension im multidimensionalen Wissensraum verdeutlicht Bild 4-16. Für ein besseres Verständnis ist die Ebene der Wissensrepräsentation nicht dargestellt. Wie bereits geschildert liegt der Schwerpunkt auf den Lösungsmustern für den Systementwurf, in denen das Wissen mit Hilfe von semiformalen Modellen abgebildet wird. Es handelt sich bei der dritten Dimension um die **Aggregation des Lösungswissens**. Diese Dimension orientiert sich u.a. an Erkenntnissen von SUHM (vgl. Kapitel 3.2.3.1). SUHM schlägt vor, komplexe maschinenbauliche Problemstellungen und zugehörige Lösungen in elementare Lösungsmuster zu gliedern. Dies erweitert ihren Einsatzbereich und ihre Anwendbarkeit durch die Entwickler.

Dieser Gedankengang wird für z.T. multidisziplinäre Lösungsmuster aufgegriffen und auf diese übertragen. Wie Bild 4-16 zeigt, verläuft die Dimension für die Aggregation des Lösungswissens vom **Elementaren** zum **Komplexen**. Elementare Lösungsmuster befinden sich beispielhaft im Vordergrund. Die schemenhaft eingezogenen Ebenen dienen lediglich der besseren Visualisierung der Zusammenhänge und verdeutlichen die wechselseitigen Beziehungen zueinander.

Ein Beispiel für ein elementares Lösungsmuster ist die Interaktionsform zweier Softwarekomponenten „*Aufgabe delegieren*“. Dieses Muster ist zugleich Bestandteil des komplexeren Lösungsmusters „*Zusammenarbeit synchronisieren*“ (vgl. Kapitel 4.3.2.1). Somit ist die Übertragbarkeit der Überlegungen von SUHM auf die Softwaretechnik gegeben. Ein Beispiel für den Zusammenhang zwischen Maschinenbau und Elektrotechnik ist ein „*elektrischer Antrieb*“, der wiederum das elementare Lösungsmuster „*Presspassung*“ enthält. Als Beispiel für ein komplexes Beispiel bestehend aus mehr als einem elementaren Lösungsmuster sind „*interagierende Deltaroboter*“ zu nennen. Es handelt sich um autark agierende Deltaroboter, die im Verbund eine spezifische Aufgabe erfüllen. Dieses Lösungsmuster setzt sich u.a. zusammen aus den Lösungsmustern „*Delta-Kinematik*“, „*elektrischer Antrieb*“ oder auch „*Zusammenarbeit synchronisieren*“.

Der Komplexitätsgrad der Lösungsmuster geht einher mit dem Komplexitätsgrad der zugrundeliegenden (Teil-)Lösungen, die den Ausgangspunkt bei der Definition der Lösungsmuster bilden. So ergeben sich analog zur Gliederung von Systemen und Subsystemen komplexere und elementarere Lösungsmuster. Der Wissensraum bietet Entwicklern die Möglichkeit, Relationen bzgl. des Komplexitätsgrads relativ zueinander festzuhalten.

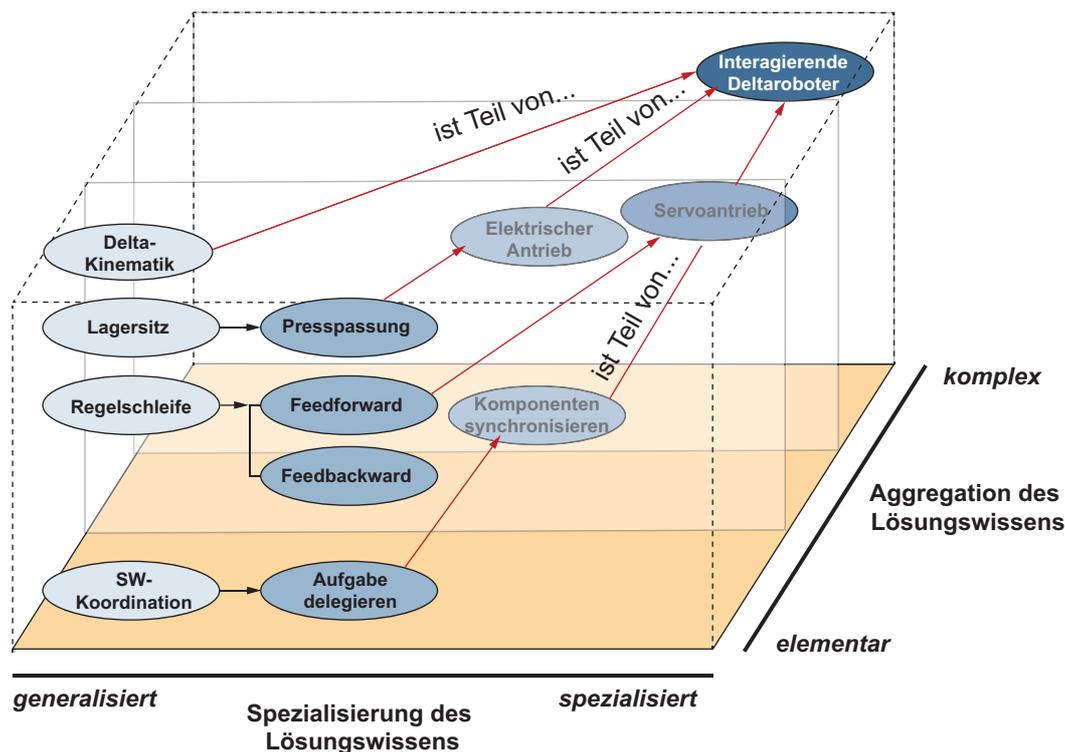


Bild 4-16: Zusammenhänge zwischen Lösungsmustern für den Systementwurf hinsichtlich Spezialisierung und Aggregation

Die Dimension „Aggregation des Lösungswissens“ unterstützt die beiden typischen Vorgehensweisen im funktionsorientierten Entwurf. Vorausgesetzt der Wissensraum ist mit Lösungsmustern gefüllt und die wechselseitigen Beziehungen sind beschrieben, sind sowohl ein top-down als auch ein bottom-up Suchvorgang möglich. Beim top-down Vorgehen navigiert der Entwickler beginnend bei den komplexen Lösungsmustern durch den Wissensraum. Beim bottom-up Vorgehen startet der Entwickler bei möglichst elementaren Lösungsmustern.

#### 4.3.3.4 Multidimensionaler Wissensraum

Zusammenfassend resultiert aus den Erläuterungen von Kapitel 4.3.3.1 bis 4.3.3.3 ein multidimensionaler Wissensraum, der die notwendigen Lösungsmuster für eine Produktentwicklung beinhaltet. Ferner bietet der Wissensraum die Möglichkeit, wechselseitige Beziehungen relativ zueinander abzubilden. Bild 4-17 zeigt den multidimensionalen Wissensraum mit seinen bereits erläuterten Dimensionen. Der Raum ist beispielhaft mit etablierten Ansätzen gefüllt. Vor der Erläuterung wird jedoch kurz auf das Thema der Multidimensionalität eingegangen.

#### Multidimensionalität

Aufgrund der zunehmenden Interdisziplinarität in der Entwicklung fortgeschrittener mechatronischer Systeme verschmelzen Disziplinengrenzen und zugehörige Sichten wie

Struktur, Verhalten oder Gestalt zunehmend. Die in Kapitel 4.3.2 vorgestellten Beispiele verdeutlichen dies. Die im Rahmen dieser Arbeit entwickelte einheitliche Strukturierung von Lösungsmustern wird diesen Gegebenheiten insofern gerecht, da die Aspekte des Musters je nach Schwerpunkt mit Struktur- und/oder Verhaltensmodellen dargestellt werden können. Deshalb werden unterschiedliche Sichten oder Fachdisziplinen nicht explizit differenziert. Sie sind in einer weiteren – nicht visualisierten – Dimension vielmehr als Filter zu interpretieren.

Während des Systementwurfs ist es durchaus sinnvoll, den Wissensraum und seine darin enthaltenen Muster nach fachdisziplinspezifischen Aspekten zu filtern, z.B. wenn die Auslegung des dynamischen Verhaltens des Gesamtsystems im Vordergrund steht. Hierbei bietet es sich z.B. an, gezielt nach gestalterischen Aspekten zu suchen, da diese die Dynamik vorrangig beeinflussen können. Im weiteren Verlauf der Entwicklung, wenn von der fachdisziplinübergreifenden in die disziplinorientierte Ebene gewechselt wird, sollte ebenfalls ein Filter bzgl. Fachdisziplinen verwendet werden. Dies erleichtert die Navigation ausgehend von den bereits verwendeten Lösungsmustern auf Gesamtsystemebene.

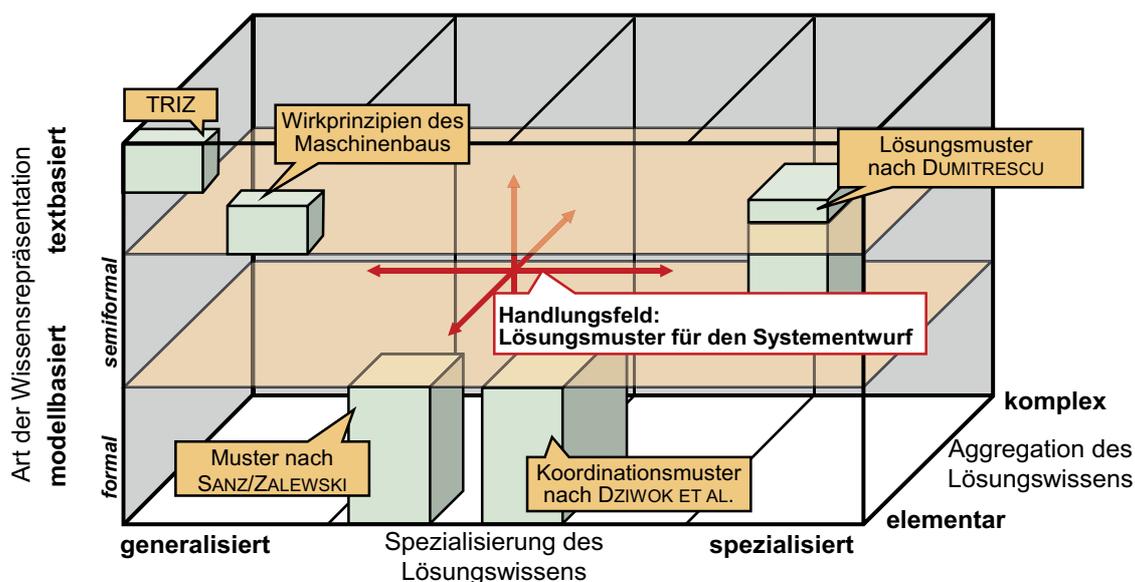


Bild 4-17: Multidimensionaler Wissensraum für Lösungsmuster in der Produktentwicklung

### Handlungsfeld: Lösungsmuster für den Systementwurf

Das Handlungsfeld der Lösungsmuster für den Systementwurf verdeutlicht Bild 4-17. In Bezug zu etablierten Ansätzen zur Wiederverwendung von Lösungswissen und zur Unterstützung des kreativen Entwicklungsprozesses sind die adressierten Lösungsmuster auf der semiformalen Ebene einzuordnen. Eine Ebene höher, nämlich im Bereich der textbasierten Ansätze, befinden sich u.a. TRIZ (vgl. Kapitel 2.4.4) und Wirkprinzipien des Maschinenbaus. Durch das Prinzip der Widerspruchsanalyse und der starken Abstraktion von realen Problemstellungen ist TRIZ auf den Großteil technischer Problem-

stellungen übertragbar. Die in TRIZ enthaltenen Prinzipien sind somit generalisierter als z.B. Wirkprinzipien, die maschinenbauliche Problemstellungen fokussieren. Einen höheren Spezialisierungsgrad weisen die Lösungsmuster für selbstoptimierende Systeme nach DUMITRESCU auf. Diese beinhalten textuell beschriebene Verfahren zur Integration von kognitiven Funktionen in mechatronische Systeme. Die daraus resultierenden selbstoptimierenden Systeme stellen derzeit noch einen sehr geringen Anteil mechatronischer Systeme dar. Die Lösungsmuster nach DUMITRESCU enthalten ferner Lösungsbeschreibungen mit der Spezifikationstechnik CONSENS. Dies erklärt die Ausdehnung bis in die semiformale Ebene.

Verglichen mit DUMITRESCU sind die Koordinationsmuster nach DZIWOK generalisierter. Die Koordinationsmuster, in denen das Lösungswissen durch formale Mechatronik-UML-Modelle dargestellt wird, dienen der Synthese von echtzeitkritischer ereignisdiskreter Kommunikation zwischen Softwarekomponenten. In Bezug auf die Architektur intelligenter technischer Systeme (vgl. Kapitel 2.2.3), dem OCM, ist der Einsatzbereich der Koordinationsmuster der reflektorische Operator. Diese Architekturebene findet sich weitaus häufiger in technischen Systemen als der kognitive Operator (Lösungsmuster nach DUMITRESCU). Ein weiterer Ansatz, der eine formale Wissenspräsentation enthält, sind die Lösungsmuster der Regelungstechnik nach SANZ/ZALEWSKI oder FÖLLINGER. Diese Muster sind Bestandteil eines jeden mechatronischen Systems.

Die im Rahmen dieser Arbeit entwickelten Lösungsmuster besetzen die bis dato nicht abgedeckte Mitte des Wissensraums. Die i.d.R. mechatronischen Lösungsmuster ergänzen die Lösungsmuster für den kognitiven Operator nach DUMITRESCU bei der ganzheitlichen Spezifikation eines fortgeschrittenen mechatronischen Systems. Als elementare Muster beinhalten sie auf der einen Seite Bestandteile einzelner Fachdisziplinen, z.B. der Softwaretechnik, sind aber im Vergleich zu den Koordinationsmustern auf die Gegebenheiten des Systementwurfs adaptiert. Auf der anderen Seite kombinieren sie die Lösungsansätze fachdisziplinübergreifend. Es sind somit komplexe mechatronische Lösungsmuster, die Bestandteile des Grundsystems, des Controllers und des reflektorischen Operators kombiniert abbilden.

### **Visualisierung der Beziehung im Lösungsmuster**

Um die Komplexität für das interdisziplinäre Entwicklerteam so gering wie möglich zu halten, verfügt ein Lösungsmuster für den Systementwurf über spezifische Piktogramme. Diese symbolisieren die wechselseitigen Beziehungen zwischen Lösungsmustern für den Systementwurf als auch zu Lösungsmustern der angrenzenden Bereiche im Wissensraum. Je nach Ausprägung des Lösungsmusters werden die Piktogramme ein- oder ausgeblendet. Die Piktogramme und deren Bedeutung zeigt Bild 4-18. Sie orientieren sich an den drei Dimensionen im Wissensraum. Auf der horizontalen Achse symbolisieren Ellipsen mit unterschiedlicher Farbintensität den Generalisierungs- bzw. Spezialisierungsgrad. Die Beziehungen auf der vertikalen Achse verdeutlichen einen möglichen Bezug zu textbasierten oder formalen modellbasierten Ansätzen. Die Symbole auf der

diagonalen Achse zeigen, ob das betrachtete Lösungsmuster elementare Muster beinhaltet (-) oder ob es in mindestens einem komplexeren Lösungsmuster aggregiert ist (+).<sup>63</sup>

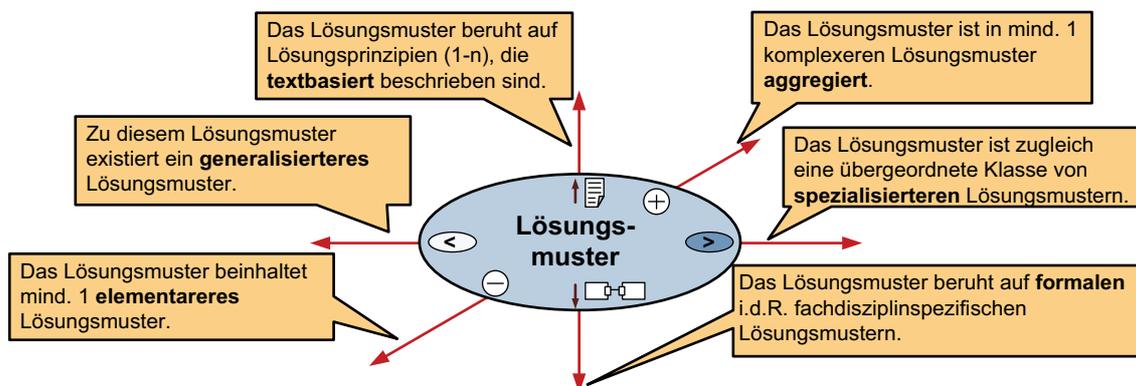


Bild 4-18: Charakteristische Zusammenhänge zwischen Lösungsmustern für den Systementwurf

#### 4.3.3.5 Zusammenspiel zwischen Wissensraum und Entwurfsebenen

Eine Anforderung an den Wissensraum ist die nahtlose Interaktion mit dem Entwurfsraum, bzw. den charakteristischen Entwurfsebenen in der Entwicklung fortgeschrittener mechatronischer Systeme. Das Zusammenspiel dieser Ebenen mit den Lösungsmustern im Wissensraum zeigt Bild 4-19. Im linken Bereich der Grafik befindet sich symbolisch der Wissensraum und im rechten Bereich die Entwurfsebenen; „Disziplinübergreifende System-Ebene“ und „Disziplinorientierte Mechatronik-Ebene“. Die zunehmende Konkretisierung in den Entwurfsebenen verläuft von links nach rechts. Da die Lösungsmuster in erster Linie Syntheseschritte unterstützen, wird der Einfachheit halber die „Disziplinintegrierende Aspektspezifische-Ebene“ vernachlässigt. Es wird bei der theoretischen Erklärung der Abläufe nachfolgend vorausgesetzt, dass Lösungsmuster im Wissensraum vorliegen.

#### Disziplinübergreifende System-Ebene

Der Entwurf auf der „Disziplinübergreifenden System-Ebene“ beginnt, wie bereits in Kapitel 4.2.2.2 skizziert, mit der sog. Zielbestimmung. Im Rahmen der Aufgabenanalyse wird die Aufgabenstellung abstrahiert, um den Kern der Entwicklungsaufgabe zu identifizieren. Anschließend wird eine Umfeldanalyse durchgeführt, in der die Randbedingungen und Einflüsse auf das System ermittelt werden. Die gesammelten Erkenntnisse werden nach der Definition aller möglichen Anwendungsszenarien in einer Anforderungsliste festgehalten. Diese ist Basis für die Definition der erforderlichen Funktionalität des Gesamtsystems in Form einer Funktionshierarchie.

<sup>63</sup> In einer potentiellen tooltechnischen Umsetzung dieses Konzepts, das nicht mehr Gegenstand der vorliegenden Arbeit ist, können die Entwickler mit Hilfe der Piktogramme durch den Wissensraum navigieren.

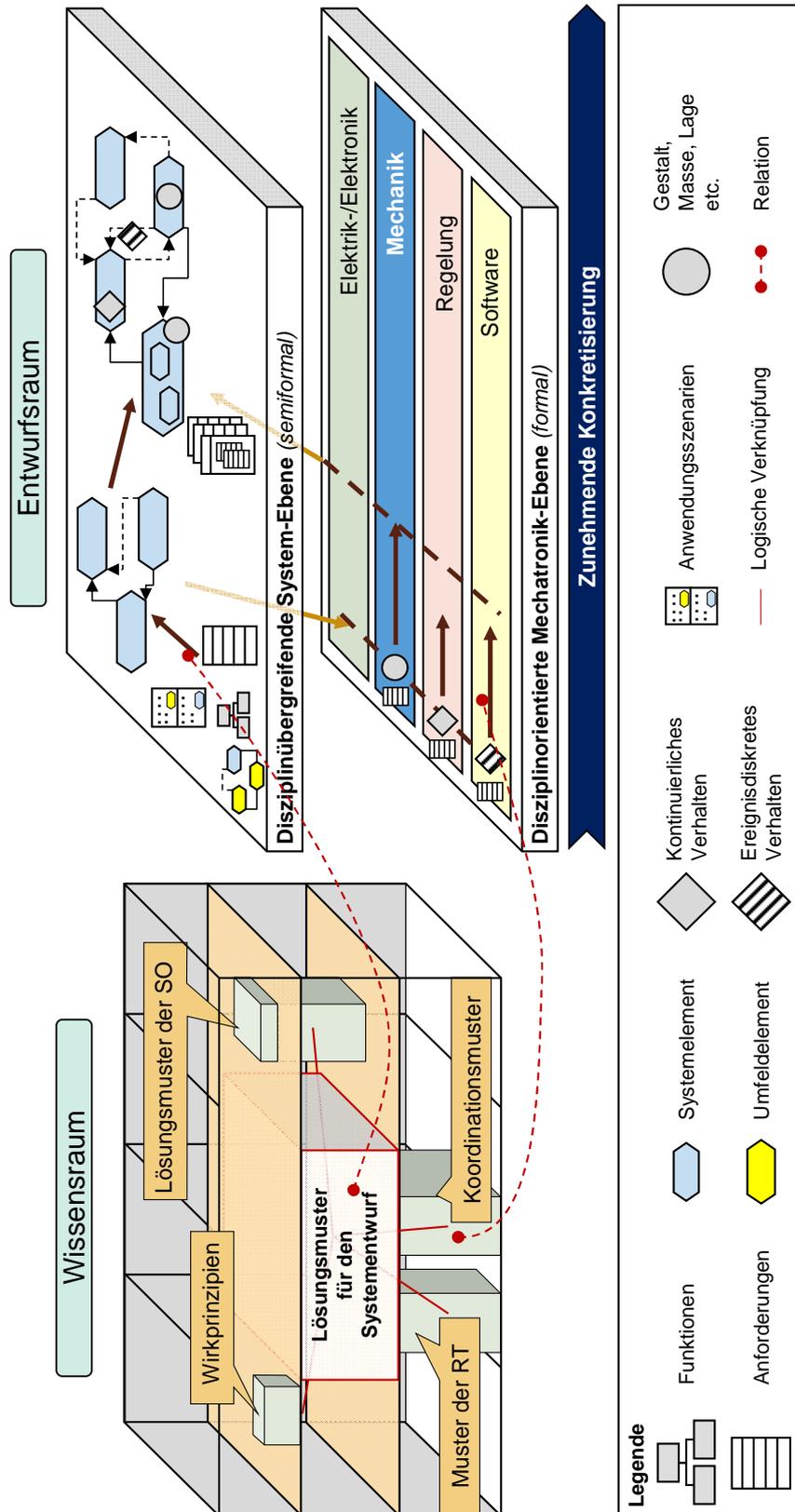


Bild 4-19: Zusammenspiel zwischen dem multidimensionalen Wissensraum und den Entwurfsebenen in der Entwicklung fortgeschrittener mechatronischer Systeme (Auszug)

Vor Beginn der Synthese erfolgt die Suche nach Lösungsmustern im Wissensraum auf Basis der zuvor erarbeiteten Anforderungen und Funktionen. Nach der Auswahl und Kombination der Lösungsmuster folgt die eigentliche Synthese. Hierzu werden die Wirkstruktur und Verhaltensmodelle der Lösungsmuster genutzt, um effizient das Produkt auf Gesamtsystemebene zu spezifizieren. Die Lösungsbeschreibung wird an die jeweilige Problemstellung angepasst und spezifisch ausgearbeitet<sup>64</sup>. Je nach Vorgehen im Entwurf (bottom-up oder top-down) und Größe des Gesamtsystems wird dieses modularisiert und/oder integriert. Die Konkretisierung der Struktur- und Verhaltensmodelle erfolgt initial bis zu dem Zeitpunkt, ab dem fachdisziplinspezifische Details erforderlich werden, um das System im Hinblick auf unterschiedliche Gesichtspunkte zu analysieren. Dies kann zum Beispiel die Analyse des dynamischen Systemverhaltens sein. In diesem Fall wird es erforderlich, vereinfachte Regelungsschleifen oder Softwarekommunikation formal zu spezifizieren.

Der Zeitpunkt, ab dem fachdisziplinspezifische Informationen erforderlich werden, ist von System zu System unterschiedlich. In jedem Fall folgt ein hochgradig iterativer Prozess zwischen der „Disziplinübergreifenden System-Ebene“ und der „Disziplinorientierten Mechatronik-Ebene“.

### **Disziplinorientierte Mechatronik-Ebene**

Ein effizienter Übergang in die „Disziplinorientierte Mechatronik-Ebene“ wird maßgeblich durch die festgehaltenen Verbindungen im Wissensraum sichergestellt. So werden in einem ersten Schritt die bereits spezifizierten Informationen von der disziplinübergreifenden Ebene genutzt und in die Mechatronik-Ebene transformiert. Von hoher Bedeutung sind hierbei in erster Linie Informationen zu den Schnittstellen, innerhalb einer Disziplin und vor allem disziplinübergreifend. Ferner werden die Informationen zu den ausgewählten Lösungsmustern übermittelt. Lösungskonzepte, die mindestens zwei Fachdisziplinen betreffen, sind bereits bekannt und werden in den Disziplinen konkretisiert. Ein Beispiel ist die Sensierung einer nicht unmittelbar messbaren Größe durch das Lösungsmuster „Beobachter“ (Regelungstechnik). Dieses besitzt Schnittstellen zum Grundsystem, also zu den Fachdisziplinen Maschinenbau und Elektrotechnik. Des Weiteren sind Schnittstellen zur ereignisdiskreten Kommunikation der Softwaretechnik denkbar.

Die Konkretisierung der spezifischen Inhalte geschieht innerhalb der Disziplinen unter Verwendung fachdisziplinspezifischer Lösungsmuster. So werden beispielsweise in der Mechanik erste Gestaltentwürfe erstellt. Die Regelungstechnik hingegen greift auf bekannte Übertragungsglieder zurück, um die Regelung und beispielsweise den Beobachter auszulegen. Erforderliche Parametrierungen werden sukzessive verfeinert. Analog

---

<sup>64</sup>In diesem Kapitel steht das übergeordnete Vorgehen und Zusammenspiel mit der fachdisziplinorientierten Mechatronik-Ebene im Vordergrund. Eine detaillierte Beschreibung, die Dritte in die Lage versetzt lösungsmusterbasiert ein System zu entwerfen, ist Gegenstand der Kapitel 4.5 und Kapitel 5.2.

zur Einschränkung des Suchraums bei den Lösungsmustern für den Systementwurf erfolgt auch für die fachdisziplinspezifischen Lösungsmuster eine Art Kettensuche. Diese resultiert aus dem steigenden Detaillierungsgrad in der Anforderungsdefinition, die den Anwendungskontext der zur Verfügung stehenden Lösungsmuster immer weiter einschränkt.

Die Zusammenführung von Erkenntnissen, die für weitere Disziplinen relevant sind, erfolgt fortlaufend auf der übergreifenden Ebene. Auf der einen Seite werden spezifische Schnittstelleninformationen und relevante Anforderungen für das Gesamtsystem transformiert. Auf der anderen Seite werden genutzte Lösungsmuster übermittelt, um die entsprechenden Verbindungen im Wissensraum langfristig zu sichern. Neben einer Plattform für die Kommunikation und Kooperation im Rahmen der Entwicklung ermöglicht der Wissensraum eine **Plattform sowohl zum Austausch über fachdisziplinspezifische als auch über fachdisziplinübergreifende Lösungsansätze**. Der Wissensraum stellt somit ein für die Entwicklung mechatronischer Systeme umfassenden Speicher für wiederverwendbare Lösungsansätze/-muster dar und vereint diese zu einem kohärenten Ganzen.

#### 4.4 Vorgehensmodell zur Identifizierung von Lösungsmustern für den Systementwurf

Dieses Kapitel beschreibt ein *Vorgehensmodell zur Identifizierung von Lösungsmustern für den Systementwurf* auf Basis eines bestehenden Produkts. Ziel ist die bedarfsgerechte Bereitstellung von einst implizit vorliegendem Lösungswissen. Den Kern bilden die in Kapitel 4.3 erarbeiteten Eigenschaften der neuartigen Lösungsmuster für den Systementwurf. Es wird beschrieben, wie die relevanten Wissensinhalte für den disziplinübergreifenden Entwurf externalisiert und repräsentiert werden. Auf diese Weise wird es zukünftig möglich, ein interdisziplinäres Entwicklerteam adäquat in der frühen Phase eines Entwurfs mit Lösungsmustern zu unterstützen. Voraussetzung für die Identifizierung eines Lösungsmusters ist die regelhafte Wiederholung bestimmter Zusammenhänge. Diese beruhen oftmals auf langjähriger Expertise und manifestieren sich u.a. in erarbeiteten Entwicklungsdokumenten oder Modellen. Eingangsinformationen für das im Folgenden vorgestellte Vorgehen sind daher Modelle von etablierten (Teil-)Lösungen, die dazugehörigen Dokumente und das in der Regel disziplinspezifische Know-how der Entwickler. In einer retrospektiven Analyse werden vorliegende Informationen in die Terminologie des Systementwurfs überführt und bestehende Zusammenhänge aufgedeckt. Identifiziertes Lösungswissen lässt sich auf diese Weise fachdisziplinübergreifend verstehen und zukünftig in anderen Entwicklungsprojekten frühzeitig einsetzen.

Das Vorgehensmodell gibt einen Überblick über die einzelnen Phasen, die wesentlichen Aufgaben und zu erarbeitenden Resultate (vgl. Bild 4-20). Zudem wird die Reihenfolge der Durchführung der Phasen festgelegt. An dieser Stelle ist darauf hinzuweisen, dass es

sich um eine sequentielle und idealtypische Darstellung handelt, deren tatsächliche Anwendung aber Iterationen und auch Rücksprünge in vorangegangene Phasen zulässt.

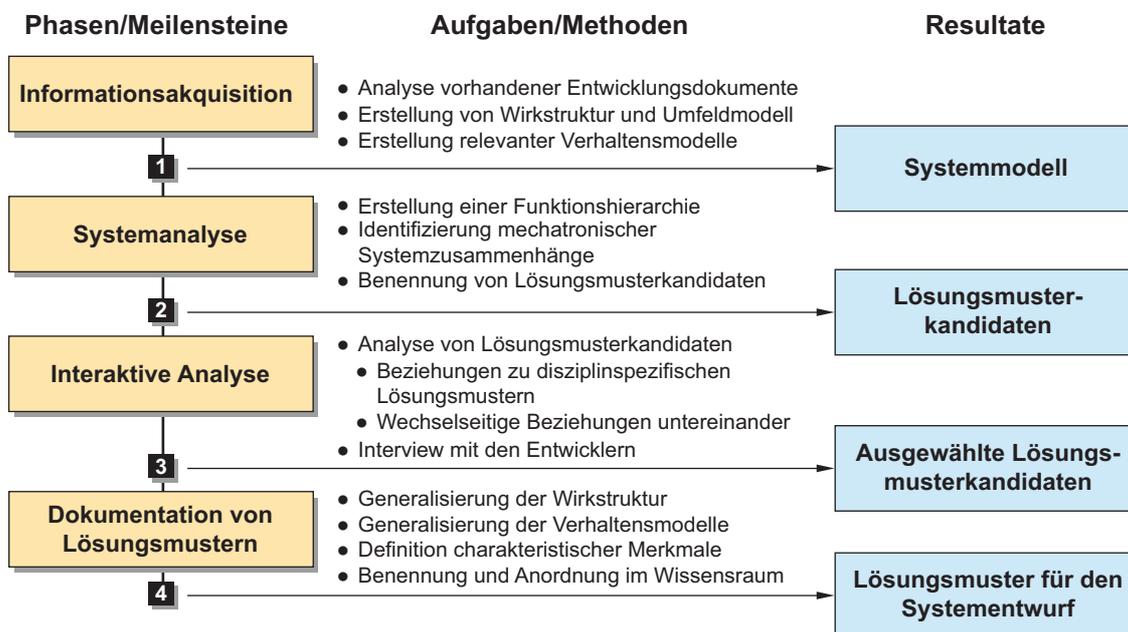


Bild 4-20: Vorgehensmodell zur Identifizierung von Lösungsmustern für den Systementwurf

#### 4.4.1 Phase 1: Informationsakquisition

Im Rahmen der ersten Phase erfolgt die Informationsakquisition. Diese Phase ist notwendig, da Unternehmen unterschiedlicher Branchen vor dem Hintergrund der zunehmenden Mechatronisierung noch nicht vollumfänglich fachdisziplinübergreifend denken und handeln. Der Anwender des Vorgehensmodells muss sich daher initial ein Überblick über das zu analysierende System und alle vorhandenen Informationen verschaffen. Es werden Fragestellungen geklärt wie beispielsweise: Gibt es eine fachdisziplinübergreifende Systembeschreibung? Gibt es eine funktionale Gliederung des Systems? Wie ist die mechanische Architektur, gibt es einen Baukasten? Wie ist die Architektur der Software?

Insbesondere in großen Unternehmen sind notwendige Informationen über Abteilungen hinweg und in vielen Entwicklungsunterlagen verteilt. Grundsätzlich müssen Informationen aus allen Disziplinen betrachtet werden. Aus der Konstruktion sind z.B. Informationen über die physische Baustruktur zu beschaffen. Diese lassen sich beispielsweise aus Montagevorranggraphen oder 3D-CAD-Daten entnehmen. Ferner sind Wirkzusammenhänge über die Fachdisziplinen hinweg aufzudecken. Dazu müssen Informationen über Signal- und Energieflüsse aus der Automatisierung – oder entsprechenden Abteilungen zur Implementierung der Informationsverarbeitung – betrachtet werden.

Auf Basis der vorhandenen Informationen entsteht die weitere Vorgehensstrategie, um Lösungsmuster für den Systementwurf zu identifizieren. In den folgenden Ausführun-

gen zur ersten Phase wird davon ausgegangen, dass eine vollumfängliche mechatronische Vorgehensweise bei der Erstellung des Produkts nicht vorliegt. Ist dies jedoch der Fall und es existiert das sog. Systemmodell, kann die Erstellung der erforderlichen Partialmodelle übersprungen werden.

Für die Abbildung des Systems und der internen Wirkzusammenhänge wird die Spezifikationstechnik CONSENS eingesetzt. Es werden sowohl die Wirkstruktur als auch die erforderlichen Verhaltensmodelle erstellt. Eine besondere Herausforderung liegt in der Wahl der Modellierungstiefe und der damit einhergehenden Hierarchisierung. Vor allem bei sehr komplexen Systemen bietet sich die Abgrenzung von Subsystemen bzw. logischen Gruppen an, um eine effektive Identifizierung von Lösungsmustern zu ermöglichen. Zusammenfassend wird in der Phase folgendes **Hilfsmittel** eingesetzt:

- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)

Das **Resultat** der Phase „*Informationsakquisition*“ ist ein Systemmodell des zu analysierenden Systems. Es umfasst die Abbildung der Struktur und des Verhaltens sowie deren wechselseitigen Beziehungen.

#### 4.4.2 Phase 2: Systemanalyse

Ziel der zweiten Phase ist die Identifizierung von Lösungsmusterkandidaten. Einen ersten Anhaltspunkt bei der Identifizierung von Lösungsmustern liefern dabei mechatronische Funktionseinheiten innerhalb des Systems. Gemäß der Definition von Lösungsmustern erfüllen diese eine bestimmte Funktion, oder Kombination von Funktionen.

In einem ersten Schritt der Systemanalyse wird daher eine funktionale Beschreibung des Systems aufgestellt. Dies erfolgt in Form einer Funktionshierarchie. Im Gegensatz zur Erstellung einer Funktionshierarchie mit zunehmender Konkretisierung im Rahmen des Entwurfs beginnt die retrospektive Modellierung *nicht* bei der Hauptfunktion. Es handelt sich hingegen um ein *Bottom-up-Vorgehen*. Die Aufnahme der Funktionen orientiert sich dabei an den bereits vorhandenen Strukturierungen im System. Dies sind auf der einen Seite physikalische Module oder Baugruppen und auf der anderen Seite informationsverarbeitende Module bzw. Gruppen von Komponenten. Je nach Hierarchisierung aus Phase 1 werden für die Systemelemente in der Wirkstruktur und in den Verhaltensmodellen die zugehörigen Funktionen aufgenommen.

Abschließend erfolgt die Analyse des Systems hinsichtlich funktionaler Zusammenhänge. Im besten Fall existiert eine 1:1-Beziehung zwischen physikalischen und informationsverarbeitenden Modulen, so dass sich mechatronische Funktionseinheiten mit einer übergeordneten Funktion direkt als Lösungsmusterkandidat ableiten lassen. Andernfalls gilt es, funktionale Zusammenhänge zwischen physikalischen und informationsumsetzenden Systemelementen auf Basis der Funktionshierarchie herauszustellen. Zusammenhängende Elemente werden samt Verhaltensbeschreibung und der zugehörigen

Funktion extrahiert und in die Strukturierung der Lösungsmuster für den Systementwurf überführt. Die für diese Phase erforderlichen **Hilfsmittel** sind:

- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)
- Einheitliche Strukturierung von Lösungsmustern für den Systementwurf (vgl. Kapitel 4.3.1)

Als **Resultat** der Phase „*Systemanalyse*“ liegen Lösungsmusterkandidaten vor. Kandidat bedeutet, dass eine Funktion bzw. Kombination von Funktionen als Problembe-  
schreibung vorliegt. Die Lösungsbeschreibung umfasst den zugehörigen Ausschnitt aus  
der Wirkstruktur und den Verhaltensmodellen. Die Problem- und Lösungsbeschreibung  
ist folglich zu diesem Zeitpunkt noch nicht generalisiert und der Kandidat kann ggf.  
verworfen werden. Dies klärt sich in den folgenden Phasen.

### 4.4.3 Phase 3: Interaktive Analyse

Aufbauend auf Phase 2 erfolgt eine interaktive Analyse der Lösungsmusterkandidaten.  
Diese ist notwendig, da sich in der Regel in den LM-kandidaten langjährige Erfahrung  
einzelner Entwickler oder Entwicklerteams verbirgt. Verwendete Lösungsmuster in  
einem Unternehmen sind meist historisch gewachsen und einzelnen Fachdisziplinen  
zuzuordnen. Die angestrebten Lösungsmuster für den Systementwurf beanspruchen  
ihren Neuheitsgrad dahingehend, dass sie Lösungsprinzipien der einzelnen Fachdiszi-  
plinen in einem übergeordneten Muster vereinen. Je unabhängiger die einzelnen Fachdis-  
ziplinen traditionell an abstrakten Lösungskonzepten gearbeitet haben (z.B.: auf Basis  
von Wirkprinzipien oder Wirkmustern der Softwaretechnik) desto wichtiger wird eine  
fachdisziplinübergreifende Analyse der zuvor identifizierten LM-kandidaten. Diese  
muss in interaktiven Diskussionen oder Workshops erfolgen, da die Erfahrung der Mit-  
arbeiter maßgeblich für Definition der Lösungsmuster erforderlich ist.

Das erste Analysekriterium sind daher die Beziehungen zu fachdisziplinspezifischen  
Lösungsprinzipien bzw. -mustern. Diese sind je nach Unternehmen bereits mehr oder  
weniger ausführlich dokumentiert. Mit Hilfe der im Rahmen dieser Dissertation erarbei-  
teten charakteristischen Dimensionen von Lösungsmustern werden wechselseitige Be-  
ziehungen herausgearbeitet.

In einem weiteren Schritt werden die Beziehungen zwischen den Lösungsmusterkandi-  
daten für den Systementwurf untersucht. Dies geschieht zum einen durch die direkte  
Analyse auf Basis der in Phase 1 erstellten Wirkstruktur. Die bereits spezifizierten Be-  
ziehungen zwischen den Systemelementen in der Wirkstruktur geben einen ersten Auf-  
schluss über die Vernetzung der LM-kandidaten. Zum anderen werden die LM-  
kandidaten hinsichtlich ihrer Beziehungen zueinander in den multidimensionalen Wis-  
sensraum überführt. Die Charakterisierung konzentriert sich dabei über die gesamte  
Phase 3 hinweg an den Dimensionen „Art der Wissensrepräsentation“ und „Aggregati-  
on des Lösungswissens“. Folgende **Hilfsmittel** werden in dieser Phase eingesetzt:

- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)
- Einheitliche Strukturierung von Lösungsmustern für den Systementwurf (vgl. Kapitel 4.3.1)
- Beziehungen zwischen den Lösungsmustern (vgl. Kapitel 4.3.3)
- Multidimensionaler Wissensraum (vgl. Kapitel 4.3.3.4)

Das **Resultat** der Phase „*interaktive Analyse*“ sind ausgewählte Lösungsmusterkandidaten, die hinsichtlich ihrer Vernetzung untereinander und zu fachdisziplinspezifischen Lösungsmustern analysiert sind.

#### 4.4.4 Phase 4: Dokumentation von Lösungsmustern

In einer abschließenden Phase werden die Lösungsmuster für den Systementwurf definiert und dokumentiert. Hierzu sind die detaillierten Informationen aus den Partialmodellen Wirkstruktur, Verhalten und Funktionen in die vorgegebene Strukturierung von Lösungsmustern zu überführen. Ferner sind spezifische Merkmale für das jeweilige Lösungsmuster zu definieren.

Während der Definition eines Lösungsmusters und der Überführung von Informationen aus dem Systemmodell in die LM-Struktur wird die dritte Dimension im Wissensraum benötigt. Um die Übertragbarkeit eines Lösungsmusters auf weitere Problemstellungen zu ermöglichen, ist ein gewisser Generalisierungsgrad in der Lösungsbeschreibung erforderlich (vgl. Definition Kapitel 4.2.1). Demzufolge sind die bis dato vorliegenden Informationen in dem Systemmodell auf das zugrundeliegende Problem zugeschnitten. Bei der Aufbereitung der Informationen hin zu einem Lösungsmuster werden diese schrittweise generalisiert.

Einen einheitlichen Generalisierungsgrad von Lösungsmustern für den Systementwurf gibt es nicht. Je nach gewünschter Übertragbarkeit auf weitere Problemstellungen ist der Grad individuell zu definieren. Es bietet sich eine schrittweise Vorgehensweise an, bei der die einzelnen Zwischenstufen des Generalisierungsprozesses im Wissensraum relativ zueinander in Beziehung gesetzt werden. Die erforderlichen **Hilfsmittel** zur Durchführung dieser Phase 4 sind:

- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)
- Einheitliche Strukturierung von Lösungsmuster für den Systementwurf (vgl. Kapitel 4.3.1)
- Beziehungen zwischen den Lösungsmustern (vgl. Kapitel 4.3.3)
- Multidimensionaler Wissensraum (vgl. Kapitel 4.3.3.4)

Das **Resultat** dieser Phase „*Dokumentation von Lösungsmustern*“ sind Lösungsmuster für den Systementwurf. Diese sind gemäß der vorgegebenen Strukturierung spezifiziert und im Wissensraum entlang der charakteristischen Dimensionen in Beziehung zueinander gesetzt.

## 4.5 Vorgehensmodell für einen Lösungsmusterbasierten Systementwurf

Gegenstand dieses Kapitels ist ein Vorgehensmodell für einen Lösungsmusterbasierten Systementwurf fortgeschrittener mechatronischer Systeme. Im Sinne der VDI2206 wird die Herangehensweise eines funktionsorientierten Entwurfs aufgegriffen. Das Vorgehen wird in der frühen Phase einer Produktentwicklung, dem sog. Systementwurf, eingesetzt. Es konzentriert sich dabei in erster Linie auf die Wiederverwendung etablierten Lösungswissens. Das Vorgehensmodell beschreibt etablierte Entwurfsschritte unter Berücksichtigung der neuartigen z.T. multidisziplinären Lösungsmuster. Eine derart enge Verzahnung ist erforderlich, da der Einsatz von Lösungsmustern nicht losgelöst vom Entwicklungsgeschehen betrachtet werden kann. Werden beispielsweise für bestimmte Problemstellungen keine passenden Lösungsmuster gefunden, müssen neuartige Lösungen durch die Entwickler erarbeitet und mit den übrigen ausgewählten Lösungsmustern kombiniert werden.

Das Vorgehensmodell gliedert sich in fünf aufeinanderfolgende Phasen. Bild 4-21 verdeutlicht den Ablauf der Phasen, Aufgaben und Resultate.

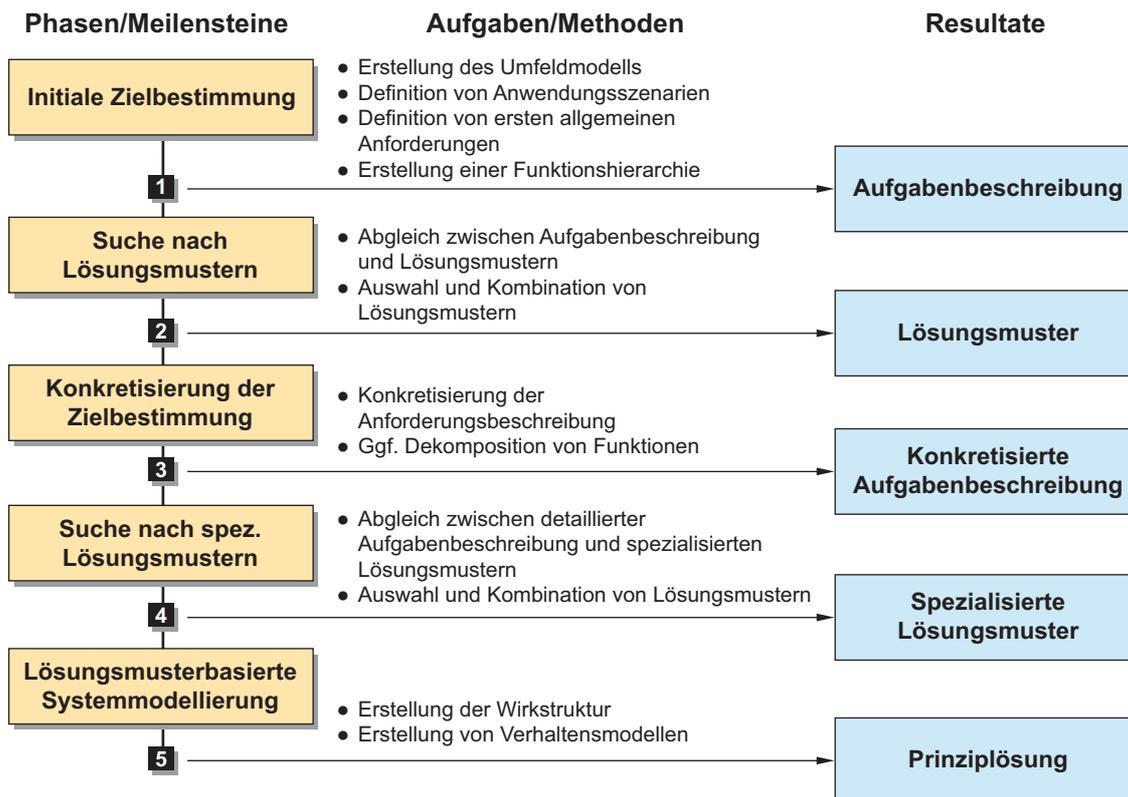


Bild 4-21: Vorgehensmodell für einen Lösungsmusterbasierten Systementwurf

Eingangsinformationen sind, in Anlehnung an die Strukturierung nach GAUSEMEIER, Erkenntnisse aus der strategischen Produktplanung (vgl. Kapitel 2.3.1). Ferner sind im Rahmen der Produktentwicklung bereits erste Architekturkonzepte des zu entwickelnden Systems beschrieben. Das Vorgehen für einen Lösungsmusterbasierten Systement-

wurf gliedert sich in insgesamt fünf Phasen. Es beginnt mit einer initialen Zielbestimmung und endet mit einer fachdisziplinübergreifenden Prinziplösung.

Innerhalb der einzelnen Phasen werden Hilfsmittel zur Durchführung der Aufgaben angewendet. Es handelt sich z.T. um bestehende Hilfsmittel aus dem Stand der Technik als auch um Neuentwicklungen des vorliegenden Instrumentariums. Bei der folgenden Erläuterung der einzelnen Phasen, Aufgaben und Resultate wird auf den Bezug zu den Hilfsmitteln näher eingegangen und entsprechend in die vorangegangenen Kapitel verwiesen.

#### 4.5.1 Phase 1: Initiale Zielbestimmung

Ausgehend von einer im Vorfeld entwickelten Produktstrategie und Architekturkonzeption erfolgt die erste Phase des Vorgehensmodells: Die initiale Zielbestimmung. Die Beschreibung der Phase ist angelehnt an die wiederkehrenden Tätigkeiten im Rahmen einer jeden Produktentwicklung: Zielbestimmung, Synthese, Analyse. Ziel ist eine erste Grundlage, auf der ein interdisziplinäres Entwicklerteam nach potentiellen Lösungsmustern suchen kann. Die Phase klärt folglich die Aufgabenbeschreibung für die weiteren Entwicklungsschritte. Das Abstraktionsniveau ist zu diesem Zeitpunkt noch sehr hoch. Grundgedanke ist es, den Entwicklern auf Basis einer ersten vage formulierten Aufgabe ein möglichst facettenreiches Spektrum an Lösungsmustern anzubieten. Andernfalls ist die bereits skizzierte Gefahr zu hoch, sich in zu diesem Zeitpunkt der Entwicklung in zu spezifischen Details zu verlieren.

Zunächst wird das Umfeld des zu entwickelnden Systems analysiert. Es folgt die Definition von Anwendungsszenarien. Auf Basis der erzielten Erkenntnisse und den in der strategischen Produktplanung bereits erarbeiteten Markt- und Kundenanforderungen werden Anforderungen an das technische System spezifiziert. Hierzu werden die i.d.R. sehr abstrakt gehaltenen Anforderungen der strategischen Produktplanung in technische Anforderungen übersetzt. Die Dokumentation geschieht in Form einer Anforderungsliste, gemäß der Strukturierung nach PAHL/BEITZ. Anschließend wird eine Funktionshierarchie abgeleitet. Im Gegensatz zur Funktionsstruktur beinhaltet die Hierarchie keine Aussagen über die Zusammenhänge zwischen den Funktionen, was aus folgendem Grund vorteilhaft ist.

Es ist im Verlaufe der ersten Phase des Vorgehensmodells darauf zu achten, die Anforderungs- und Funktionsdokumentation so lösungsneutral wie möglich auszuführen. Dies ist Grundvoraussetzung zur Ausschöpfung des vollen Innovationspotentials in der anschließenden Phase *Auswahl von Lösungsmustern*. Gelingt dies nicht, legen sich Entwickler unbewusst auf ihnen bekannte Lösungsmuster fest.

Zur Durchführung der ersten Phase werden die entsprechenden Partialmodelle und Konstrukte der *Spezifikationstechnik CONSENS* angewendet. Es wird in erster Linie davon ausgegangen, dass es sich bei dem Entwicklungsgegenstand um eine Neuentwicklung

handelt. Im Falle einer Anpassungsentwicklung ist sicherzustellen, dass bereits vorliegende Entwicklungsdokumente auf das erforderliche Abstraktionsniveau gebracht werden. Dies kann dann z.B. die Umfeld-, Anforderungs- und Funktionsbeschreibung betreffen. Die Inhalte sind auf den Kern des Entwicklungsgegenstandes zu reduzieren. Je nach Komplexität und Anpassungsfähigkeit des umliegenden Gesamtsystems sind bereits vorhandene Schnittstellen im Umfeldmodell detailliert zu spezifizieren. Ein Beispiel ist die Modifikation einer neuen Scheinwerfergeneration in einem Automobil bei bereits vorgegebenen Umgebungsbedingungen (Bauraum, elektr. Versorgung etc.) Zusammenfassend werden in der Phase initiale Zielbestimmung folgende **Hilfsmittel** eingesetzt:

- Kategorien zur Anforderungsspezifikation nach PAHL/BEITZ [PBF+07]
- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)

Das **Resultat** der Phase „*initiale Zielbestimmung*“ ist eine erste Aufgabenbeschreibung. Diese umfasst im Kern eine Anforderungs- und Funktionsbeschreibung auf einem sehr hohen Abstraktionsniveau.

#### 4.5.2 Phase 2: Suche nach Lösungsmustern

Ziel einer ersten Suche und Auswahl von Lösungsmustern ist die grobe Einschränkung des zur Verfügung stehenden Suchraums. Auf Basis der Funktionshierarchie lassen sich zwei grundlegende Suchstrategien – Top-Down oder Bottom-Up – durchführen. Beim Top-Down-Ansatz startet die Suche bei der Hauptfunktion. Wird keine Lösung gefunden, werden die einzelnen Ebenen der Hierarchie nacheinander durchlaufen. Zielsetzung ist, möglichst schnell Lösungsmuster mit einem hohen Aggregationsgrad für die vorliegende Entwicklungsaufgabe zu identifizieren. Dies vermeidet zeit- und kostenaufwändige Neuentwicklungen für Systembestandteile, die bereits etabliert sind. Die Bottom-Up-Suchstrategie startet auf der untersten Ebene der Hierarchie und fokussiert die Lösungsalternativen für die einzelnen Teilfunktionen. Dies bietet sich beispielsweise an, wenn für ein bestehendes System eine alternative Umsetzung betrachtet werden soll. Die gefundenen Lösungen ermöglichen in diesem Fall Potentiale für Optimierungen.

In dieser zweiten Phase des Vorgehensmodells navigiert das Entwicklerteam auf der Suche nach passenden Lösungsmustern durch den multidimensionalen Wissensraum. Die grundsätzliche Richtung ist dabei immer vom *generalisierten* zum *spezialisierten*. Je nach Strategie kann sowohl mit der Suche nach *elementaren* als auch nach *komplexen* Lösungsmustern begonnen werden. Bei der Suche werden die in der ersten Phase spezifizierten Anforderungen und Funktionen mit der Problembeschreibung in den Lösungsmustern abgeglichen. Aufgrund der sehr lösungsneutralen Funktionsbeschreibung ist das Suchergebnis zu Beginn sehr vielseitig. Die Anforderungen dienen einer ersten Einschränkung. Der hohe Generalisierungsgrad in den Lösungsmustern unterstützt die Entwickler in ihren kreativen Entwurfsprozessen. Ziel ist es, alle Funktionen mit poten-

tiellen Lösungsmustern jedweden Generalisierungsgrads zu erfüllen. Dabei kann ein Lösungsmuster 1-n Funktionen erfüllen, dies hängt von seiner Komplexität ab. Für Funktionen, für die kein Lösungsmuster gefunden wird, sind neuartige Konzepte durch die Entwickler zu erarbeiten. Nach Auswahl der Lösungsmuster werden diese in einem morphologischen Kasten gesammelt und auf Kombinierbarkeit geprüft. Dies geschieht auf der Basis von Entwicklerexpertise.

Die nachfolgende Aufzählung fasst die benötigten **Hilfsmittel** zusammen:

- Lösungsmuster für den Systementwurf (vgl. Kapitel 4.3.1 und 4.3.2)
- Beziehungen zwischen den Lösungsmustern (vgl. Kapitel 4.3.3)
- Multidimensionaler Wissensraum (vgl. Kapitel 4.3.3.4)
- Morphologischer Kasten [PBF+07]

Das **Resultat** dieser „Suche nach Lösungsmustern“ ist ein morphologischer Kasten, der potentielle Lösungsmuster unterschiedlicher Spezialisierungsniveaus (i.d.R. stark generalisiert) beinhaltet und diesen den zuvor gestellten Funktionen zeilenweise gegenüberstellt.

### 4.5.3 Phase 3: Konkretisierung der Zielbestimmung

Die Auswahl und Kombination von Lösungsmustern in Phase 2 führt in der Regel aufgrund der bereitgestellten Merkmale zu neuen Erkenntnissen bei den Entwicklern. Die Merkmale, die in den ausgewählten Lösungsmustern zugeordnet sind, erfordern eine Konkretisierung der zuvor in Phase 1 spezifizierten Zielbestimmung. Zum einen kann es erforderlich sein, existierende Anforderungen zu konkretisieren oder Restriktionen zu analysieren. Zum anderen können durch die Merkmale neue Anforderungen hinzukommen. Jedes ausgewählte Lösungsmuster liefert dem Entwickler folglich weitere Merkmale, die in die Anforderungsliste für das Gesamtsystem aufgenommen werden müssen. Ein Beispiel, das diesen Zusammenhang verdeutlicht, ist das Lösungsmuster „elektrischer Antrieb“. Dieses wird z.B. auf Basis der Funktion *Welle antreiben* als geeignetes Lösungsmuster in Phase 2 ausgewählt. Es erfolgt somit zwangsläufig die Festlegung auf die Verwendung von elektrischer Energie anstelle z.B. pneumatischer Energie. Folglich muss die Anforderungsliste in dieser Phase um die Notwendigkeit einer *Versorgungsspannung* ergänzt werden. Dies wiederum bedingt weitere Anforderungen, etwa Schutz vor Umgebungseinflüssen wie z.B. Feuchtigkeit etc.

Ähnlich wie mit den Anforderungen verhält es sich mit der Funktionshierarchie. Die funktionale Beschreibung ist entscheidend für die Problemlösung sowie für ein einheitliches Verständnis unter den Entwicklern. Mit einer Konkretisierung der Anforderungen geht z.T. eine Verfeinerung der Funktionshierarchie einher. Durch eine weitere Hierarchisierung der zu erfüllenden Funktionen kann das gewünschte Systemverhalten disziplinübergreifend konkretisiert werden. So hat die Wahl eines Lösungsmusters Einfluss auf die Anzahl der zu erfüllenden Subfunktionen und bedingt eine erneute Suche. An

dieser Stelle sei das Beispiel des „elektrischen Antriebs“ zur Verdeutlichung noch einmal aufgegriffen. Die Funktion, auf Basis derer das Lösungsmuster ausgewählt wurde, ist „Welle antreiben“. Potentielle Subfunktionen, die nach Auswahl des Lösungsmusters Verfeinerungsschritte der Funktionshierarchie darstellen, sind z.B. „elektrische Energie in mechanische Energie wandeln“ und „Drehmoment abstützen“. Die letztgenannte Funktion hat wiederum Auswirkungen auf weitere ausgewählte Lösungsmuster bzw. angrenzende Komponenten im Gesamtsystem.

Wie bereits erwähnt, handelt es sich bei der Darstellung eines Phasen-Meilenstein-Diagramms um eine idealtypische Darstellung. An dieser Stelle ist darauf hinzuweisen, dass insbesondere die Phasen drei und vier iterativ durchlaufen werden. Die für diese Phase benötigten **Hilfsmittel** sind:

- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)
- Informationen aus den Lösungsmustern (vgl. Kapitel 4.3.1)

Als **Resultat** der Phase „*Konkretisierung der Zielbestimmung*“ liegt eine Aufgabenbeschreibung vor, bestehend aus einer konkretisierten Anforderungsliste und Funktionshierarchie.

#### 4.5.4 Phase 4: Suche nach spezialisierten Lösungsmustern

Die Suche nach spezialisierten Lösungsmustern verläuft prinzipiell analog zur ersten Suche in Phase 2. Ausgangspunkt ist die in Phase 3 spezifizierte detaillierte Aufgabenbeschreibung. Als weitere Information stehen den Entwicklern die bereits ausgewählten Lösungsmuster zu Verfügung. Diese sind zudem Startpunkt für die Suche nach spezialisierten Lösungsmustern.

Im Gegensatz zu untersuchten Ansätzen aus dem Stand der Technik (vgl. Kapitel 3.3.2 und 3.3.3), bietet die geführte Suche einen neuen Ansatz. Die Entwickler werden ab dem Zeitpunkt erster abstrakter Vorstellungen vom System bei der Einschränkung des Suchraums so unterstützt, dass Ihnen zu jedem Zeitpunkt die größtmögliche Anzahl an alternativen Lösungsmustern zur Verfügung steht. Ausgehend von den ausgewählten Lösungsmustern navigieren die Entwickler vom *generalisierten* zum *spezialisierten*, um das System im Entwurf weiter zu *konkretisieren*.

Zur Verdeutlichung der geschilderten Zusammenhänge wird an dieser Stelle erneut das Lösungsmuster „elektrischer Antrieb“ verwendet. Ein spezialisiertes Lösungsmuster, das diesem Lösungsmuster oder der Lösungsmusterklasse angehört, ist z.B. das Lösungsmuster „Servoantrieb“. Dieses wiederum steht mit den spezialisierteren Lösungsmustern „Servo-Synchronantrieb“ und „Servo-Asynchronantrieb“ in Beziehung. Durch die Definition und die Abbildung derartiger wechselseitiger Beziehungen ist es den Entwicklern möglich, sukzessive den Suchraum einzuschränken oder umgekehrt, bei Ausschluss von Lösungsmustern zu erweitern. Folgende Auflistung gibt einen Überblick über die zu verwendenden **Hilfsmittel**:

- Lösungsmuster für den Systementwurf (vgl. Kapitel 4.3.1 und 4.3.2)
- Beziehungen zwischen den Lösungsmustern (vgl. Kapitel 4.3.3)
- Multidimensionaler Wissensraum (vgl. Kapitel 4.3.3.4)
- Morphologischer Kasten [PBF+07]

Das **Resultat** der vierten Phase ist eine Auswahl und Kombination spezialisierter Lösungsmuster. Erstrebenswert ist ein solcher Spezialisierungsgrad, der unmittelbar im Anschluss an den Systementwurf eine Überführung der Informationen in aussagekräftige formale Simulationen ermöglicht.

#### 4.5.5 Phase 5: Lösungsmusterbasierte Systemmodellierung

Die abschließende Phase des Vorgehensmodells ist die lösungsmusterbasierte Systemmodellierung. Die Kombination der Lösungsmuster aus Phase 4 markiert den Startpunkt für den Aufbau der Struktur- und Verhaltensmodelle auf fachdisziplinübergreifender Gesamtsystemebene. Jedes ausgewählte Lösungsmuster beinhaltet für die jeweilige Problemstellung generalisierte Informationen, die den Entwickler bei der Synthese unterstützen. Dies führt bei der Modellierung der Partialmodelle zu einer signifikanten Zeitersparnis.

Bei der Erstellung der Wirkstruktur werden für jedes Lösungsmuster die bereitgestellten Systemelemente und internen Beziehungen innerhalb eines Musters aus dem Wissensraum abgerufen. In der Regel ist es erforderlich, die bereitgestellten Lösungsmuster auf die jeweils spezifische Problemstellung der Entwicklungsaufgabe zu adaptieren. Dies geschieht sowohl über die Modifikation der Systemelemente als auch über die Modifikation der Flussbeziehungen. Ferner ist es Aufgabe des Entwicklers, die Systemelemente der Lösungsmuster zu einem in sich verträglichen Gesamtsystem zu kombinieren.

Gemäß der Spezifikationstechnik und Methode CONSENS sind in Ergänzung zur Wirkstruktur die in Beziehungen stehenden Partialmodelle zu erarbeiten. Dies geschieht analog zur Modellierung der Wirkstruktur mit den generalisierten Verhaltensmodellen (z.B. Verhalten-Zustände) der Lösungsmuster. Die Erstellung der Partialmodelle geschieht im Wechselspiel. Eine Besonderheit stellt das im Rahmen dieser Arbeit entwickelte Partialmodell Verhalten-Sequenz dar. Es bildet eine Ausnahme bei der Verbindung der Systemelemente in der Wirkstruktur durch die Entwickler. So beinhalten die entwickelten Lösungsmuster der Softwaretechnik detaillierte Informationen zu einer bidirektionalen Kommunikation zwischen mindestens zwei Systemelementen. Diese Anteile des Verhaltens werden unmittelbar in die Wirkstruktur integriert.

Die Hardware, auf der die Software später ausgeführt wird, ist nicht Teil der Modellierung. Die erforderlichen Hardwarekomponenten werden erst während der nachfolgenden Konkretisierung und tatsächlichen Implementierung ausgewählt. Die erforderlichen **Hilfsmittel** für die letzte Phase sind:

- Spezifikationstechnik CONSENS (vgl. Kapitel 3.2.1.1)

- Erweiterung von CONSENS um Verhalten-Sequenz (vgl. Kapitel 4.3.2.1)
- Informationen aus den Lösungsmustern (vgl. Kapitel 4.3.1)

Am Ende der Phase „*Lösungsmusterbasierte Systemmodellierung*“ liegt als abschließendes **Resultat** eine fachdisziplinübergreifende Spezifikation des fortgeschrittenen mechatronischen Systems vor. Falls es ist im Rahmen des Entwurfs, aufgrund nicht vorhandener Lösungsmuster, zu Neuentwicklungen gekommen ist, sollten diese im Anschluss an die gesamte Entwicklung gemäß des Vorgehens aus Kapitel 4.4 in den Wissensraum aufgenommen werden.



## 5 Anwendung und Bewertung

In diesem Kapitel wird das *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* anhand ausgewählter Anwendungsbeispiele validiert. Kern des Instrumentariums ist die erarbeitete Systematik, die auf dem Paradigma der Lösungsmuster in der Produktentstehung fußt und aus der Kombination der beiden Vorgehensmodelle zur Identifizierung von Lösungsmustern und für den lösungsmusterbasierten Entwurf resultiert. Die Vorgehensmodelle greifen die Ergebnisse zur Dokumentation des Lösungswissens aus Kapitel 4.3 auf. Aus diesem Grund wird zunächst in Kapitel 5.1 das Vorgehensmodell zur Identifizierung von Lösungsmustern anhand eines „Separators“ validiert. Kapitel 5.2 beinhaltet die Validierung des Vorgehensmodells für einen lösungsmusterbasierten Systementwurf anhand des Beispiels „Kooperierende Deltaroboter“. Kapitel 5.3 bildet den Abschluss des Kapitels. Es werden die erarbeiteten und validierten Inhalte des Instrumentariums anhand der Anforderungen aus der Problemanalyse (Kapitel 2.6) bewertet.

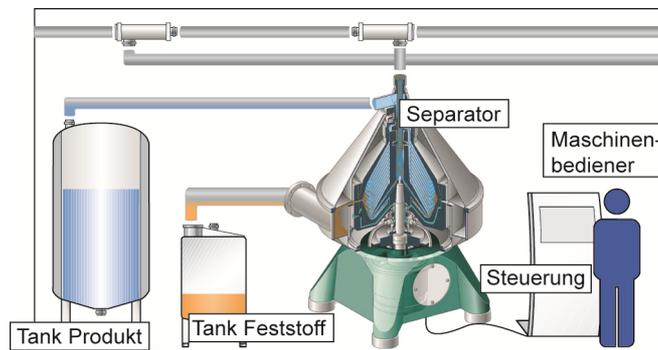
### 5.1 Anwendung des Vorgehensmodells zur Identifizierung von Lösungsmustern für den Systementwurf am Beispiel „Separator“

Die Validierung des in Kapitel 4.4 vorgestellten *Vorgehensmodells zur Identifizierung von Lösungsmustern für den Systementwurf* erfolgt beispielhaft an einem Separator. Aufgabe eines Separators ist die mechanische Trennung eines Rohprodukts. Ein Separator ist in der Regel Teil eines übergeordneten Produktionsprozesses (vgl. Bild 5-1). Die Arbeitsweise beruht auf der Zentrifugalkraft und den unterschiedlichen Massenträgheiten innerhalb des zugeführten Rohprodukts. Partikel mit höherer Dichte werden aufgrund der höheren Trägheit vom Drehzentrum weg befördert. Diese verdrängen hierbei denjenigen Anteil des Rohprodukts mit niedrigerer Dichte, der hierdurch auf einem kleineren Durchmesser bleibt bzw. zum Drehzentrum hin befördert wird. Mittels Zentrifugalkraft trennen Separatoren u.a. Feststoffe aus Flüssigkeiten ab. Ebenso effektiv trennen sie Flüssigkeitsgemische, bei gleichzeitiger Sedimentation von Feststoffen<sup>65</sup>. Aufgrund des breiten Anwendungsfeldes existieren eine Vielzahl von Separatoren-Ausführungen: Vollmantelseparatoren, Kammerseparatoren, selbstentleerende Tellerseparatoren etc. Die Anwendungsgebiete reichen dabei von Trennprozessen in der chemischen und pharmazeutischen Industrie, der Öl- und Fettgewinnung, bis hin zur Herstellung von Molkereiprodukten, Bier, Wein, Frucht- und Gemüsesäften oder der Verarbeitung von Mineralöl und Mineralölprodukten. Im Rahmen der vorliegenden Arbeit wird ein selbstentleerender Tellerseparator für fest-flüssig und/oder flüssig-flüssig Trennungen betrachtet. Die Entleerung des Feststoffes – Austrag aus der Trommel in den Fest-

---

<sup>65</sup>Für detaillierte Informationen sei an dieser Stelle auf [Sta04] verwiesen.

stofftank – erfolgt dabei zyklisch. Die Abführung der flüssigen Phase in den Produkt-tank ist hingegen ein kontinuierlicher Prozess.



*Bild 5-1: Selbstentleerender Tellerseparator als Teil eines übergeordneten Produktionsprozesses*

Am ausgewählten Anwendungsbeispiel wird gezeigt, wie das System ausgehend von existierenden Entwicklungsdokumenten und personenbezogener Expertise retrospektiv analysiert wird. Ziel ist es, Lösungsmuster zu identifizieren, zu definieren und im Wissensraum gegenüberzustellen.

### 5.1.1 Phase 1: Informationsakquisition

Im Rahmen der Informationsakquisition wird das Fundament für die Identifizierung von Lösungsmustern geschaffen. Im Mittelpunkt dieser Phase steht das Aufstellen eines Systemmodells des zu analysierenden Produkts. Hierzu werden vorhandene Entwicklungsdokumente zusammengetragen und gesichtet. Ziel ist die Zusammenführung aller existierenden Informationen, die bereits in expliziter Form vorliegen. Die zugrundeliegenden Informationen im gewählten Anwendungsbeispiel zeigt Bild 5-2.

Im Wesentlichen liegen Informationen aus den Fachbereichen Maschinenbau (**Mechanik**) und Elektrotechnik (**Automatisierung**) vor. Aus Konstruktionsunterlagen gehen Erkenntnisse über die physische Baustruktur hervor. Ferner existiert ein mechanischer Baukasten, dessen Gliederung in Bild 5-2 schematisch eingeführt wird. So liegen beispielsweise unterschiedliche Ausführungen, bezogen auf geometrische Abmaße, des *Zulaufs*, der *Trommel* (Kombination von *Trommeloberteil und -unterteil*), oder auch des *Antriebs* mit standardisierten Wirkflächen vor. Der Baukasten ist notwendig, um die Variantenvielfalt, begründet in der hohen Anzahl unterschiedlicher Applikationen (Rohöl, Lebensmittel etc.), wirtschaftlich bedienen zu können. Demgegenüber steht ein feingliedriger Funktionsplan aus der Automatisierung. Dieser beinhaltet sämtliche Programmabläufe in der Steuerung mit entsprechenden Schnittstellen zum physikalischen System (Sensorik und Aktorik). Wirkzusammenhänge über die Fachdisziplinen hinaus liegen zu diesem Zeitpunkt noch in keinem gemeinsamen Modell vor. Ein fachdisziplinübergreifendes Systemmodell ist daher für das weitere Vorgehen essentiell.

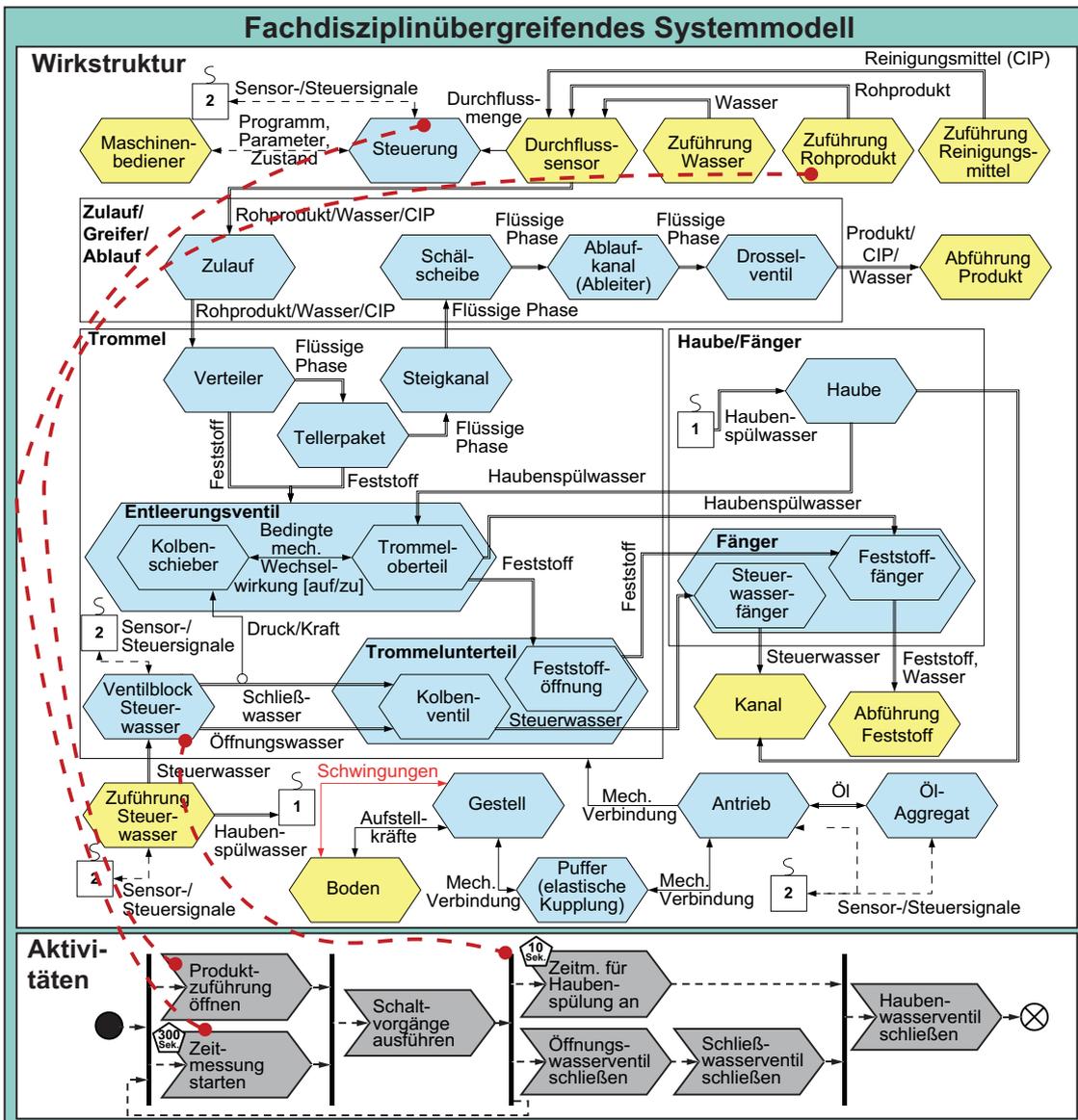
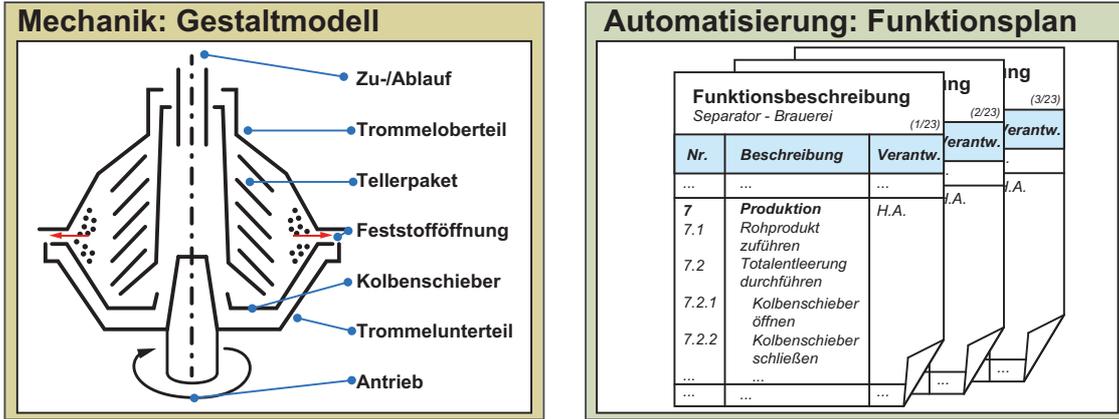


Bild 5-2: Erstellung eines Systemmodells auf Basis disziplinspezifischer Dokumente

Bild 5-2 verdeutlicht das retrospektiv aufgestellte Systemmodell eines Separators, das sich in die beiden Partialmodelle Wirkstruktur und Verhalten-Aktivitäten unterteilt. Der Abstraktionsgrad der Wirkstruktur und der Umfang der Aktivitätsfolge sind entsprechend des anvisierten Ziels – der Identifizierung von multidisziplinären Lösungsmustern – gewählt. Das vollständige Systemmodell ist deutlich umfangreicher.

Die **Wirkstruktur** zeigt das Zusammenwirken einzelner Systemelemente untereinander sowie mit notwendigen Umfeldelementen. Da ein Separator Teil eines übergeordneten Produktionsprozesses ist, orientiert sich die nachfolgende Beschreibung der Wirkungsweise am Produktfluss durch das System. Über eine „Zuführung“ im Umfeld gelangt das zu verarbeitende „Rohprodukt“ (je nach Betriebsmodi auch „Wasser“ oder „Reinigungsmittel“) über den „Zulauf“ in das System. Die Menge wird geregelt über einen im Zulauf befindlichen „Durchflusssensor“. In der Trommel wird das zugeführte Medium mit Hilfe eines „Verteilers“ beschleunigt. Für die Beschleunigung der gesamten Trommel sorgt ein „elektrischer Antrieb“. Die Separation des Rohprodukts erfolgt größtenteils im „Tellerpaket“. Der Feststoff sammelt sich sukzessive an der Trommelinnenwand an, wohingegen die flüssige Phase kontinuierlich über einen „Steigkanal“ mit anschließender „Schälscheibe“ abgeführt wird. Über einen speziellen „Entleerungsmechanismus“ wird der abgelagerte Feststoff zyklisch entleert. Mit Hilfe einer Ansteuerung von Wasserventilen, lässt sich ein sog. „Kolbenschieber“ im System absenken und anheben. Dieser „Kolbenschieber“ gibt spezielle Öffnungen in der Trommelwand frei, so dass der Feststoff in den umliegenden „Feststofffänger“ gelangen kann.

Das Partialmodell **Verhalten-Aktivitäten** in Bild 5-2 beschreibt diesen komplexen Ablauf und verdeutlicht das Interagieren zwischen mechanischen Elementen im System und Programmabläufen in der Steuerung. Neben dem bereits beschriebenen Produktfluss im System stellt das Partialmodell die zeitliche Reihenfolge von Schaltungsvorgängen dar, die für die Feststoffentleerung notwendig sind. Zunächst wird die „Zeitmessung“ bis zur nächsten Entleerung gestartet. Die Größe der Zykluszeit wird durch den Maschinenbediener vor Produktionsbeginn festgelegt. Es folgt die Ansteuerung einzelner Ventile, die sich im Systemelement „Ventilblock Steuerwasser“ (s. Wirkstruktur) befinden.

### 5.1.2 Phase 2: Systemanalyse

Aufbauend auf dem fachdisziplinübergreifenden Systemmodell aus Phase 1 folgt eine Systemanalyse. Im Rahmen der Analyse wird das Ziel verfolgt, initial Lösungsmusterkandidaten zu identifizieren. Einen ersten Anhaltspunkt für derartige Kandidaten bieten mechatronische Funktionseinheiten. Vor diesem Hintergrund wird zunächst eine Funktionshierarchie retrospektiv aufgestellt. Das bedeutet, es werden je Systemelement oder je logischer Gruppe von Systemelementen zugehörige Funktionen identifiziert und in einer Hierarchie aggregiert. Einen Auszug der erstellten Funktionshierarchie zeigt Bild 5-3.

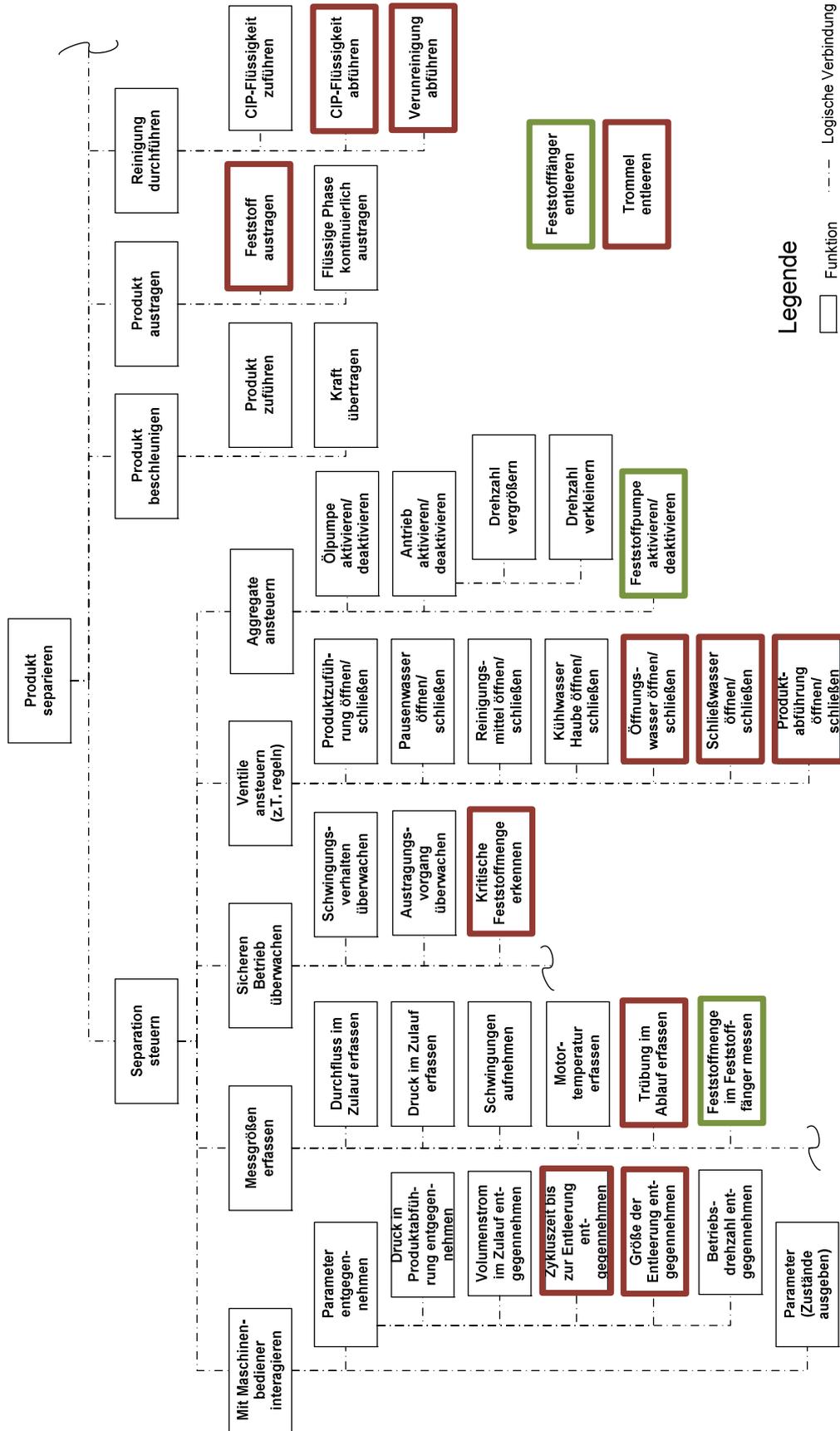


Bild 5-3: Funktionshierarchie eines Separators (Auszug)

Gesamtfunktion eines Separators ist „*Produkt separieren*“. Diese unterteilt sich u.a. in die Teilfunktionen „*Separation steuern*“, „*Produkt beschleunigen*“, „*Produkt austragen*“ und „*Reinigung durchführen*“. Anhand dieser Gliederung lässt sich auch die Zuordnung in die Wirkstruktur erkennen. So unterteilt sich die Funktionalität der Steuerung (s. Wirkstruktur, Bild 5-2) in Subfunktionen wie beispielsweise „*Mit Maschinenbediener interagieren*“, „*Sicheren Betrieb überwachen*“ oder auch „*Ventile / Aggregate ansteuern*“. Subfunktionen von den Austragemechanismen für die unterschiedlichen Medien sind u.a. „*Feststoff austragen*“ oder „*Flüssige Phase kontinuierlich austragen*“. Es wird deutlich, dass sich die gemeinsam mit den Entwicklern aufgenommene Funktionshierarchie entlang der Entwicklungsschwerpunkte der Fachdisziplinen gliedert. Die Mitarbeiter der Automatisierung verantworten die Subfunktionen von „*Separation steuern*“, die Konstrukteure die Subfunktionen von „*Produkt beschleunigen*“ oder „*Produkt aufheben*“. Dieses Denken und Handeln in unterschiedlichen Fachdisziplinen gilt es aufzuheben. Im Hinblick auf multidisziplinäre Lösungsmuster ist es essentiell, Teillösungen aus den beteiligten Fachdisziplinen zu vereinen und geeignet zu dokumentieren.

Vor diesem Hintergrund findet in einem weiteren Schritt eine **Analyse der Subfunktionen** statt. Hierzu wird die aufgestellte Hierarchisierung im Hinblick auf übergeordnete, i.d.R. **mechatronische Funktionseinheiten** untersucht und neustrukturiert. Es wird im Grunde genommen eine neue Sicht auf die Funktionshierarchie und die bereits erarbeitete Wirkstruktur entwickelt. Zwei ausgewählte Beispiele im Gesamtkontext eines Separators sind die übergeordneten Funktionen „*Feststofffänger entleeren*“ und „*Trommel entleeren*“ (vgl. Bild 5-3, grüne und rote Markierung). Die erste Funktion erstreckt sich über mehrere Systemelemente in der Wirkstruktur, die vor allem räumlich voneinander getrennt sind. Die Subfunktion „*Feststoffmenge im Feststofffänger messen*“ und „*Feststoffpumpe aktivieren*“ sind in der Funktionshierarchie der Funktion „*Separation steuern*“ und folglich der Automatisierung zugeordnet. Zweifelsohne existiert eine Vielzahl von Schnittstellen zur mechanischen Konstruktion, insbesondere im Bereich des Feststofffängers. Es ist daher in Abstimmung mit den Experten im weiteren Verlauf des Vorgehens zu prüfen, ob die Definition eines Lösungsmusters an dieser Stelle sinnvoll erscheint.

Ein weiteres Beispiel ist die Funktion „*Trommel entleeren*“. Diese ist von der Komplexität her deutlich umfangreicher als das zuvor beschriebene Beispiel. Involvierte Subfunktionen erstrecken sich über die gesamte Funktionshierarchie. Hierzu zählen u.a. „*Trübung im Ablauf erkennen*“, „*Kritische Feststoffmenge erkennen*“, „*Feststoff austragen*“ sowie „*CIP-Flüssigkeit und Verunreinigungen abführen*“. Aufgrund der Involvierung aller beteiligten Bereiche und des Stellenwerts der Funktionen für den sicheren Betrieb eines Separators folgt eine detaillierte Analyse im Hinblick auf potentielle Lösungsmusterkandidaten. Hierfür werden die zugehörigen Funktionen extrahiert, neustrukturiert und die Verbindungen in der Wirkstruktur visualisiert. Die beschriebenen



Farblich rot hervorgehoben sind die Kernfunktionen einer Entleerung. Diese sind erforderlich um eine Entleerung durchführen zu können. Da diese auf unterschiedliche Art und Weise initiiert werden kann, gibt es drei weitere Subfunktionen. Diese sind ebenfalls farblich hervorgehoben. Es handelt sich um die Funktionen „*Trübung im Ablauf erfassen*“ (grün), „*Kritische Feststoffmenge erkennen*“ (orange) und „*Zykluszeit bis zur Entleerung entgegennehmen*“. Die umsetzenden Systemelemente sind zur Visualisierung der Querbeziehungen ebenfalls farblich hervorgehoben.

Eine Besonderheit bei der Identifizierung von Lösungsmusterkandidaten liegt in der Modellierung der Softwarearchitektur. Im Vergleich zum Systemmodell aus Phase 1 ist es zwingend erforderlich, die Architektur der Software mindestens auf der ersten Subsystemebene nach der Gesamtsteuerung mit zu modellieren. Andernfalls lassen sich die Teillösungen aus den beteiligten Disziplinen nicht in der erforderlichen Detailtiefe spezifizieren, um die Schnittstellen transparent abzubilden. Die angesprochene Erweiterung ist in Bild 5-4 zu sehen. Auf der untersten Ebene der Steuerung sind Systemelemente lokalisiert, die den Durchgriff der Steuerung auf die Hardware umsetzen. In der markierten Funktionseinheit ist dies z.B. die „*Ventilverwaltung*“. Es handelt sich um eine Gruppe von Softwarekomponenten, die sämtliche Ventile am Separator mit der Steuerung und dem HMI verbinden. Auf einer darüber liegenden Ebene wird die Logik modelliert, die für die Steuerung des Separators erforderlich ist. Für die Funktion „*Trommel entleeren*“ handelt es sich um das Systemelement „*Trommelentleerung*“, welches wiederum gemäß der umfangreichen Funktionalität in die Subsystemelemente „*Zeitüberwachung*“, „*Trübungsüberwachung*“ und „*Feststoffüberwachung*“ unterteilt ist. Das Zusammenspiel der aneinandergereihten Teillösungen erstreckt sich folglich über alle Bereiche der Referenzarchitektur bzw. Grundstruktur mechatronischer Systeme (vgl. Kap. 2.2.1): beginnend in der Logik über die untere Interface-Ebene hin zu den Ventilen, von dort auf mechanische Komponenten wie den „*Entleerungsmechanismus*“, weiter über Sensoren wie „*Trübungssensor*“ oder „*Schwimminitiator*“ zurück über die Interface-Ebene in die Logik.

Das zeitliche Verhalten der Systemelemente im Zusammenspiel ist im Aktivitätendiagramm in Bild 5-5 dargestellt. Es komplettiert die Systembeschreibung aus Sicht der Automatisierung. Den Startpunkt bildet die parallel ablaufende „*Feststoffüberwachung*“, „*Trübungsüberwachung*“ und die Aktivität „*Zeitmessung starten*“. parallele Abfragen, die die „*Initiierung der Entleerung*“ auslösen. Im Anschluss an die Entleerungsinitiierung werden in einer fest vorgegebenen Reihenfolge spezielle Ventile angesteuert, die für die Öffnung der Trommel und folglich für die eigentliche Entleerung sorgen.

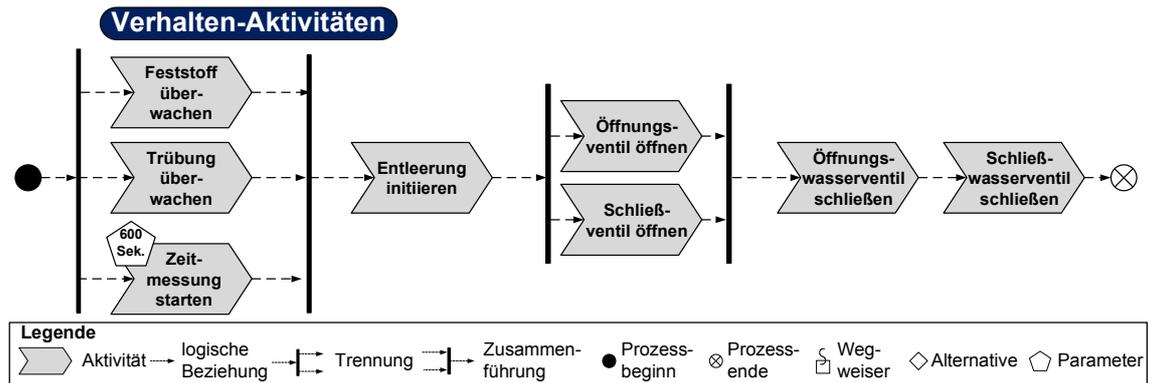


Bild 5-5: Verhalten-Aktivitäten der Funktionseinheit „Trommelentleerung“

In dem gewählten Beispiel der Trommelentleerung existiert eine Vielzahl von fachdisziplinspezifischen „Teillösungen“. In einem abschließenden Schritt werden diese im Hinblick auf ihre Aggregation analysiert. Um die vorhandene Komplexität zusammenhängender Teillösungen zu reduzieren, ist eine Aufteilung in elementare Teillösungen erforderlich. Der Fokus liegt dabei, wenn immer sinnvoll, auf der Wahrung der Interdisziplinarität. Die resultierende Aggregation des gewählten Lösungsmusterkandidaten „Trommelentleerung“ verdeutlicht Bild 5-6. So setzt sich der erwähnte Kandidat aus den elementaren Kandidaten „Trübungsüberwachung“, „Feststoffüberwachung“ und „Aktion befehlen“<sup>66</sup> zusammen. In der Feststoffüberwachung ist wiederum der Lösungsmusterkandidat „Feststoffdetektion“ enthalten. Hierbei handelt es sich um eine spezialisierte mechanische Lösung, bei der eine potentielle Adaption an weitere Problemstellungen sinnvoll erscheint. Die extrahierten Kandidaten gilt es jedoch in einem anschließenden Schritt mit den Experten der Fachdisziplinen gemeinsam zu diskutieren.



Bild 5-6: Aggregation des multidisziplinären Lösungsmusterkandidaten „Trommelentleerung“

### 5.1.3 Phase 3: Interaktive Analyse

Die interaktive Analyse ist beim Vorgehen zur Identifizierung von Lösungsmustern entscheidend, da ein Großteil der Expertise implizit in den Köpfen der Entwickler vor-

<sup>66</sup>Bei dem Kandidaten „Aktion befehlen“ handelt es sich um ein bereits im Rahmen dieser Arbeit vorgestelltes Lösungsmuster. Dieses ist bereits in dem Lösungsmuster „Zusammenarbeit synchronisieren“ in Kapitel 4.3.2.1 eingeführt.

liegt. Daher sind die Lösungsmusterkandidaten in mindestens einem Review mit den Entwicklern zu besprechen und ggf. zu detaillieren. Es ist durchaus denkbar, dass Kandidaten nicht in Lösungsmuster überführt werden, wenn diese z.B. entgegen der Annahme keine eigene Problem-Lösungs-Paarung repräsentieren.

Im Hinblick auf eine zielgerichtete Analyse der extrahierten Kandidaten wird das Systemmodell auf die wesentlichen Systemelemente und Funktionen reduziert. Lediglich die Systemelemente, die unmittelbar mit der potentiellen Mustergruppe von Systemelementen interagieren, werden weiterhin aufgeführt. Die auf die wesentlichen Informationen reduzierten Partialmodelle Funktionen und Wirkstruktur des Lösungsmusterkandidaten „Trommelentleerung“ zeigt Bild 5-7.

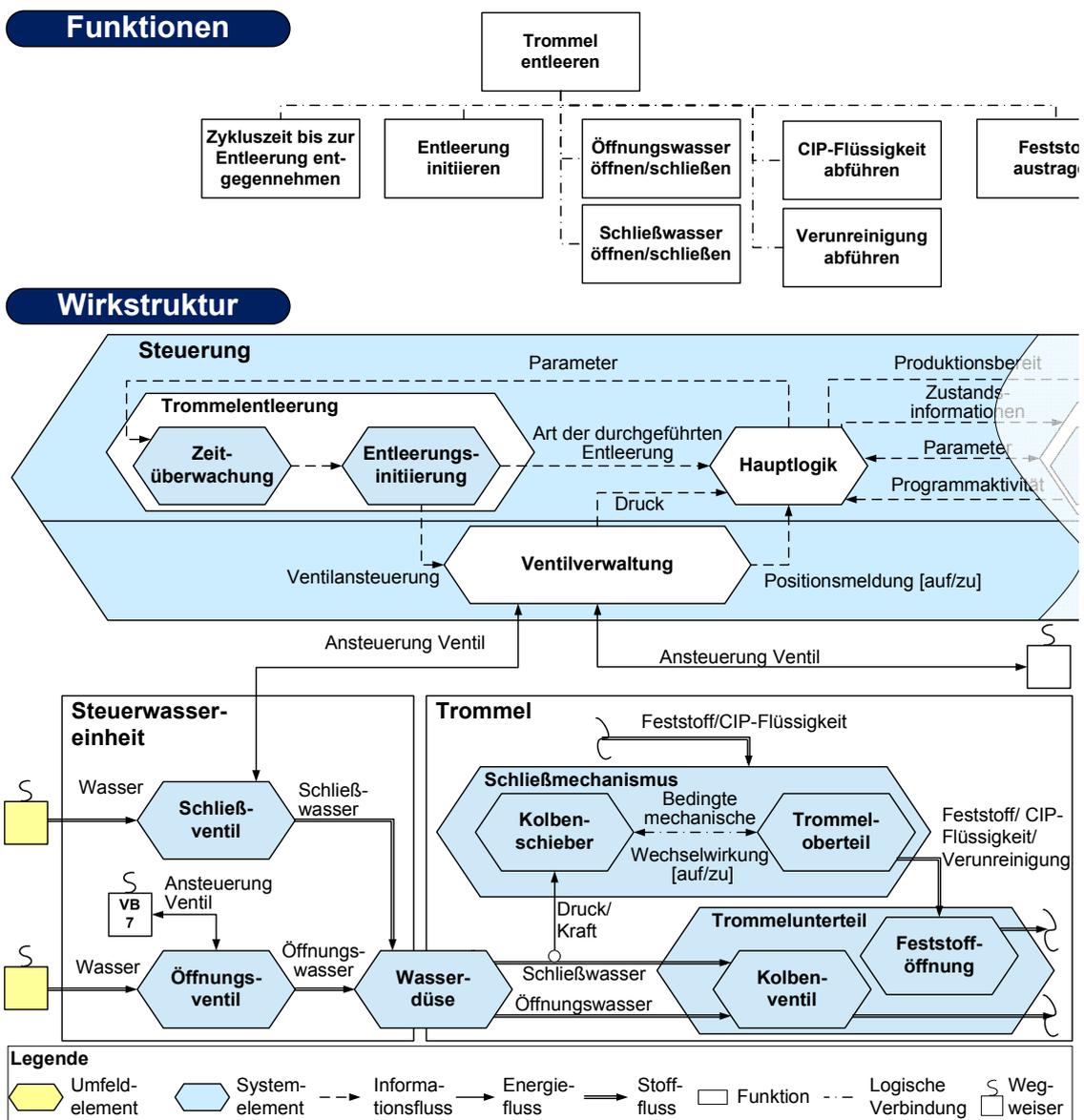


Bild 5-7: Ausgewählte Aspekte des Lösungsmusterkandidaten „Trommelentleerung“

Die Reduzierung beruht auf den zuvor ermittelten Erkenntnissen bzgl. der vorliegenden Aggregation aus Phase 2. In diesem Zusammenhang sind die Varianten zur Auslösung der Entleerungsinitiierung in der Steuerung und im mechanischen System nicht mehr vorhanden.

Die unterschiedlichen Möglichkeiten zur Auslösung einer Entleerung werden als separate Lösungsmuster aufgeführt. Um die Komplexität in den Lösungsmustern zu reduzieren, werden komplexe Muster in elementare unterteilt. Parallel zur Diskussion der Lösungsmusterkandidaten anhand von Wirkstruktur und Funktionen, findet die Analyse ihrer wechselseitigen Abhängigkeiten statt. Hierzu wird das resultierende Wirkgefüge mit Hilfe des multidimensionalen Wissensraums transparent visualisiert. Ein gemeinschaftliches Verständnis der identifizierten Musterkandidaten und ihrer Wechselwirkungen lässt sich insbesondere durch Zuordnung zu fachdisziplinspezifischen Lösungsprinzipien fördern. Diese sind in der Regel einzelnen Entwicklern zugeordnet. Der Wissensraum bietet allen Beteiligten eine Plattform, sich über das Zusammenwirken und Verschmelzen einzelner z.T. disziplinspezifischer Lösungsprinzipien auszutauschen. Hierdurch lässt sich insbesondere das Bewusstsein für disziplinübergreifende Schnittstellen für nachfolgende Entwicklungsprojekte schärfen. Die Relationen der Kandidaten untereinander, entlang der in Kapitel 4.3.3 erläuterten Dimensionen, sind in Bild 5-8 abgebildet.

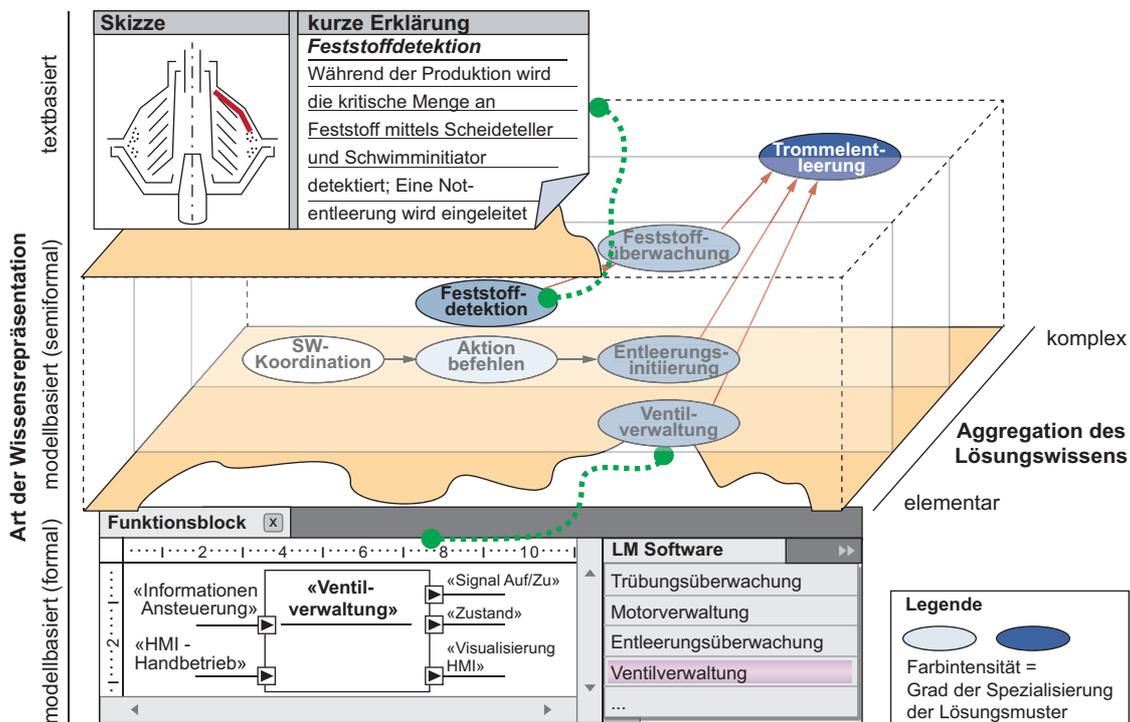


Bild 5-8: Relationen ausgewählter Lösungsmusterkandidaten im multidimensionalen Wissensraum

Die Aggregation des Lösungswissens ist größtenteils aus den in Bild 5-6 eingeführten Abhängigkeiten übernommen. So setzt sich die „Trommelentleerung“ in erster Instanz

zusammen aus „Feststoffüberwachung“, „Entleerungsinitiierung“ und „Ventilverwaltung“. Die „Entleerungsinitiierung“ ist dabei eine spezialisierte Ausprägung des Lösungsmusters „Aktion befehlen“, das sich wiederum der Klasse „SW-Koordination“ unterordnen lässt (vgl. Kapitel 4.3.2.1).

Ferner sind im multidimensionalen Wissensraum entlang einer dritten Dimension die Beziehungen zu etablierten Dokumenten oder Werkzeugen der einzelnen Fachdisziplinen dargestellt. So ist das Prinzip der Feststoffdetektion und die mechanische Expertise bestimmter Detailfragen bereits in einem Steckbrief dokumentiert. Als Pendant liegt u.a. detailreiches Know-how über die Ventilsteuerung in Form eines parametrierbaren SPS-Funktionsbausteins vor. Vor dem Hintergrund einer zielgerichteten Informationsbereitstellung während der zunehmenden Konkretisierung entlang einer Produktentwicklung, ist die Dokumentation derartiger Querbeziehungen von enormer Wichtigkeit.

#### 5.1.4 Phase 4: Dokumentation von Lösungsmustern

Den Abschluss des Vorgehensmodells bildet die Definition und Dokumentation der Lösungsmuster. Hierzu werden die ermittelten Informationen aus den vorangegangenen Phasen in die in Kapitel 4.3.1 eingeführte einheitliche Strukturierung von Lösungsmustern überführt. Neben der Expertise der Entwickler über spezifische Merkmale sind es vor allem die Informationen aus den Partialmodellen Wirkstruktur, Verhalten und Funktionen.

Ein beispielhaftes Lösungsmuster aus dem Gesamtsystem Separator zeigt Bild 5-9. Die Spezifikation des multidisziplinären Lösungsmusters „Trommelentleerung“ enthält als Problembeschreibung die Aspekte „Funktionen“ und „Merkmale“ und als Lösungsbeschreibung die „Wirkstruktur“ sowie „Verhalten-Aktivitäten“. Der Anwendungskontext ist die mechanische Trennung eines Rohprodukts in einem Tellerseparator, visualisiert im Systemmodell in Bild 5-2 und Bild 5-4. Vor diesem Hintergrund entfällt die erneute Modellierung.

**Problembeschreibung:** Übergeordnetes Problem (gleichzusetzen mit der Hauptfunktion), das durch das Lösungsmuster gelöst wird, ist die Entleerung einer rotierenden Trommel. Diese unterteilt sich u.a. in die Subfunktionen „Entleerung berechnen“, „Entleerung initiieren“ oder „Öffnungsmechanismus auslösen“. Ergänzt wird diese Beschreibung um Angaben zu spezifischen Merkmalen. So liegt ein Hauptaugenmerk, im Hinblick auf die Einsatzbedingungen des Lösungsmusters, auf der Zuverlässigkeit. Dies betrifft die Abläufe in der Steuerung, Sensorik/Aktorik sowie mechanische Komponenten gleichermaßen. Die Entleerung ist von entscheidender Bedeutung bei der Gewährleistung des sicheren Betriebs eines Separators. Weitere Merkmale beziehen sich u.a. auf die Kinematik. So ist beispielsweise als Voraussetzung für den Einsatz dieses Lösungsmusters die Rotation der Trommel zu nennen.

**Lösungsbeschreibung:** Die Wirkstruktur lässt sich auf höchster Ebene in die drei logischen Gruppen „Steuerung“, „Ventileinheit“ und „Trommel“ unterteilen. Die einzelnen Systemelemente sind im Hinblick auf eine Adaption in zukünftigen Entwicklungsprojekten bzw. -aufträgen abstrahiert. So ist z.B. die Übertragung auf spezialisierte Kundenanforderungen wie „Verwendung von Magnetventilen“ oder „Verwendung von Motorventilen“ sichergestellt. Wie eingangs im Rahmen der vorliegenden Arbeit erwähnt, ist der Grad der Abstraktion bzw. Generalisierung produkt- und unternehmensspezifisch festzulegen. Es muss sowohl die Übertragbarkeit als auch die Effizienzsteigerung durch ihren Einsatz sichergestellt werden. Komplettiert wird die Lösungsbeschreibung durch das Partialmodell Verhalten-Aktivitäten. Es repräsentiert die für den Einsatz des Lösungsmusters notwendigen generischen Abläufe in der Steuerung.

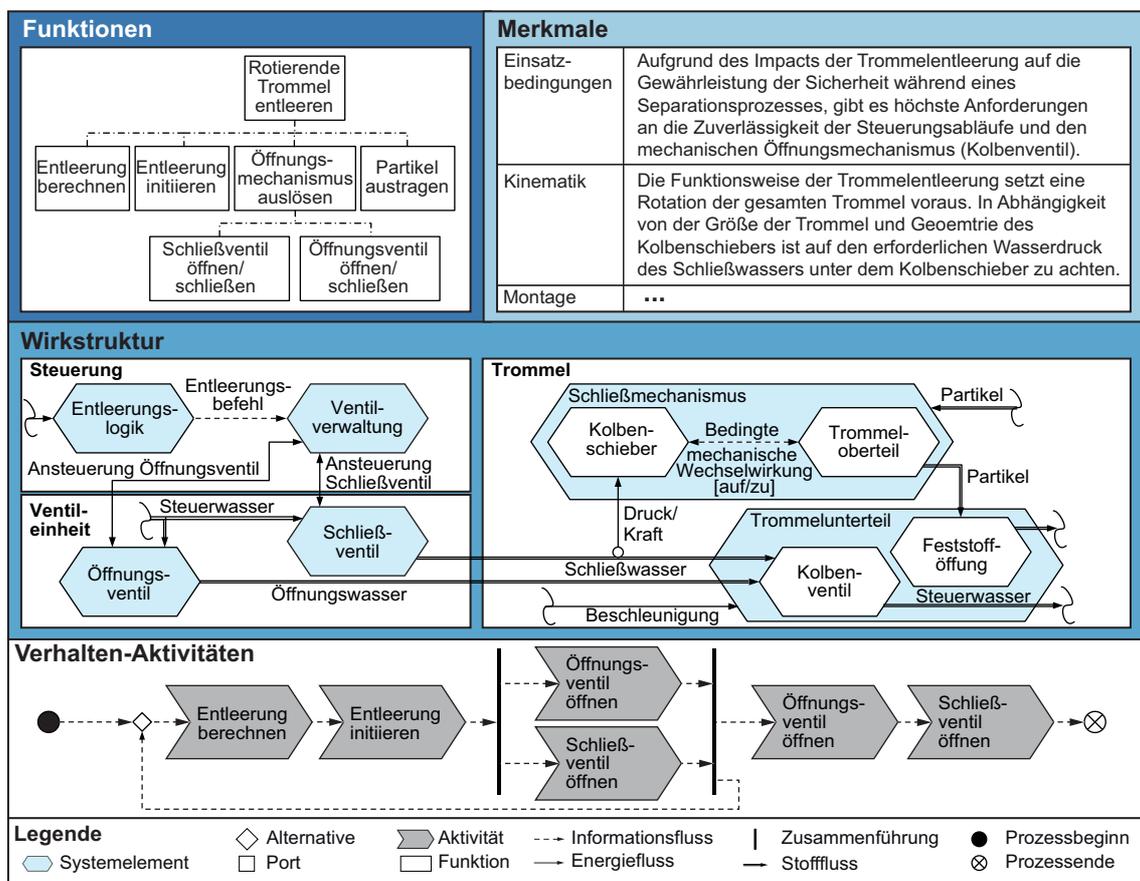


Bild 5-9: Multidisziplinäres Lösungsmuster „Trommelentleerung“

## 5.2 Anwendung des Vorgehensmodells für einen lösungsmusterbasierten Systementwurf am Beispiel „Kooperierende Deltaroboter“

Als Anwendungsbeispiel für das im Rahmen dieser Arbeit entwickelte *Vorgehensmodell für einen lösungsmusterbasierten Systementwurf* dient ein Robotersystem. Es ist in Bild 5-10 dargestellt. Das System besteht aus zwei identisch aufgebauten Robotern mit einer Delta-Kinematik sowie drei Elektromotoren. Im Tool-Center-Point ist eine Spiel-

platte angebracht die das Zuspieren eines Ball zwischen den beiden Robotern ermöglicht. Eine externe Erfassung zur Ermittlung der Flugkurve und des Auftreffpunkts erfolgt nicht. Jeder Roboter agiert als autonomer Spieler und berechnet die voraussichtliche Flugbahn des Balls. Hierzu werden zwischen den Robotern Informationen über den aktuell durchgeführten Schlag ausgetauscht: Abschlagzeitpunkt, Abschlagkoordinaten, Schlagstrategie. Die berechneten Daten werden mit den tatsächlichen Daten beim Auftreffen des Balls verglichen. Dabei werden Abweichungen erkannt und ausgeglet. Ein Schwerpunkt in der Entwicklung des Systems ist die integrierte Spezifikation der software- und regelungstechnischen Anteile. Dies betrifft die frühe fachdisziplinübergreifende Phase des Systementwurfs in gleichem Maße wie die anschließende Konkretisierung. Am Beispiel des Demonstrators „Kooperierende Deltaroboter“ wird veranschaulicht, wie die Erarbeitung der Zielbestimmung, die Suche und Auswahl von Lösungsmustern sowie der darauf aufbauende Systementwurf abläuft.



*Bild 5-10: Anwendungsbeispiel „Kooperierende Deltaroboter“*

### **5.2.1 Phase 1: Initiale Zielbestimmung**

Der erste Prozessschritt im Rahmen des Systementwurfs ist die Zielbestimmung. Innerhalb der Zielbestimmung werden das Umfeldmodell, eine Anforderungsliste, Anwendungsszenarien und eine Funktionshierarchie erarbeitet. Die Vorgehensweise sieht grundsätzlich vor, dass alle gesammelten Erkenntnisse bei der Erstellung des Umfeldmodells und der Anwendungsszenarien in die Anforderungsliste und anschließend in die Funktionshierarchie einfließen (vgl. Kapitel 4.5.1). Bild 5-11 zeigt das Umfeldmodell sowie einen ersten Stand der Anforderungsliste. Ferner sind die Zusammenhänge abgebildet, die zwischen dem Umfeldmodell und der Anforderungsliste bestehen.

### Umfeldmodell

Im Mittelpunkt des Umfeldmodells steht der zu entwickelnde Roboter, gekennzeichnet mit der Beschriftung „Roboter 1“. Da das gesamte Robotersystem aus zwei identisch aufgebauten Robotern bestehen soll, ist die Betrachtung nur eines Roboters im Entwurf ausreichend. Um die Interaktion adäquat abzubilden, ist der „Roboter 2“ als Umfild-element spezifiziert. Weitere Umfild-elemente sind „Benutzer“, „Spielobjekt“, „Umwelt“, „Energiequelle“ und „Untergrund“. Wechselseitige Beziehungen zwischen dem System „Roboter 1“ und den Umfild-elementen sind als Flussbeziehung dargestellt. Es wird zwischen störenden und gewollten Flussbeziehungen unterschieden. Beispiele für störende Einflüsse für „Roboter 1“ sind „Stöße“ oder „Temperatur“ ausgehend von der „Umwelt“. Ein Beispiel für eine beabsichtigte Flussbeziehung ist der „Datenaustausch“ mit „Roboter 2“. Die gesammelten Informationen werden sukzessive in eine Anforderungsliste überführt.

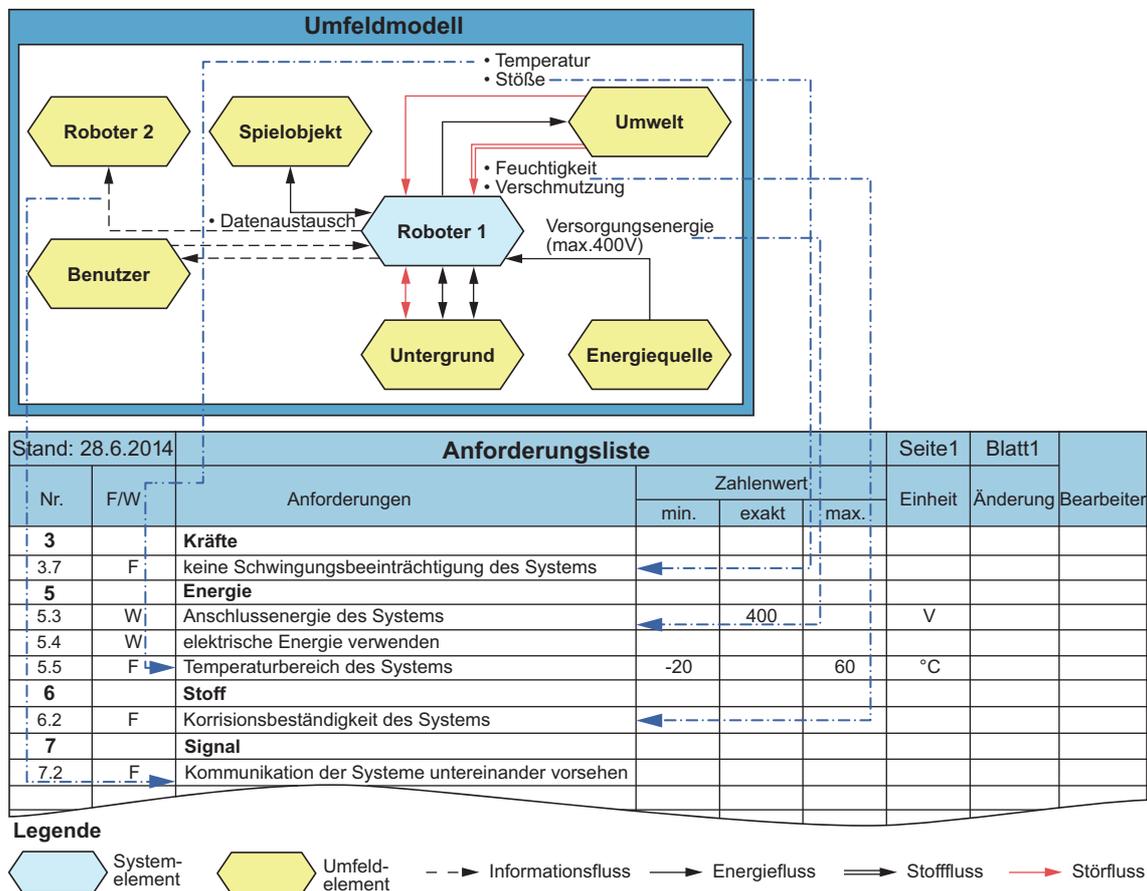


Bild 5-11: Darstellung des Umfeldmodells, der Anforderungsliste und der bestehenden Zusammenhänge

### Anforderungsliste

Aus den im Umfeldmodell identifizierten Wechselwirkungen wie z.B. „Feuchtigkeit“, „Verschmutzung“ oder „Stöße“ ergeben sich die Anforderungen der „Korrosionsbe-

ständigkeit“ oder „keine Schwingungsbeeinträchtigung des Systems“. Aus der Flussbeziehung des Roboters mit der externen Energiequelle resultiert die Wunschforderung für die Anschlussleistung von max. 400 V. Damit wird dem geplanten Einsatzort, z.B. auf einer Messe, Rechnung getragen. Als weitere Anforderungen sind zum einen der „Temperaturbereich“ und zum anderen die geforderte „Kommunikation“ mit einem weiteren Roboter zu nennen. Die Zuordnung zu übergeordneten Kategorien orientiert sich an den Kategorien nach PAHL/BEITZ [PBF+07].

**Anwendungsszenarien**

Parallel zur Erstellung des Umfeldmodells und der Überführung der Informationen in eine Anforderungsliste werden Anwendungsszenarien erarbeitet. Bild 5-12 (rechts oben) verdeutlicht schematisch z.B. das Szenario „Erfolgreiches Spielen“. Es wird textuell beschrieben, wie die Deltaroboter im Normalfall miteinander agieren sollen. Ein Beispiel für ein weiteres Szenario ist z.B. „Entfernung des Spielobjektes durch den Benutzer“. In diesem Fall wird in dem zugehörigen Anwendungsszenario beschrieben, in welche Ruheposition die Roboter beispielsweise gehen sollen. Darauf aufbauend ist es denkbar und für den sicheren Betrieb sinnvoll, dass der Roboter kontinuierlich seinen Status an eine Schnittstelle zum Benutzer weitergibt.

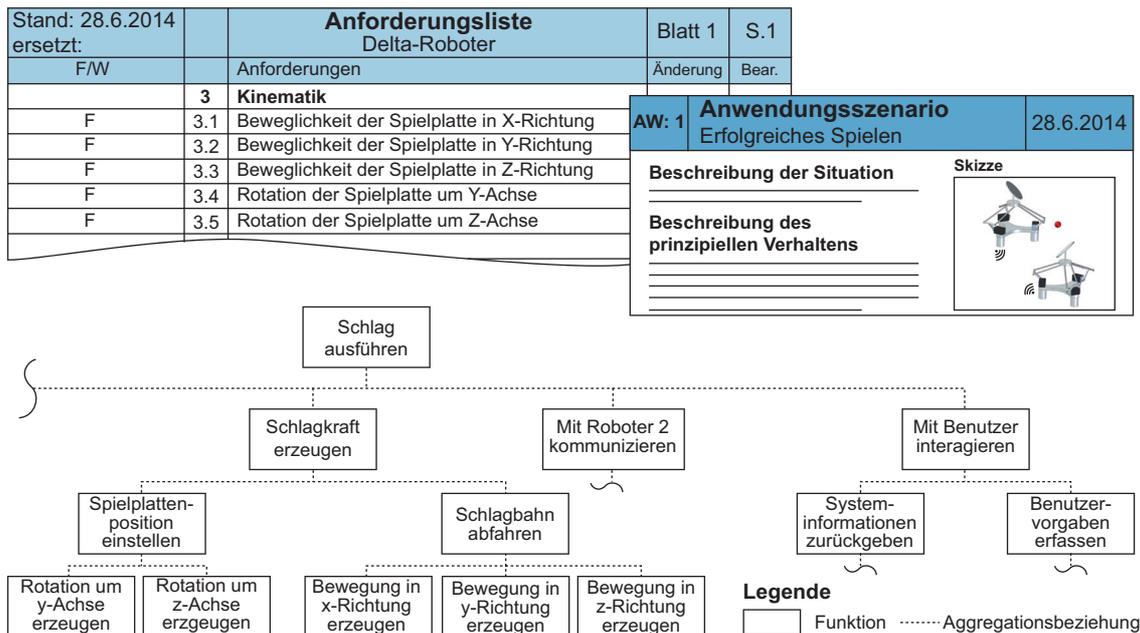


Bild 5-12: Definition von Anwendungsszenarien und Ableitung einer initialen Funktionshierarchie

**Funktionshierarchie**

Abschließend wird zur Grundlage für die Suche in Phase 2 eine initiale Version der Funktionshierarchie erstellt. Hierzu werden zum einen alle funktionalen Anforderungen in Teilfunktionen überführt. Zum anderen lassen sich aus den definierten Anwendungsszenarien zusätzliche Funktionen ableiten wie z.B. „Systeminformation zurückgeben“.

In Bild 5-12 (unten) sind die notwendigen Teilfunktionen dargestellt, die zum „Erzeugen der Schlagkraft“ benötigt werden sowie die Funktionen zur „Kommunikation mit Roboter 2“. Hierzu zählen das „Erzeugen der Bewegungen“ in den drei Raumrichtungen sowie die „Rotation der Spielplatte“ um zwei Achsen zum Einstellen eines Abschlagwinkels. Zusätzlich wurde der dezentrale Datenaustausch in die einzelnen Funktionen aufgeteilt. Prinzipiell wird die Hauptfunktion des Robotersystems so lange unterteilt, bis die zuvor gesammelten Informationen durch die Funktionen abgedeckt sind. Das Hauptaugenmerk liegt dabei auf dem lösungsneutralen Charakter der formulierten Funktionen. Die anschließende Suche nach möglichen Lösungsmustern ist Gegenstand der nächsten Phase.

## 5.2.2 Phase 2: Suche nach Lösungsmustern

Die Basis einer ersten Suche nach passenden Lösungsmustern bilden die Teilfunktionen in der Funktionshierarchie sowie die Anforderungen. Bild 5-13 veranschaulicht den Ablauf einer ersten Vorauswahl. Die Grafik zeigt das Zusammenspiel der Informationen aus dem Entwurf (disziplinübergreifende System-Ebene), dem Wissensraum und der Gegenüberstellung der ausgewählten Lösungsmuster in einem morphologischen Kasten.

Wie im Rahmen der vorliegenden Arbeit erwähnt, beinhaltet der Wissensraum Lösungsmuster auf unterschiedlichen Aggregations- und Spezialisierungsniveaus. Bei der Anwendung des Vorgehens, speziell bei der Suche, hat der Entwickler bei der Funktion „Bewegung erzeugen“ die Wahlmöglichkeit zwischen den Lösungsmustern „elektrischer Antrieb“, „hydraulischer Aktor“ und „pneumatischer Aktor“. Letztlich erfüllt jeder Aktor die geforderte Teilfunktion. Auch die Berücksichtigung der Anforderung „Anschlussleistung max. 400 V“ würde die zur Verfügung stehenden Lösungsmuster im Suchraum nur bedingt einschränken. Zu diesem Zeitpunkt im Entwurf ist erstmals seitens der Entwickler eine Entscheidung über die Umsetzung des Systems notwendig.

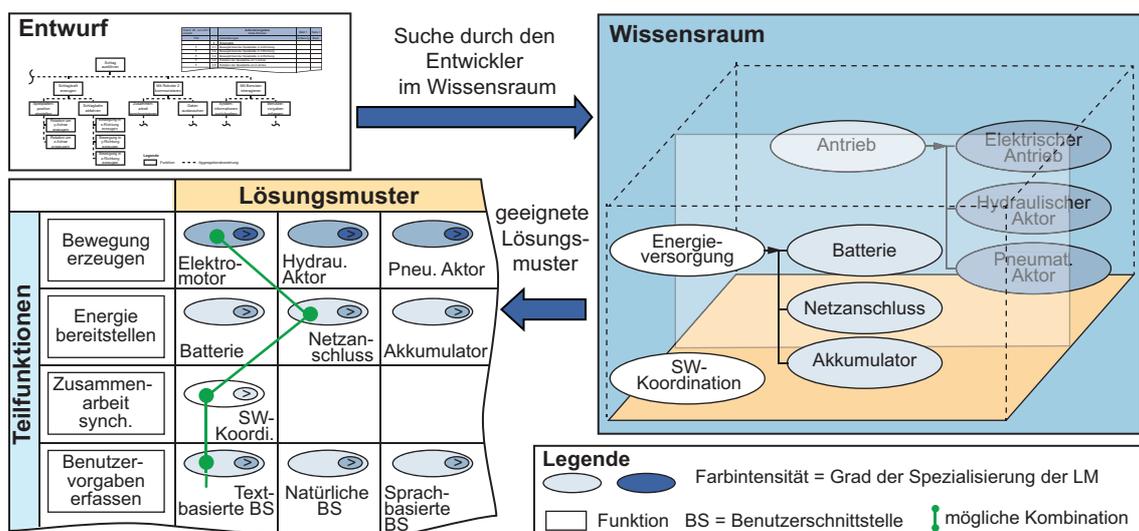


Bild 5-13: Vorauswahl möglicher Lösungsmuster

Eine ähnliche Entscheidung ist für die Auswahl eines Lösungsmusters zur Umsetzung der Teilfunktion „*Benutzervorgaben erfassen*“ erforderlich. So führt die genannte Funktion bei einer Suche im Wissensraum beispielsweise zu einer „*textbasierten*“ oder „*sprachbasierten Benutzerschnittstelle*“. Auch wenn diese Wahl für den Aufbau des Roboters nur eine untergeordnete Bedeutung hat, determiniert sie die weitere Ausarbeitung der Software.

Ein weiteres Beispiel ist das Suchergebnis für die Teilfunktion „*Zusammenarbeit synchronisieren*“. Die Funktion resultiert aus einem Anwendungsszenario, aus dem hervor geht, dass die Roboter autonom und nicht mit Hilfe einer Zentralsteuerung agieren sollen. Eine detaillierte Spezifikation ist zu diesem Zeitpunkt des Entwurfs ohne Kenntnisse über den tatsächlichen Systemaufbau nur schwer möglich. Die Entwickler finden daher bei der Suche beispielsweise das Lösungsmuster „*Software-Koordination*“. Es beschreibt sehr generalisiert die Möglichkeit der Koordination zwischen Softwarekomponenten. Der Generalisierungsgrad ist so hoch, dass diesbezüglich eine erneute verfeinerte Suche in den nachfolgenden Phasen des Vorgehensmodells erforderlich sein wird. Zur besseren Navigation werden die Entwickler durch entsprechende Piktogramme auf spezialisiertere Lösungsmuster – falls vorhanden – aufmerksam gemacht. In Bild 5-13 sind die Piktogramme der Einfachheit lediglich im morphologischen Kasten dargestellt.

Im abschließenden Schritt werden passende Lösungsmuster im morphologischen Kasten ausgewählt und hinsichtlich ihrer Kombinationsverträglichkeit analysiert. Dieser Prozess erfordert das Erfahrungswissen der Entwickler. Er kann nur bedingt durch Angaben innerhalb eines Lösungsmusters unterstützt werden.

### 5.2.3 Phase 3: Konkretisierung der Zielbestimmung

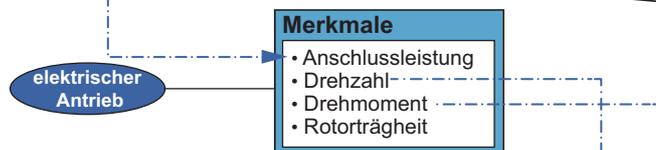
Nach der Vorauswahl von Lösungsmustern erfolgt eine Konkretisierung der Zielbestimmung. Wie bereits erwähnt, sind die Definition der Aufgabenbeschreibung und die Auswahl von Lösungsmustern hochgradig iterativ. So hat die Wahl eines Lösungsmusters Einfluss auf die Verfeinerung der Funktionshierarchie und der Anforderungsliste und bedingt eine erneute Suche. Weiterhin können sich bei der Auswahl neue Merkmale ergeben, die existierende Anforderungen detaillieren oder neue Restriktionen für die Auswahl spezialisierter Lösungsmuster definieren.

Jedes Lösungsmuster beinhaltet eine definierte Liste von Merkmalen, die für das jeweilige Muster charakteristisch sind. Der Entwickler überführt die Merkmale der ausgewählten Lösungsmuster in die Anforderungsliste. Anschließend verfeinert er die Anforderungen durch Festlegung spezifischer Werte. Bild 5-14 zeigt die Konkretisierung der Anforderungen des Anwendungsbeispiels anhand des ausgewählten Lösungsmusters „*elektrischer Antrieb*“.

Das Lösungsmuster „*elektrischer Antrieb*“ liefert die Merkmale „*Drehzahl*“, „*Drehmoment*“, „*Anschlussleistung*“, „*Rotorträgheit*“ und „*Gewicht*“. Die Anschlussleistung

wurde bei der Auswahl des Musters bereits berücksichtigt. Der Entwickler hat nun die Aufgabe, die Anforderungsliste um Anforderungen bzgl. des benötigten „Drehmoments“, der „Drehzahl“, der „Rotorträgheit“ und des „Gewichts“ zu erweitern. Die Konkretisierung hilft dem Entwickler unter zweierlei Gesichtspunkten. Zum einen lässt sich im Anschluss an die Verfeinerung eine spezifische Suche im Wissensraum durchführen. Zum anderen führt die Auswahl erster generalisierter Lösungsmuster zu weiterführenden Erkenntnissen für die der Art des Systemaufbaus. Das Drehmoment des Elektromotors hat z.B. Auswirkungen auf die Komponenten, die mit der Welle verbunden werden. Derartige Erkenntnisse sind für alle ausgewählten Lösungsmuster in der Anforderungsliste zu ergänzen. Diese sukzessive Konkretisierung auf Basis der vorhandenen Informationen in den Lösungsmustern erhöht die Wahrscheinlichkeit, dass alle relevanten Anforderungen an das System festgehalten werden. Späte, mitunter sehr kostenintensive Änderungen lassen sich auf diese Weise verringern.

Stand: 26.7.2014		Anforderungsliste			Seite1	Blatt1		
Nr.	F/W	Anforderungen	Zahlenwert			Einheit	Änderung	Bearbeiter
			min.	exakt	max.			
<b>3</b>		<b>Kräfte</b>						
3.7	F	keine Schwingungsbeeinträchtigung des Systems						
<b>5</b>		<b>Energie</b>						
5.3	W	Anschlussenergie des Systems		400		V		
5.4	W	elektrische Energie verwenden						
5.5	F	Temperaturbereich des Systems	-20		60	°C		



Stand: 28.7.2014		Anforderungsliste			Seite1	Blatt1		
Nr.	F/W	Anforderungen	Zahlenwert			Einheit	Änderung	Bearbeiter
			min.	exakt	max.			
<b>3</b>		<b>Kräfte</b>						
3.1		Drehzahl des Elektromotors	6.000		9.000	U/Min	M	
3.2		Nenn Drehmoment des Elektromotors	25		35	Nm	M	
3.7	F	keine Schwingungsbeeinträchtigung des Systems					M	
<b>5</b>		<b>Energie</b>						
5.3	W	Anschlussenergie des Systems		400		V		
5.4	W	elektrische Energie verwenden						
5.5	F	Temperaturbereich des Systems	-20		60	°C		

**Legende**

Lösungsmuster

M Anforderung aus der Merkmalsliste eines Lösungsmusters

Bild 5-14: Konkretisierung der Anforderungsliste am Beispiel „elektrischer Antrieb“

Anforderungen die aus den Merkmalen von Lösungsmustern hervorgehen, werden in der Anforderungsliste mit einem „M“ gekennzeichnet (vgl. Bild 5-14). Sie sind entsprechend des Charakters eines Lösungsmusters in gewissem Umfang änderbar, müssen aber über alle Elemente im System konsistent sein.

Ebenso wie die Anforderungsliste erfordert auch die Funktionshierarchie eine Konkretisierung. Bild 5-15 verdeutlicht anhand eines Auszugs die getroffenen Erweiterungen. Auf Basis der Auswahl des „*elektrischen Antriebs*“ ist die Teilfunktion „*elektr. Energie in mech. Energie wandeln*“ hinzugekommen. Ferner steht die neue Teilfunktion „*Stromfluss regeln*“ für eine notwendige Beeinflussung des dynamischen Systemverhaltens zur Gewährleistung eines konstanten Spielflusses.

Darüber hinaus ist es erforderlich, die Funktion „*Mit Roboter 2 kommunizieren*“ zu detaillieren. Die Vorauswahl von Lösungsmustern hat auf der bis dato existierenden Hierarchisierungsstufe nur ein sehr generisches Lösungsmuster ergeben<sup>67</sup>. Die Funktion wird folglich weiter unterteilt in die Subfunktionen „*Spielanfrage senden*“ und „*Antwort empfangen*“. Beide Funktionen werden in Absprache mit den Entwicklern aus der Softwaretechnik weiter verfeinert. Gleiches gilt für die Subfunktionen bzgl. der Interaktion mit dem Benutzer.

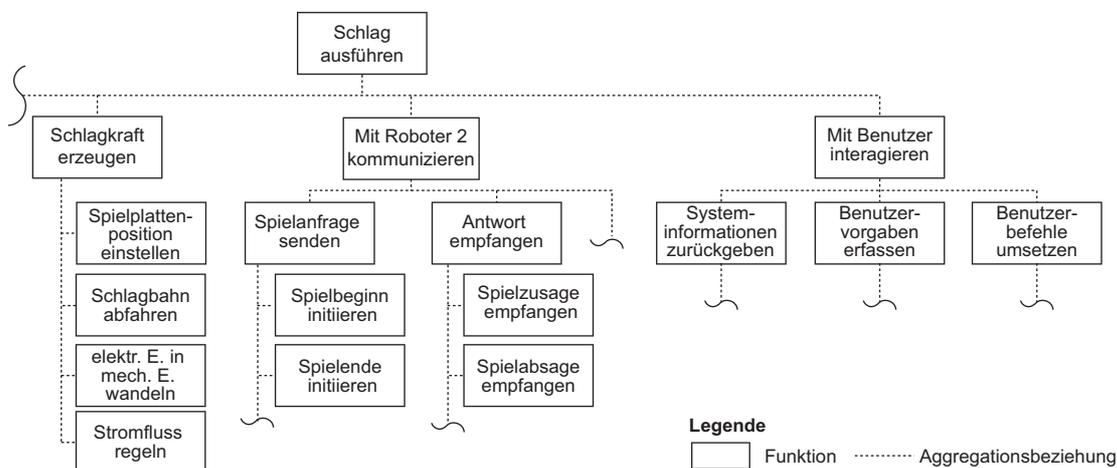


Bild 5-15: Erweiterung der Funktionshierarchie (Auszug)

#### 5.2.4 Phase 4: Suche nach spezialisierten Lösungsmustern

Die Detailsuche nach Lösungsmustern verläuft prinzipiell analog zur ersten Suche anhand der Teilfunktionen. Eingangsinformationen sind die zuvor ausgewählten Lösungsmuster und die in der Zwischenzeit detaillierten Funktionen und Anforderungen. Bild 5-16 veranschaulicht das prinzipielle Vorgehen bei der Suche nach spezialisierten Lösungsmustern anhand ausgewählter Beispiele (vgl. bzgl. Funktionen auch Bild 5-15).

Der Wissensraum zeigt im Vergleich zur Vorauswahl (vgl. Bild 5-13) spezialisiere Lösungsmuster, die den bereits ausgewählten Lösungsmustern zugeordnet sind. Mit Hilfe der Anforderungen wird das Suchergebnis immer weiter eingeschränkt. Das Er-

<sup>67</sup>Die Vorgehensweise orientiert sich an dem Top-Down-Vorgehen. Es wird zur Veranschaulichung der Methode davon ausgegangen, dass beispielsweise das in Kapitel 4.3.2.1 beschriebene komplexe Lösungsmuster noch nicht im Wissensraum zu finden ist.

gebnis der Detailsuche wird wiederum in einem morphologischer Kasten festgehalten. In der Regel werden die Funktionen zeilenweise den Lösungsmustern zugeordnet. Komplexe Lösungsmuster, die mehr als eine Funktion erfüllen, bilden eine Ausnahme. Zur Veranschaulichung dieser Zusammenhäng sowie der sukzessiven Einschränkung des Wissensraums dienen die beiden Funktionen „*Bewegung erzeugen*“ und „*Zusammenarbeit synchronisieren*“.

Im Rahmen der Vorauswahl wurde zur Erfüllung der Funktion „*Bewegung erzeugen*“ das Lösungsmuster „*elektrischer Antrieb*“ ausgewählt. Dieses beinhaltet auch zugleich die Funktion „*elektr. Energie in mech. Energie wandeln*“. Der morphologische Kasten wird um die erweiterten Funktionalitäten der Lösungsmuster ergänzt. Ferner sind aufgrund der Vorauswahl im Zuge der voranschreitenden Konkretisierung weitere Funktionen hinzugekommen. Beispiele sind in Bild 5-16 die Funktionen „*Stromfluss regeln*“ und „*Rotation in Translation wandeln*“. Beide Teilfunktionen gehören zur Funktion „*Bewegung erzeugen*“ mit einem direkten Bezug zu dem „*elektrischen Antrieb*“. Eine erneute Suche im Wissensraum führt die Entwickler zum Lösungsmuster „*Servoantrieb*“. Es erfüllt die beiden Funktionen „*elektr. Energie in mech. Energie wandeln*“ sowie „*Stromfluss regeln*“. Es handelt sich folglich hinsichtlich des Aggregationsgrades um ein komplexeres Lösungsmuster. Hinsichtlich der Position im Wissensraum ist zu erwähnen, dass es eine spezialisierte Form eines „*elektrischen Antriebs*“ ist und zugleich Lösungsmuster für die Regelung wie z.B. „*Feedback*“ enthält. Um die Bewegung in die erforderlichen Raumrichtungen zu erzeugen, ist zusätzlich das Lösungsmuster „*Exzenterstange*“ erforderlich.

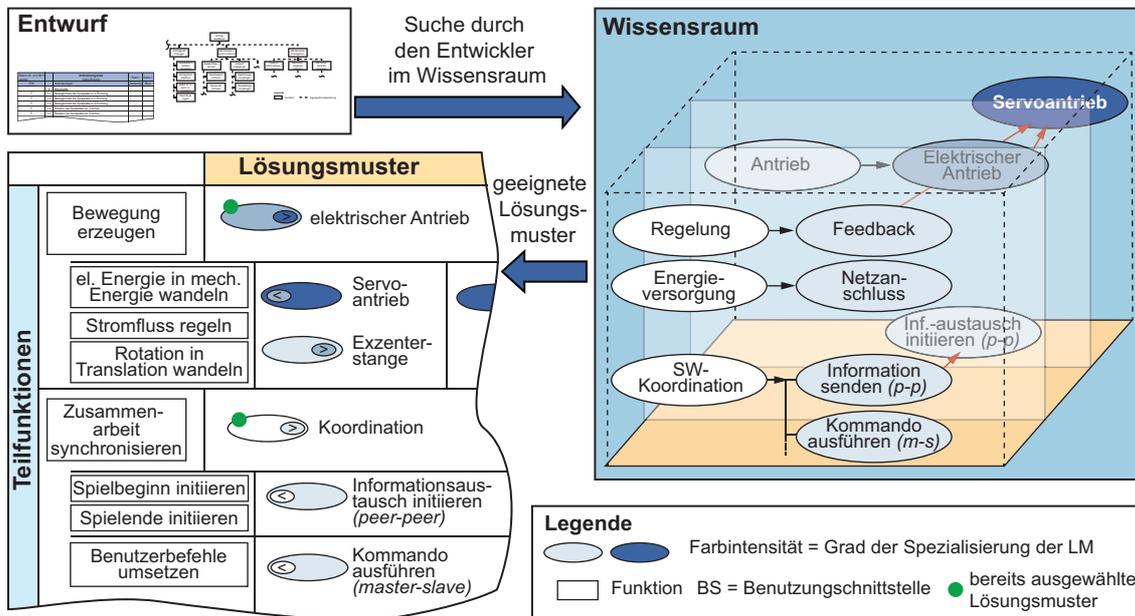


Bild 5-16: Detailsuche nach Lösungsmustern

Ein weiteres Beispiel für ein zu konkretisierendes Lösungsmuster ist „*Koordination*“ mit der zugehörigen Teilfunktion „*Zusammenarbeit synchronisieren*“<sup>68</sup>. Zum Zeitpunkt der Vorauswahl waren die Informationen über den Systemaufbau noch zu abstrakt. Erst mit zunehmender Konkretisierung auf Gesamtsystemebene ist ab einem bestimmten Zeitpunkt eine Unterteilung der Softwarefunktionalität möglich. So ist die Funktion zur Synchronisation weiter unterteilt in die Subfunktionen „*Spielbeginn initiieren*“, „*Spielende initiieren*“ und z.B. „*Benutzerbefehle umsetzen*“. Für die beiden erstgenannten Funktionen finden die Entwickler beispielsweise das Lösungsmuster „*Informationsaustausch initiieren*“ mit einer Rechtevergabe der Ports vom Typ „*peer to peer*“. Anhand der Position im Wissensraum lässt sich erkennen, dass es sich um ein aggregiertes Lösungsmuster handelt, das u.a. das elementare Lösungsmuster „*Informationen senden (p-p)*“ beinhaltet. Beide Lösungsmuster sind vom Spezialisierungsgrad identisch, unterscheiden sich jedoch hinsichtlich der Aggregation. Ein weiteres Beispiel für ein elementares Lösungsmuster, das direkt aufgrund der Teilfunktion „*Benutzerbefehle umsetzen*“ gefunden wird, ist das Lösungsmuster „*Kommando ausführen (master-slave)*“.

In einem letzten Schritt der Suche werden die Lösungsmuster erneut miteinander kombiniert. Die Entwickler berücksichtigen hierzu alle vorliegenden Informationen über das Gesamtsystem und lassen diese in die Auswahl und Kombination mit einfließen. Es folgt die eigentliche Synthese der Prinziplösung in der abschließenden Phase 5.

### 5.2.5 Phase 5: Lösungsmusterbasierte Systemmodellierung

Der Aufbau der Prinziplösung des Deltaroboters startet mit der Wirkstruktur im fortlaufenden Wechselspiel mit den erforderlichen Partialmodellen zum Verhalten – in diesem Anwendungsbeispiel sind es Verhalten-Sequenz und Verhalten-Zustände. Zunächst werden die Informationen über die Systemelemente der einzelnen Lösungsmuster aus dem Wissensraum abgerufen. Jedes Lösungsmuster enthält generalisierte Informationen, die den Entwickler bei der Synthese unterstützen. Die Auswahl und Adaption entsprechend aufbereiteter Informationen führt zu einer signifikanten Zeitersparnis im Entwurfsprozess.

Als Ergebnis des Systementwurfs zeigt Bild 5-17 einen Ausschnitt aus der erstellten Prinziplösung der kooperierende Deltaroboter. Zu sehen sind Partialmodelle Wirkstruktur, Verhalten-Zustände und Verhalten-Sequenz. Zur Visualisierung von Zusammenhängen zwischen den Partialmodellen sind Ablaufvorgänge in der Steuerung die auf ausgewählten Lösungsmustern basieren, farblich hervorgehoben. Diese, sowie Lö-

---

<sup>68</sup>Zur Erläuterung der Sachverhalte zwischen den Informationen aus der Entwicklung und dem Wissensraum wird an dieser Stelle davon ausgegangen, dass sich das Lösungsmuster „*Zusammenarbeit synchronisieren*“ (vgl. Kapitel 4.3.2.1) nicht im Wissensraum befindet. Das Vorgehen wird stattdessen mit weniger komplexen Lösungsmustern erläutert.

lungsmuster auf Basis derer die Wirkstruktur modelliert ist, werden im Folgenden erläutert.

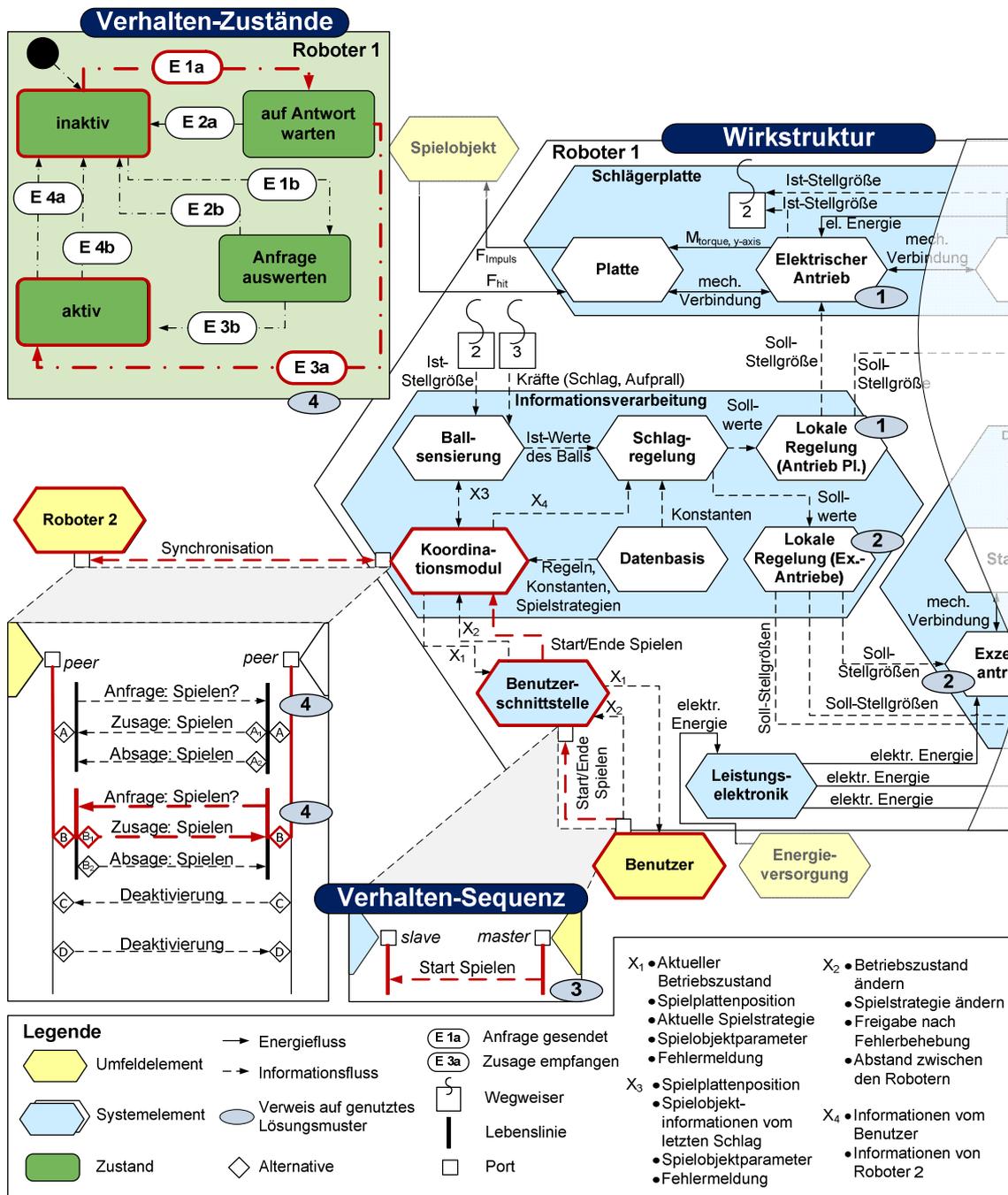


Bild 5-17: Prinziplösung des Anwendungsbeispiels „Kooperierende Deltaroboter“ (Ausschnitt)

Wie am Beispiel des „Servoantriebs“ gezeigt, können sich Lösungsmuster aus anderen Lösungsmustern zusammensetzen und somit bereits eine komplexere Wirkstruktur beinhalten (vgl. Kapitel 4.3.2.2). Beim Aufbau der Prinziplösung werden die Systemelemente der Lösungsmuster nicht nur in die **Wirkstruktur** integriert und mit Flussbeziehungen verbunden. In der Regel erfordert eine spezifische Ausprägung des zugrundelie-

genden Problems eine Adaption der bereitgestellten Wirkstruktur. Dies zeigt beispielsweise der Verweis „1“ auf das verwendete Lösungsmuster „*Servoantrieb*“. Zu sehen sind die beiden zugehörigen Systemelemente „*lokale Regelung*“ und „*elektrischer Antrieb*“. Das Lösungsmuster fungiert nach wie vor als Einheit, seine Bestandteile sind jedoch in zwei übergeordnete logische Gruppen von Systemelementen aufgeteilt. Die „*lokale Regelung*“ ist Teil der „*Informationsverarbeitung*“ wohingegen der „*elektrische Antrieb*“ ein Subelement der „*Schlägerplatte*“ ist. Ähnlich verhält es sich mit der Kombination der Lösungsmuster „*Servoantrieb*“ und „*Exzenterstange*“, hervorgehoben durch den Verweis „2“ in Bild 5-17.

In diesem Fall ist ebenfalls die Regelung des „*Servoantriebs*“ in die „*Informationsverarbeitung*“ integriert. Der übrige Teil des Antriebs wurde mit dem Lösungsmuster „*Exzenterstange*“ kombiniert und in der Wirkstruktur durch das Systemelement „*Exzenterantrieb*“ dargestellt. Zur Vervollständigung der Delta-Kinematik wird der „*Exzenterantrieb*“ dreimal über Verbindungsstäbe mit der Schlägerplatte verbunden. Ein Hauptaugenmerk liegt neben den größtenteils physikalisch geprägten Lösungsmustern auf den Lösungsmustern zur Spezifikation der Kommunikation.

Beispiele für solche Lösungsmuster zeigen die Verweise „3“ und „4“ in Bild 5-17. Die verwendeten Lösungsmuster liefern keine neuen Systemelemente, sondern fokussieren sich auf den Informationsaustausch zwischen zwei Systemelementen. Hinterlegte Informationen über die Modelle **Verhalten-Zustände** und **Verhalten-Sequenz** beeinflussen den Aufbau der Wirkstruktur demnach nicht direkt. Der im Lösungsmuster spezifizierte Informationsfluss lässt sich jedoch in der Wirkstruktur zuordnen und im Zusammenspiel mit dem Partialmodell Verhalten-Zustände spezifizieren. Die durch die Muster hervorgerufene Interaktion wird beispielhaft an dem farblich hervorgehobenen Zusammenspiel verdeutlicht. Der eigentliche Spielvorgang zwischen den beiden Deltarobotern wird durch den Benutzer initiiert. In der technischen Übersetzung geschieht dies durch das Lösungsmuster „*Kommando ausführen*“, wobei der Benutzer vom Typ „*master*“ und der Roboter als ausführender Akteur vom Typ „*slave*“ ist. Über das Systemelement „*Benutzerschnittstelle*“ wird der Befehl an das sog. „*Koordinationsmodul*“ weitergegeben. Zwischen dem „*Koordinationsmodul*“ und dem „*Roboter 2*“ findet im Vorfeld des Spielvorgangs die Synchronisation statt. Es handelt sich hierbei um einen komplexen Ablauf von z.T. aufeinanderfolgenden Informationsflüssen, dargestellt in dem Partialmodell Verhalten-Sequenz. Die Entwickler verwendeten bei der Synthese dieser Sequenz das Lösungsmuster „*Informationsaustausch initiieren*“, gekennzeichnet durch den Verweis „4“ an den beiden Alternativen „A“ und „B“. In diesem Fall wird das gleiche Lösungsmuster zweimal verwendet, entsprechend adaptiert und kombiniert. Komplettiert wird die Sequenz durch die Alternativen „C“ und „D“, die für eine „*Deaktivierung*“ des Spielvorgangs sorgen. Farblich hervorgehoben ist das Szenario, dass „*Roboter 1*“ die Anfrage zum Spielen sendet und „*Roboter 2*“ die Anfrage positiv bestätigt.

Dies wiederum bedeutet ebenfalls, dass Roboter 1 von dem Zustand „inaktiv“ in den Zustand „aktiv“ wechselt (vgl. Bild 5-17, Verhalten-Zustände). Der abgebildete Zustandsautomat ist ebenfalls den verwendeten Lösungsmustern entnommen. Die erforderliche Modifikation im Rahmen des Entwurfs durch die Entwickler zeigt Bild 5-18. Ausgehend von dem Fall, dass der Roboter in den Zustand „aktiv“ wechselt, finden systemintern weitere Zustandsübergänge statt. Diese sind im gezeigten Anwendungsbeispiel individuell entworfen. Die untergeordneten Zustände sind: „Initialisierung“, „Standby“, „Spielvorbereitung“, „Zuspielen“ und „Spielende“. Die Zustandsübergänge geschehen nicht beliebig, sondern sind ereignisgetrieben.

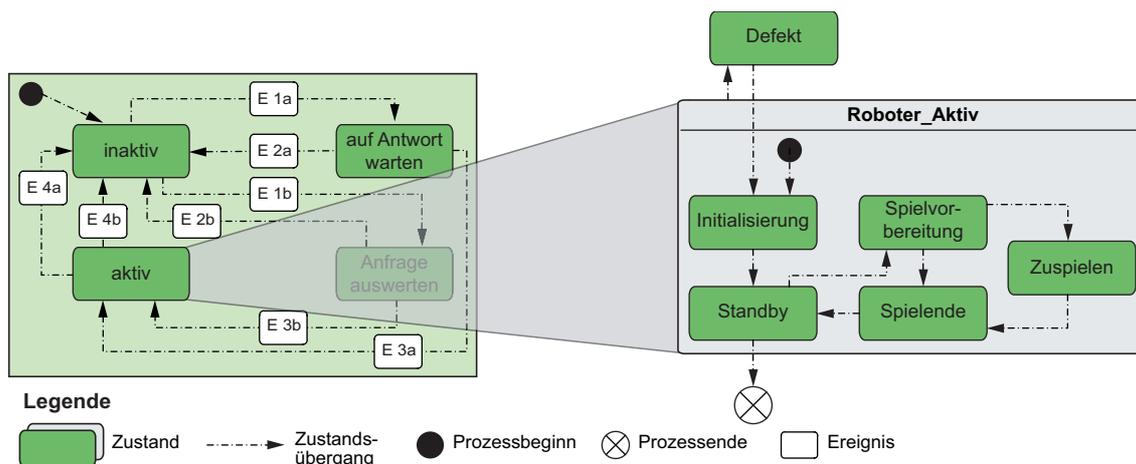


Bild 5-18: Konkretisierung des Partialmodells Verhalten-Zustände auf Basis verwendeter Lösungsmuster (Ausschnitt)

### 5.3 Bewertung der Arbeit anhand der Anforderungen

Im Rahmen dieses Kapitels findet abschließend eine Bewertung des erarbeiteten *Instrumentariums für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* anhand der Anforderungen aus Kapitel 2.6 statt. Hierzu wird eine präzise Erläuterung je Anforderung gegeben, inwiefern sie durch einzelne Bestandteile oder das erzielte Zusammenwirken erfüllt wird. Einen Überblick über die Anforderungen und ihren Bezug zum Instrumentarium zeigt Bild 5-19.

**A1) Ganzheitlichkeit:** Das Fundament dieser Arbeit und der lösungsmusterbasierten Produktentstehung ist ein gemeinsames Begriffsverständnis. Hierzu wurden unterschiedliche Interpretationen des Lösungsmusterbegriffs analysiert und in eine für die Produktentstehung *allgemeingültige Begriffsdefinition* überführt. Insgesamt wurde das *Paradigma der Lösungsmuster in der Produktentstehung* anhand von Beispielen ausführlich beschrieben (vgl. Kap. 4.2). Neben den vorrangig an der Produktentwicklung beteiligten Disziplinen fanden auch angrenzende Bereiche wie die Organisationsplanung, strategische Produktplanung sowie Produktionssystemplanung Berücksichtigung.

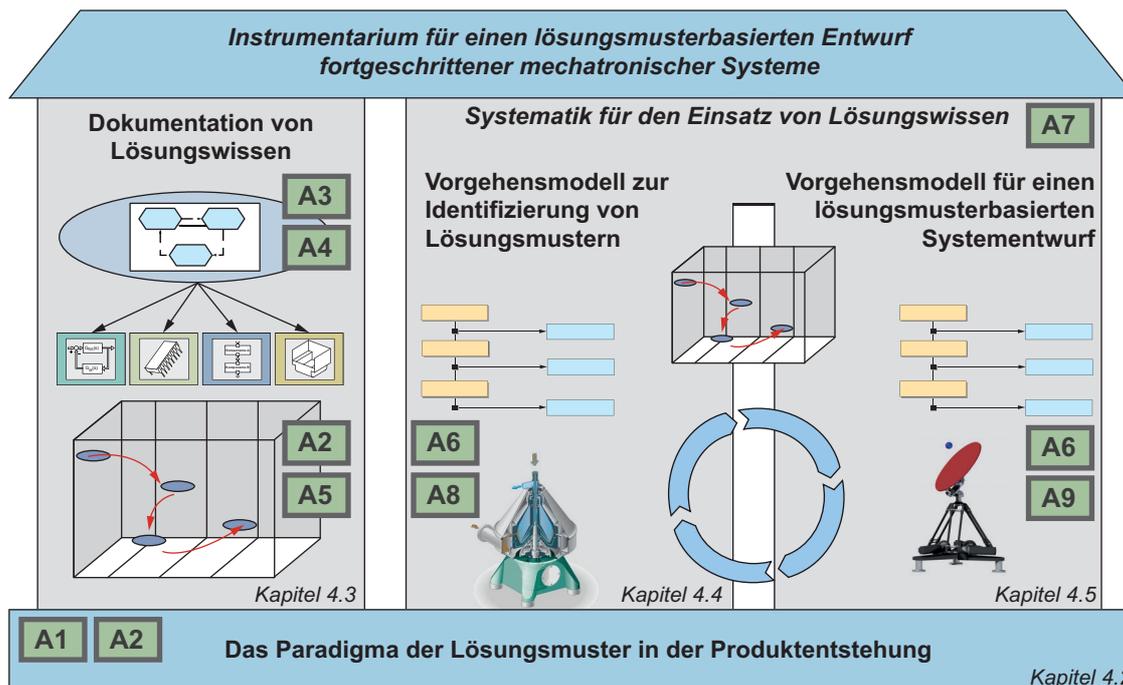


Bild 5-19: Erfüllung der Anforderungen durch das Instrumentarium

**A2) Charakterisierung von Lösungsmustern:** Lösungsmuster, der Detaillierungsgrad der Wissensinhalte in den Lösungsmustern sowie wechselseitige Abhängigkeiten variieren stark. Die charakteristischen Eigenschaften von Lösungsmustern transparent darzustellen, ist eine Herausforderung. Das Instrumentarium beschreibt hierzu zunächst auf einer abstrakten Ebene das *Wirkgefüge von Lösungsmustern* in der Produktentstehung (vgl. Kap. 4.2.3). Lösungsmuster für den Systementwurf werden darüber hinaus ausführlich entlang *dreier Dimensionen an eingängigen Beispielen charakterisiert* (vgl. Kap. 4.3.3). Mit Hilfe der Dimensionen lassen sich auch die Zusammenhänge mit fachdisziplinspezifischen Lösungsmustern abbilden (vgl. Kap. 4.3.3.4).

**A3) Einheitliche Strukturierung in Anlehnung an ALEXANDER:** Das entwickelte Instrumentarium beinhaltet eine *einheitliche Strukturierung* zur Dokumentation und Wiederverwendung von Lösungswissen. Diese erfüllt die grundlegenden Kategorien nach ALEXANDER zur Beschreibung von Problem-Lösungskombinationen (vgl. Kap. 4.3.1). Im Zuge dieser Arbeit wurden charakteristische Lösungsmuster, sowohl fachdisziplinenorientiert als auch multidisziplinär, gemäß der erarbeiteten Strukturierung aufgenommen (vgl. Kap. 4.3.2 und 5.1). Sie verdeutlichen die Anwendbarkeit der Strukturierung und der zielgerichteten Informationsbereitstellung während des Systementwurfs.

**A4) Interdisziplinarität und Verständlichkeit:** Am Systementwurf intelligenter mechatronischer Systeme ist neben dem leitenden Systems Engineer auch ein fachdisziplinübergreifendes Team von Experten beteiligt. Der Dialog über bestimmte Sachverhalte – Systemkonzept und notwendiges Lösungswissen – gelingt nur über eine gemeinsame Terminologie. Das Instrumentarium stützt sich in diesem Zusammenhang auf die *Spezifikationstechnik CONSENS*. Diese wird im Hinblick auf die Besonderheiten der

adressierten Systeme (wie z.B. zunehmende verteilte Koordination untereinander) um ein *Sequenzdiagramm* ergänzt. Es werden sowohl *Beispiele für Lösungsmuster* als auch deren erfolgreiche Anwendung in der Entwicklung verteilter mechatronischer Systeme gezeigt (vgl. Kap. 4.3.2.1 und 4.5.5).

**A5) Durchgängige Dokumentation:** Durchgängigkeit ist eine der zentralen Forderungen in der Entwicklung mechatronischer Systeme. Diesen Herausforderungen muss sich auch ein zugehöriges Wissensmanagement stellen. Der entwickelte *multidimensionale Wissensraum* schafft hierfür eine zentrale Grundlage. Übergänge z.B. zu Wirkprinzipien des Maschinenbaus und wiederverwendbaren SPS-Funktionsbausteinen lassen sich transparent darstellen und zu einem zusammenhängenden Wirkgefüge verbinden. Ein generisches Vorgehen verdeutlicht hierbei das *Zusammenspiel mit den Entwurfsebenen* (vgl. Kap. 4.3.3.5). Die *Validierung anhand zweier Anwendungsbeispiele* bestätigt dabei zweierlei Dinge. Zum einen wird gezeigt, wie Lösungsmuster Schritt für Schritt abstrahiert und zusammenhängend dokumentiert werden können (vgl. Kap. 5.1). Zum anderen ermöglicht die durchgängige Dokumentation eine fortlaufende Verwendung im Entwurf mit der anschließenden transparenten Darstellung von Schnittstellen zu den Fachdisziplinen (vgl. Kap. 5.2).

**A6) Systematische Vorgehensweise:** Die entwickelten *Vorgehensmodelle* ermöglichen eine systematische Vorgehensweise. Sie beschreiben die durchzuführenden Aufgaben und empfehlen notwendige Hilfsmittel, die größtenteils für das Instrumentarium entwickelt wurden. Beide Vorgehensmodelle unterteilen sich jeweils in vier wesentliche Phasen (vgl. Kap. 4.4 und 4.5). Die Anwendbarkeit der Vorgehensmodelle wurde anhand von zwei Beispielen – Selbstentleerer Tellerseparator sowie ein kooperierendes Robotersystem – belegt (vgl. Kap. 5.1 und 5.2).

**A7) Geschlossener SECI-Kreislauf:** Ein Wissensmanagementansatz ist dann erfolgreich, wenn er gemäß den SECI-Phasen einen in sich geschlossenen Kreislauf bildet. Die im Rahmen dieser Arbeit entstandene *Systematik* erfüllt diese Anforderungen in Gänze. Die Externalisierung wird unterstützt durch das *Vorgehensmodell zur Identifizierung von Lösungsmustern*. Die Kombination und Internalisierung von Lösungswissen z.T. fachfremder Disziplinen ist aufgrund des entwickelten *multidimensionalen Wissensraums* jedem Entwickler möglich. Die Sozialisation wird vorrangig durch die Anwendung des *Vorgehensmodells für einen lösungsmusterbasierten Systementwurf* adressiert, wenn nämlich der Systems Engineer gemeinsam mit den Entwicklern potentielle Lösungsmuster diskutiert und verwendet. Ferner haben Lösungsmuster im Gegensatz zu vielen anderen Ansätzen wie z.B. Steckbriefen eine verminderte Gefahr, in Vergessenheit zu geraten. Die *Lösungsmuster dieses Instrumentariums sind so strukturiert*, dass sie *proaktiv* auf Basis von Anforderungen und Funktionen in die Entwicklung eingebunden werden können (vgl. Kap. 4.3.1, 4.3.3.4, 5.1 und 5.2).

**A8) Abstraktion und Generalisierung:** In der Aufbereitung von Lösungswissen ist der Grad der Abstraktion ein wesentlicher Schlüssel zum Innovationserfolg. Der Grad ist

entscheidend für die Übertragbarkeit des dokumentierten Lösungswissens auf ähnliche Problemstellungen und somit möglicher Treiber für Kreativität und neue Innovationen. Ist das Wissen in einem Lösungsmuster zu spezialisiert, ist der Anwendungskontext sehr eingeschränkt. Ist das Wissen zu abstrakt, zieht die Auswahl und Anwendung eine langwierige Konkretisierung nach sich. Das *Vorgehensmodell zur Identifizierung von Lösungsmustern* und die *Dimensionen im multidimensionalen Wissensraum* begleiten diesen Prozess, beginnend bei einer detaillierten Lösung (vgl. Kap. 4.4 und 4.3.3.4). Die lösungsneutrale Spezifikation des Problems in der *Strukturierung von Lösungsmustern* durch Funktionen, unterstützt diese Besonderheiten zusätzlich (vgl. Kap. 4.3.1). Auf diese Weise lassen sich Dritte unterschiedlicher Unternehmen bei der individuell gewünschten und erforderlichen Abstraktion ihres Lösungswissens begleiten (vgl. z.B. Kap. 5.1).

**A9) Orientierung an etablierten Standards im Systementwurf:** Sowohl in der klassischen Konstruktionslehre, ihren Fortführungen in der VDI2206 als auch im Systems Engineering beginnt eine Produktentwicklung mit der Aufnahme von Anforderungen und der anschließenden funktionalen Beschreibung des zu entwickelnden Systems. Es folgt die Synthese von Architektur (z.B. Wirkstruktur) und Logik (z.B. Verhalten) unter Einbeziehung von Lösungswissen. Diese Denkweise greift das Instrumentarium, speziell die *einheitliche Strukturierung der Lösungsmuster* und das *Vorgehensmodell für einen lösungsmusterbasierten Systementwurf*, auf (vgl. Kap. 4.3.1 und 4.5). Der Fokus liegt dabei auf der Schnittstelle zwischen der Zielbestimmung und der Synthese – gleichnamige Phasen in dem angesprochenen Vorgehen. Durch die im Wissensraum dokumentierten Zusammenhänge hinsichtlich der *Spezialisierung von Lösungswissen* werden auch Iterationen in der fortlaufenden Konkretisierung einer Entwicklung berücksichtigt (vgl. Kapitel 5.2).

Das vorgestellte *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* erfüllt damit alle Anforderungen in vollem Umfang. Das Instrumentarium ermöglicht es, fachdisziplinübergreifend relevantes Lösungswissen für den Systementwurf *zu identifizieren, zu dokumentieren* und bei Neuentwicklungen *anzuwenden*. Es wurde mit Erfolg an den Anwendungsbeispielen „Separator“ und „Kooperierende Deltaroboter“ validiert.

## 6 Zusammenfassung und Ausblick

Innovationen für den modernen Maschinenbau beruhen schon heute auf dem engen Zusammenwirken von Mechanik, Elektrik/Elektronik, Regelungstechnik und Software. Hierfür steht der Begriff Mechatronik. Wesentlicher Funktions- und somit auch Komplexitätstreiber ist die voranschreitende Entwicklung von Technologien der Informations- und Kommunikationsbranche. Perspektivische Ziele sind **fortgeschrittene mechatronische Systeme**, die z.T. global untereinander vernetzt sind, eigenständig agieren und ihr Systemverhalten situationsspezifisch optimieren können. Dadurch erweitert sich die zu betrachtende Systemgrenze während der Entwicklung stetig. Aufgrund der steigenden Interdisziplinarität und der Involvierung unterschiedlichster Stakeholder muss das System in seiner Gesamtheit mehr denn je in den Mittelpunkt der Entwicklung gerückt werden.

Auf dem Weg zu derart fortgeschrittenen mechatronischen Systemen wird die **Dokumentation und Wiederverwendung von etabliertem Lösungswissen** als ein entscheidender Erfolgsgarant gesehen. Es steigt die Forderung nach einem Wissensmanagement, das einerseits innerhalb einzelner Disziplinen in der Produktentstehung Anklang findet und sich andererseits zu einem ganzheitlichen Wirkgefüge vereinen lässt. Ein erfolgversprechender Ansatz sind **Lösungsmuster**, die abstrahierte Problem-Lösungspaare darstellen. Um zeit- und kostenintensive Iterationen zu einem späten Zeitpunkt einer Entwicklung zu vermeiden, liegt ein Schwerpunkt zweifelsohne auf der frühzeitigen Berücksichtigung relevanten Lösungswissens im Rahmen des Systementwurfs. Hierbei mangelt es jedoch aktuell an einer geeigneten Verzahnung von etablierten Vorgehensweisen wie der VDI2206 oder dem Model-Based Systems Engineering mit Wiederverwendung von Lösungswissen in Form von Lösungsmustern. Die drei wesentlichen **Herausforderungen** liegen in einem *ungeklärten und für die Produktentstehung allgemeingültigen Lösungsmusterbegriff*, in einer *fehlenden durchgängigen Dokumentation von interdisziplinärem Lösungswissen für den Systementwurf* und in einer *fehlenden Entwicklungssystematik*. Letztere muss die Fragen nach der Identifizierung von Lösungswissen und ihrer Anwendung im Systementwurf beantworten.

Um diesen Herausforderungen zu begegnen, müssen drei **Handlungsfelder** erschlossen werden. Es muss das *Paradigma der Lösungsmuster in der Produktentstehung* *umfänglich beschrieben* werden. Ferner werden *Techniken für eine einheitliche Strukturierung von Lösungsmustern für den Systementwurf sowie deren Charakterisierung* benötigt. Es muss gezeigt werden, wie diese mit fachdisziplinspezifischen Ansätzen zusammenhängen und wie die fortlaufende Einbindung in die Produktentwicklung umgesetzt werden kann. Des Weiteren sind *Vorgehensmodelle notwendig, die Dritte in die Lage versetzen, Lösungsmuster zu identifizieren und anzuwenden*.

Im Rahmen dieser Arbeit wurden etablierte Ansätze zur Beschreibung von Lösungsmustern in der Produktentstehung wie auch im Systems Engineering, Lösungsmuster für

den Systementwurf (fachdisziplinübergreifende sowie -spezifische LM als auch Techniken für die Wissensrepräsentation) und Systematiken für den Einsatz von Lösungsmustern untersucht. Die Analysen liefern jedoch nur partiell Ergebnisse, die genutzt und erweitert werden können. Ein ganzheitliches Instrumentarium, das alle Handlungsfelder abdeckt, existiert nicht.

Einen umfangreichen Überblick über Lösungsmuster in Form einer *Hierarchie liefert CLOUTIER*, jedoch mangelt es an eingängigen Beispielen. Weitere Ansätze wie die *Universal Design Theory nach GRABOWSKI*, das *SE Entwurfsmuster-Metamodell nach PFISTER* oder die *SE-Mustersprache nach SIMPSON* sind sehr generisch und abstrakt; sie liefern wenn überhaupt nur erste Gliederungsmöglichkeiten. Im Hinblick auf Lösungswissen für den Systementwurf wurden dreierlei Themenfelder analysiert. Von den Techniken für die Wissensrepräsentation erfüllt die *Spezifikationstechnik CONSENS* wichtige Anforderungen, was ihre Verwendung insbesondere hinsichtlich einer barrierefreien Kommunikation und Kooperation nahelegt. Sie ist sowohl als Sprache zur Beschreibung des Lösungswissens als auch zur Erstellung des Systemmodells im Rahmen der Entwicklung geeignet. Von den analysierten fachdisziplinübergreifenden Lösungsmustern bieten die Arbeiten *nach DUMITRESCU für s.o. Systeme* gute Anhaltspunkte für eine einheitliche Strukturierung; die Charakterisierung von und Verzahnung mit disziplinspezifischen Ansätzen findet nur unzureichend statt. Bestehende fachdisziplinspezifische Lösungsmuster wie die *Koordinationsmuster der Softwaretechnik* oder die Arbeiten *nach SANZ/ZALEWSKI für die Regelungstechnik* sind etabliert und sollten im Hinblick auf die Durchgängigkeit berücksichtigt werden. *SUHM* liefert darüber hinaus noch übertragbare Gliederungsansätze anhand von *Lösungsmustern für den Maschinenbau*. Die drei untersuchten Systematiken für den Einsatz von Lösungsmustern erfüllen die gestellten Anforderungen nur partiell. Eine retrospektive sehr generische *Vorgehensweise zur Identifizierung von Lösungsmustern* liefert lediglich *KALAWSKY*. Ansätze für einen *musterbasierten Entwurf beschreiben DUMITRESCU und GAUSEMEIER*, jedoch ohne detaillierte Berücksichtigung einer durchgängigen und verzahnten Dokumentation von Lösungswissen entlang einer Produktentwicklung. Aus diesen Gründen besteht **Handlungsbedarf** für ein ganzheitliches *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme*.

Das erarbeitete Instrumentarium greift einige Überlegungen der untersuchten Ansätze auf, überträgt sie auf die Gegebenheiten der modernen Produktentstehung, erweitert und ergänzt sie um neu entwickelte Hilfsmittel. Das **Ergebnis** setzt sich aus den folgenden vier übergeordneten Bestandteilen zusammen, wobei die beiden letztgenannten eine eigenständige Systematik ergeben:

- eine umfangreiche **Beschreibung des Paradigmas der Lösungsmustern in der Produktentstehung**, die neben einer allgemeingültigen Begriffsdefinition auch das PE-weite Wirkgefüge von Lösungsmustern anhand von Beispielen darlegt,

- eine Technik zur **Dokumentation von Lösungswissen für den Systementwurf**, die unter Verwendung der Spezifikationstechnik CONSENS eine einheitliche Strukturierung von Lösungsmustern enthält und in einem multidimensionalen Wissensraum Lösungsmuster entlang dreier Dimensionen in Zusammenhang zueinander setzt,
- ein **Vorgehensmodell zur Identifizierung von Lösungsmustern**, das auf bereits realisierte Systeme aufsetzt und dabei die durchzuführenden Aufgaben in vier zentrale Phasen einteilt; notwendige Hilfsmittel, wie z.B. die Technik zur Dokumentation von Lösungswissen werden integriert,
- ein **Vorgehensmodell für einen lösungsmusterbasierten Systementwurf**, das einen Systems Engineer in die Lage versetzt, gemeinsam mit einem interdisziplinären Entwicklungsteam sukzessive unter Verwendung etablierten Lösungswissens ein Systemkonzept zu erstellen.

Die **Validierung** des Instrumentariums erfolgte anhand zweier Anwendungsbeispiele. Bei dem Separator handelt es sich um ein Industrieerzeugnis wohingegen die kooperierenden Deltaroboter Gegenstand eines Forschungsprojekts war. Im Rahmen der Validierung wurden die entwickelten Vorgehensmodelle vollständig durchlaufen und die speziell für das Instrumentarium entwickelten Hilfsmittel verwendet. Auf diese Weise wurden die einzelnen Phasen der gesamten Systematik mit ihren jeweiligen Resultaten vorgestellt. Die Anwendbarkeit der Strukturierung von Lösungsmustern sowie des multidimensionalen Wissensraums wurden zudem an charakteristischen Beispielmustern (multidisziplinär als auch diszipliniert) belegt. Die Validierung zeigt, dass das entwickelte *Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme* die gestellten Anforderungen in vollem Umfang erfüllt.

Im Hinblick auf den Einsatz von Lösungsmustern in der Entwicklung intelligenter technischer Systeme besteht **weiterer Forschungsbedarf**. Zukünftige Arbeiten müssen die Verzahnung von Lösungswissen für den Systementwurf aus den beteiligten Disziplinen verstärkt vorantreiben. Das entwickelte Instrumentarium legt diesbezüglich einen Schwerpunkt auf die frühen Phasen des modellbasierten Systementwurfs mit der Vorbereitung des Übergangs in den disziplinspezifischen Entwurf. Weiterführende Arbeiten könnten großen Nutzen stiften, wenn sie z.B. die durchgängige Verzahnung der abstrakten „Disziplinübergreifenden System-Ebene“ und der „Disziplinintegrierenden Aspektspezifischen-Ebene“ in den Vordergrund rücken. Übergeordnetes Ziel zukünftiger Forschungsarbeiten mit einem **längeren Horizont** sollte die lösungsmusterbasierte Entstehung intelligenter technischer Systeme sein. Hierzu stellt das im Rahmen dieser Arbeit entwickelte Instrumentarium, speziell das verdeutlichte Wirkgefüge von Lösungsmustern in der Produktentstehung, den Startpunkt dar. Dieses entwickelte Fundament sollte aufgegriffen, vertieft und zu einer übergeordneten Mustersprache weiterentwickelt werden.



## 7 Abkürzungsverzeichnis

bzw.	beziehungsweise
CAD	Computer Aided Design
CPS	Cyber-physisches System
CO	Controller
DLR	Deutsches Zentrum für Luft- und Raumfahrt
el.	elektrisch
HF	Handlungsfeld
HW	Hardware
i.d.R.	in der Regel
INCOSE	International Council on Systems Engineering
KO	Kognitiver Operator
LE	Lösungselement
LM	Lösungsmuster
MBSE	Model-Based Systems Engineering
o.A.	oder Andere
OCM	Operator-Controller-Modul
OMG	Object Management Group
PE	Produktentstehung
RO	Reflektorischer Operator
ROI	Return on Invest
SE	Systems Engineering
s.o.	selbstoptimierend
sog.	sogenannt
SPS	Speicherprogrammierbare Steuerung
SW	Software
u.Ä.	und Ähnliche
UDT	Universal Design Theory

VDI      Verein Deutscher Ingenieure

vgl.      vergleiche

z.B.      zum Beispiel

z.T.      zum Teil

## 8 Literaturverzeichnis

- [ADG+09] ADELT, P.; DONOTH, J.; GAUSEMEIER, J.; GEISLER, J.; HENKLER, S.; KAHL, S.; KLÖPPER, B.; KRUPP, A.; MÜNCH, E.; OBERTHÜR, S.; PAIZ, C.; PORRMANN, M.; RADKOWSKI, R.; ROMAUS, C.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITKOWSKI, U.; WITTING, K.; ZNAMENSHCHYKOV, O.: Selbstoptimierende Systeme des Maschinenbaus – Definition, Anwendungen, Konzepte. HNI-Verlagsschriftenreihe, Band 234, 2009
- [AG12] ALBERS, A.; GAUSEMEIER, J.: Von der fachdisziplinorientierten Produktentwicklung zu vorausschauenden und systemorientierten Produktentstehung. In ANDERL, R.; EIGNER, M.; SENDLER, U.; STARK, R. (Hrsg.): Smart Engineering – Interdisziplinäre Produktentstehung. Acatech DISKUSSION, 2012
- [AIS+95] ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M.; JACOBSON, M.; FIKSDAHLKING, I.; ANGEL, S.; CZECH, H.(Hrsg.): Eine Muster-Sprache – Städte, Gebäude, Konstruktion. Löcker Verlag, Wien, 1995
- [AIS+77] ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M.; JACOBSON, M.; FIKSDAHLKING, I.; ANGEL, S.: A Pattern Language. Oxford University Press, 1st Edition, New York, USA, 1977
- [Alt12] ALT, O.: Modell-basierte Systementwicklung mit SysML – In der Praxis. Carl Hanser Verlag, München, 2012
- [Amb98] AMBLER, S. W.: An Introduction to Process Patterns. AmbySoft Inc. White Paper, 1998
- [AT11] ALBERS, A.; TURKI, T.: Abbildung von Erfahrungswissen zur Unterstützung des Entwurfsprozesses in der Mikrotechnik. In: Mikrosystemtechnik-Kongress, 10.-12. Oktober, Darmstadt, 2011
- [Bar98] BARTER, R. H.: A Systems Engineering Pattern Language. In: Proceedings of 8th Annual International Symposium on the International Council on Systems Engineering, Vancouver, BC, Juli, 1998
- [BBB+12] BECKER, S.; BRENNER, C.; BRINK, C.; DZIWOK, S.; HEINZEMANN, C.; LÖFFLER, R.; POHLMANN, U.; SCHÄFER, W.; SUCK, J.; SUDMANN, O.: The Mechatronic-UML DesignMethod – Process, Syntax, and Semantics. Technischer Bericht, tr-ri-12-326. Vers. 0.3, Lehrstuhl für Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn, 2012
- [BBK+09] BALZERT, H.; BALZERT, H.; KOSCHKE, R.; LÄMMEL, U.; LIGGESMEYER, P.; QUANTE, J.: Lehrbuch der Software-Technik. Spektrum, Heidelberg, 3. Auflage, 2009
- [BC87] BECK, K.; CUNNINGHAM, W.: Using Pattern Languages for Object-Oriented Programs. Technical Report No. CR-87-43, Submitted to the OOPSLA-87 Workshop on the Specification and Design for Object-Oriented Programming, 17. September, 1987
- [BD93] BRÖHLE, A.-P.; DRÖSCHL, W.: Das V-Modell – Der Standard für die Softwareentwicklung mit Praxisleitfaden. Oldenbourg Verlag, München, 1993
- [BGH+96] BLECK, A.; GOEDECKE, M.; HUSS, S.; WALDSCHMIDT, K. (Hrsg.): Praktikum des modernen VLSI-Entwurfs. B.G. Teubner, 1996
- [BGP+14] BAUER, F.; GAUSEMEIER, J.; PRUSCHEK, P.; REHAGE, G.: Arbeitsvorbereitung 4.0 – Cloud-basierte Nutzung virtueller Werkzeugmaschinen. Wt Werkstattstechnik online, Ausgabe 2-2013 Sonderheft Industrie 4.0, S. 118-123
- [BHL07-ol] BRAUN, S.; HELLENBRAND, D.; LINDEMANN, U.: Kostentransparenz in der Mechatronik – Eine Studie über Komplexitäts- und Kostentreiber mechatronischer Produkte, unter: <http://www.shaker.de/de/content/catalogue/index.asp?lang=&ID=8&ISBN=OND-00000-0000004>

- [Bic00] BICHLMAIER, C.: Methoden zur flexiblen Gestaltung von integrierten Entwicklungsprozessen. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Herbertz Utz Verlag, München, 2000
- [BK06] BEIERLE, C.; KERN-ISBERNER, G.: Methoden wissensbasierter Systeme: Grundlagen – Algorithmen – Anwendungen. Vieweg Verlag, Berlin, 2006
- [BKF+14] BARBIERI, G.; KERNSCHMIDT, K.; FANZUZZI, C.; VOGEL-HEUSER, B.: A SysML based design pattern for the high-level development of mechatronic systems to enhance reusability. In: Proceedings of Preprints of the 19th World Congress of the international federation of automatic control, 24-29 August, Cape Town, South Africa, 2014.
- [Buc03] BUCHER, P. E.: Integrated Technology Roadmapping – Design and implementation for technology-based multinational enterprises. Dissertation, Eidgenössische Technische Hochschule Zürich, 2003
- [Bue04] BÜHNER, R.: Betriebswirtschaftliche Organisationslehre. Oldenbourg Wissenschaftsverlag, München, 10. Auflage, 2004
- [Bul94] BULLINGER, H.-J.: Einführung in das Technologiemanagement – Modelle, Methoden, Praxisbeispiele. Teubner Verlag, Stuttgart, 1994
- [CD06] CLOUTIER, R.; VERMA, D.: Applying Pattern Concepts to Systems (Enterprise) Architecture. Journal of Enterprise Architecture, Volume 2, Number 2, 2006
- [Clo06] CLOUTIER, R.: Applicability of patterns to architecting complex systems. Dissertation, Stevens Institute of Technology, New York, 2006
- [Cop91] COPLIEN, J.: Advanced C++ Programming Styles and Idioms. Addison-Wesley, 1991
- [Dei09] DEIGENDESCH, T.: Kreativität in der Produktentwicklung und Muster als methodisches Hilfsmittel. Dissertation, Fakultät für Maschinenbau, Karlsruher Institut für Technologie (KIT), Forschungsbericht Band 41, Institut für Produktentwicklung (IPEK), Karlsruhe, 2009
- [DDG+14] DOROCIAC, R.; DUMITRESCU, R.; GAUSEMEIER, J.; IWANEK, P.: Specification Technique CONSENS for the Description of Self-optimizing Systems. In: GAUSEMEIER, J.; RAMMIG, F.; SCHÄFER, W. (Hrsg.): Design Methodology for Intelligent Technical Systems, Springer, Berlin, 2014
- [DHT12] DZIWOK, S.; HEINZEMANN, C.; TICHY, M.: Real-Time Coordination Patterns for Advanced Mechatronic Systems. In: Coordination, 2012, S. 166-180
- [Dud14-ol] DUDEN: Rechtschreibung. Unter: <http://www.duden.de/rechtschreibung>, 23.11.2014
- [Dum10] DUMITRESCU, R.: Entwicklungssystematik zur Integration kognitiver Funktionen in fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 286, Paderborn, 2010
- [Ehr03] EHRENSPIEGEL, K.: Integrierte Produktentwicklung. Carl Hanser Verlag, München, 2003
- [FHV92] FRITZON, P.; HERBER, J.; VIKLUND, L.: The implementation of ObjectMath - a high-level programming environment for scientific computing. In KASTENS, U.; PFAHLER, P.: Compiler Construction – 4th International Conference (CC92), Volume 641 of Lecture Notes in Computer Science, Springer Verlag, Berlin, 1992, S. 312-318
- [FMS12] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A practical guide to SysML – The systems modeling language. Morgan Kaufmann, Waltham, 2. Auflage, 2012
- [Foe72] FÖLLINGER, O.: Regelungstechnik – Einführung in die Methoden und ihre Anwendung. Hüthig Verlag, Heidelberg, 1. Auflage, 1972
- [Foe13] FÖLLINGER, O.: Regelungstechnik – Einführung in die Methoden und ihre Anwendung. Hüthig Verlag, Heidelberg, 11. Auflage, 2013

- [For07] FORBIG, P.: Objektorientierte Softwareentwicklung mit UML. Carl Hanser Verlag, München, 3. Auflage, 2007
- [Fra13] FRANKE, J. (Hrsg.): Räumliche elektronische Baugruppen (3D-MID) - Werkstoffe, Herstellung, Montage und Anwendungen für spritzgegossene Schaltungsträger. Carl Hanser Verlag, München, 2013
- [Fri04] FRITZON, P.: Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. IEEE Press, Wiley-Interscience, 2004
- [Gau10] GAUSEMEIER, J. (Hrsg.): Frühzeitige Zuverlässigkeitsanalyse mechatronischer Systeme. Carl Hanser Verlag, München, 2010
- [GDS13] GAUSEMEIER, J.; DUMITRESCU, R.; STEFFEN, D. (Hrsg.): Systems Engineering in der industriellen Praxis. Studie, wentker druck GmbH, 2013
- [GEK01] GAUSEMEIER, J.; EBBESMEYER, P.; KALLMEYER, F.: Produktinnovation – Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag, München, Wien, 2001
- [GF06] GAUSEMEIER, J.; FELDMANN, K.: Integrative Entwicklung räumlicher elektronischer Baugruppen. Carl Hanser Verlag, München, 2006
- [GFD+08a] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus (Teil1). Konstruktion, 7/8 – 2008, Fachaufsatz Mechatronik, VDI-Verlag, Berlin, 2008, S. 59-66
- [GFD+08b] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus (Teil2). Konstruktion, 9 – 2008, Fachaufsatz Mechatronik, VDI-Verlag, Berlin, 2008, S. 91-99
- [GFC13] GASSMANN, O.; FRANKENBERGER, K.; CSIK, M.: Geschäftsmodelle entwickeln – 55 innovative Konzepte mit dem St. Galler Business Model Navigator. Carl Hanser Verlag, München, 2013
- [GG98] GASKA, M. T.; GAUSE, D. C.: An Approach for Cross-Discipline Requirements Engineering Process Patterns. In: Proceedings of 3rd International Conference on Requirements Engineering (ICRE '98), Putting Requirements Engineering to Practice, 6.-10. April, Colorado Springs, USA, 1998
- [GHJ+94] GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley Verlag, München, 1994
- [GHK+06] GAUSEMEIER, J.; HAHN, A.; KESPOHL, H.D.; SEIFERT, L.: Vernetzte Produktentwicklung – Der erfolgreiche Weg zum Global Engineering Networking. Carl Hanser Verlag, München, 2006
- [GJD+13] GAUKSTERN, T.; JÜRGENHAKE, C.; DUMITRESCU, R.; GAUSEMEIER, J.: Eine Methodik für den wissensbasierten Entwurf von dreidimensionalen spritzgegossenen Schaltungsträgern (MID). In: 9. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagschriftenreihe, Band 310, 18.-19. April, Paderborn, 2013
- [GL00] GRABOWSKI, H.; LOSSACK, R.: The Axiomatic Approach in the Universal Design Theory. In: Proceedings of the First International Conference on Axiomatic Design (ICAD2000), Cambridge, 21.-23. Juni, Cambridge, USA, 2000
- [GL03] GRABOWSKI, H.; LEUTSCH, G. R.: Integrierte Produkt- und Produktionsmodelle als Grundlage für eine wissensbasierte Produktentwicklung. In: NAGEL, M.; WESTFECHTEL, B.: Modelle, Werkzeuge und Infrastrukturen zur Unterstützung von Entwicklungsprozessen, Wiley-VCH Verlag, Weinheim, 2003, S. 45-48
- [GLL12] GAUSEMEIER, J.; LANZA, G.; LINDEMANN, U. (Hrsg.): Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung. Carl Hanser Verlag, München, 2012

- [GP14] GAUSEMEIER, J.; PLASS, C. (Hrsg.): Zukunftsorientierte Unternehmensgestaltung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. Carl Hanser Verlag, München, 2. Auflage, 2014
- [GRS03] GÖRZ, G.; ROLLINGER, C.-R.; SCHNEEBERGER, J. (Hrsg.): Handbuch der Künstlichen Intelligenz. Oldenbourg Wissenschaftsverlag, München, 4. Auflage, 2003
- [GSA+11] GAUSEMEIER, J.; SCHÄFER, W.; ANACKER, H.; BAUER, F.; DZIWOK, S.: Einsatz semantischer Technologien im Entwurf mechatronischer Systeme. In: 8. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 294, 19.-20. Mai, Paderborn, 2011
- [GTD13] GAUSEMEIER, J.; TSCHIRNER, C.; DUMITRESCU, R.: Der Weg zu Intelligenten Technischen Systemen. Industrie Management, GITO Verlag, 1/2013, S. 49-52
- [GTS14] GAUSEMEIER, J.; TRÄCHTLER, A.; SCHÄFER, W. (Hrsg.): Semantische Technologien im Entwurf mechatronischer Systeme – Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten. Carl Hanser Verlag, München, 2014
- [Has05] HASKINS, C.: Application of Patterns and Pattern Languages to Systems Engineering. In: Proceedings of 15th Annual International Symposium of the International Council on Systems Engineering, Rochester, New York, Juli, 2005
- [Hit07] HITCHINS, D. K.: Systems engineering – A 21st century systems methodology. John Wiley, West Sussex, England, 2007
- [HK01] HAN, J.; KAMBER, M.: Data mining concepts and techniques. Morgan Kaufmann Publisher, 2001
- [HO03] HESTERMEYER, T.; OBERSCHELP, O.: Selbstoptimierende Fahrzeugregelung – Verhaltensbasierte Adaption. In: GAUSEMEIER, J.; LÜCKEL, J.; WALLASCHECK, J.: 1. Paderborner Workshop Intelligente mechatronische Systeme. HNI-Verlagsschriftenreihe, Band 122, 2003
- [Hoo05] HOOD, C.: Optimieren von Requirements Management & Engineering. Springer-Verlag, Berlin, 2005
- [Hoo07] HOOD, C.; WIEDEMANN, S.; FICHTINGER, S.; PAUTZ, U.: Requirements Management – The Interface Between Requirements Development and All Other Systems Engineering Processes. Springer Verlag, Berlin, 2007
- [HTF96] HARASHIMA, F.; TOMIZUKA, M.; FUKUDA, T.: Mechatronics – „What Is It, Why and How?“ An Editorial. In: IEEE/ASME Transactions on Mechatronics, Volume 1, Nr. 1, 1996
- [HWF+12] HABERFELLNER, R.; WECK DE, O. L.; FRICKE, E.; VÖSSNER, S.: Systems Engineering – Grundlagen und Anwendung. Orell Füssli, Zürich, 2012
- [Ise08] ISERMANN, R.: Mechatronische Systeme – Grundlagen. Springer Verlag, Berlin, 2008
- [JRH+04] JECKLE, M.; RUPP, C.; HAHN, J.; ZENGLER, B.; QUEINS, S.: UML 2 – Glasklar. Carl Hanser Verlag, München, 2004
- [Kah13] KAHL, S.: Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 308, Paderborn 2013
- [Kai13] KAISER, L.: Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 327, Paderborn 2013
- [Kas03] KASTOR, M.: Psychologie der Individualität. Verlag Königshausen & Neumann GmbH, Würzburg, 2003
- [KCH+90] KANG, K.; COHEN, S.; HESS, J.; NOWAK, W.; PETERSON, S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study, Technical report CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, 1990

- [Kel00] KELLER, H. B.: Maschinelle Intelligenz – Grundlagen, Lernverfahren, Bausteine intelligenter Systeme. Vieweg Verlag, Braunschweig, 2000
- [KJT+13] KALAWSKY, R. S.; JOANNOU, D.; TIAN, Y.; FAYOUMI, A.: Using architecture patterns to architect and analyze systems of systems. In Proceedings of Conference on Systems Engineering Research (CSER'13), Atlanta, USA, 19.-22. März, 2013
- [KK98] KOLLER, R.; KASTRUP, N.: Prinziplösungen zur Konstruktion technischer Produkte. Springer Verlag, Berlin, 2. Auflage, 1998
- [Kle10] KLEIN, B.: FEM – Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau. Springer Verlag, Berlin, 8. Auflage, 2010
- [Kop97] KOPETZ, H.: Real-Time Systems – Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, Dordrecht, 1997
- [KT05] KUHLIN, B.; THIELMANN, H.: Real-Time Enterprise in der Praxis: Fakten und Ausblick. Springer Verlag, Berlin, 2005
- [LBW93] LULLIES, V.; BOLLINGER, H.; WELTZ, F.: Wissenslogistik. Campus Verlag, Frankfurt, 1993
- [Len02] LENZEN, M.: Natürliche und künstliche Intelligenz – Einführung in die Kognitionswissenschaft. Campus Verlag, Frankfurt am Main, 2002
- [LJH+13] LEIMEISTER, J. M.; JANZEN, A.; HOFFMANN, A.; HOFFMANN, H.: Anforderungsmuster im Requirements Engineering. Working Paper No. 2., University of Kassel, Information Systems Sciences, 2013
- [Lug06-ol] LUGER, S.: Goldfire Innovator – Ein umfassendes Innovationswerkzeug. 2006. Unter: [http://www.triz-online.de/uploads/media/artikel\\_2006\\_04\\_01.pdf](http://www.triz-online.de/uploads/media/artikel_2006_04_01.pdf), 30. Januar 2014
- [Mau11] MAURER, M.: Knowledge Transfer Applying the structural Complexity Management Approach. In: Journal of Information Retrieval and Knowledge Management, Vol. 1, 2011
- [MBD+05] MABEN, T. v. D.; BACHER, H.-V.; DÖRR, J.; GEISBERGER, E.; HOUDEK, F.; MACGREGOR, J.; MÜLLER, K.; PAECH, B.; SINGH, H.; WUBMANN, H.: Einsatz von Features im Software-Entwicklungsprozess - Abschlussbericht des GI-Arbeitskreises "Features". Aachener Informatik Berichte, 2005
- [Mod10] MODELICA ASSOCIATION: Modelica® A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification – Version 3.2. Linköping, Sweden, 2010
- [Mur99] MURR, O.: Adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen. In: REINHART, G.: iwv-Forschungsberichte, Band 130, Herbert Utz Verlag, München, 1999
- [Nau00] NAUMANN, R.: Modellierung und Verarbeitung vernetzter intelligenter mechatronischer Systeme. Fortschrittsbericht VDI Reihe 20, Nr. 318, VDI Verlag, Düsseldorf, 2000
- [Nor05] NORTH, K.: Wissensorientierte Unternehmensführung – Wertschöpfung durch Wissen. Gabler Verlag, Wiesbaden, 2005
- [NT97] NONAKA, I.; TAKEUCHI, H.: Die Organisation des Wissens – Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen. Campus Verlag, Frankfurt am Main, 1997
- [Oes05] OESTERREICH, B.: Objektorientierte Softwareentwicklung. Analyse und Design mit UML 2.0. Oldenbourg Verlag, München, 7. Auflage, 2005
- [OHK+02] OBERSCHELP, O.; HESTERMAYER, T.; KLEINJOHANN, B.; KLEINJOHANN, L.: Design of Self-Optimizing Agent-Based Controllers. In: CFP Workshop 2002 – Agent-Based Simulation 3, Passau, 2002
- [OMG12] OMG: Systems Modeling Language (OMG SysML™) – Version 1.3, 2012

- [PB77] PAHL, G.; BEITZ, W.: Konstruktionslehre – Methoden und Anwendung. Springer-Verlag, Berlin, 1977
- [PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung – Methoden und Anwendung. Springer Verlag, Berlin, 7. Auflage, 2007
- [PCH+11] PFISTER, F.; CHAPURLAT, V.; HUCHARD, M.; NEBUT, C.: A Design Pattern meta model for Systems Engineering. In: Proceedings of 18th IFAC (International Federation of Automatic Control) World Congress, Mailand, Italien, 28. August - 2. September, 2011
- [PDS+12] POHLMANN, U.; DZIWOK, S.; SUCK, J.; WOLF, B.; LOH, C. C.; TICHY, M.: A Modelica Library for Real-Time Coordination Modeling. In: Proceedings of the 9th International MODELICA Conference, München, Deutschland, 3.-5. September, 2012
- [PMZ03] PAVAN, P.; MAIDEN, N. A. M.; ZHU, X.: Towards a Systems Engineering Pattern Language: Applying i\* to Model Requirements-Architecture Patterns. In Proceedings of the 2nd International workshop Software Requirements to Architectures (STRAW) in ICSE '03, Portland, Oregon, USA, 9. Mai, 2003
- [Pol66] POLANYI, M.: The Tacit Dimension. The University of Chicago Press, Chicago, 1966
- [Pol85] POLANYI, M.: Implizites Wissen. Suhrkamp Verlag, Frankfurt am Main, 1985
- [PRR06] PROBST, G.; RAUB, S.; ROMHARDT, K.: Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Gabler Verlag, Wiesbaden, 2006
- [Ris98] RISING L.: The Patterns Handbook: Techniques, Strategies, and Applications. Cambridge University Press, 1998
- [Ris00] RISING, L.: The Pattern Almanac. Cambridge University Press, 2000
- [RN05] RUSSEL, S.-J.; NORVIG, P.: Artificial Intelligence – A Modern Approach. Pearson Education, New Jersey, 2005
- [Roe11] ROELOFSON, J. K. M.: Situationsspezifische Planung von Entwicklungsprozessen. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Verlag Dr. Hut, München, 2011
- [Rop12] ROPOHL, G.: Allgemeine Systemtheorie – Einführung in transdisziplinäres Denken. Edition sigma, Berlin, 2012
- [Rot01] ROTH, K.: Konstruieren mit Konstruktionskatalogen: Band 2 – Kataloge. Springer Verlag, Berlin, 3. Auflage, 2001
- [Sal01] SALUSTRI, F. A.: Using Design Patterns to Promote Multidisciplinary Design. In: Proceedings of CSME International Conference on Multidisciplinary Design Engineering, November, Montreal, Canada, 2001
- [Sal05] SALUSTRI, F. A.: Using Design Pattern Languages in Design Engineering. In: Proceedings of 15th International Conference Engineering Design (ICED2005), Melbourne, Australien, 15.-18. August, 2005
- [Sau06] SAUER, T.: Ein Konzept zur Nutzung von Lösungsobjekten für die Produktentwicklung in Lern- und Assistenzsystemen. VDI-Fortschritt-Berichte, VDI Reihe 1, Nr. 390, VDI Verlag, Düsseldorf, 2006
- [SBD+10] SCHATTE, A.; BIEFL, S.; DEMOLSKY, M.; GOSTISCHA-FRANTA, E.; ÖSTEREICH, T.; WINKLER, D.: Best Practice Software-Engineering – Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen. Spektrum, Heidelberg, 2010
- [SBE11] SCHUMANN, H.; BERRES, A.; ESCHER, S.: Model-based System Design with the PrEMISE Model. Tag des Systems Engineering, Hamburg, Deutschland, 2011

- [Sch06] SCHMIDT, A.: Wirkmuster zur Selbstoptimierung – Konstrukte für den Entwurf selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 175, Paderborn, 2006
- [Sch12] SCHUH, G. (HRSG.): Innovationsmanagement – Handbuch Produktion und Management 3. Springer Verlag, Berlin, 2012
- [See03] SEEL, N. M.: Psychologie des Lernens – Lehrbuch für Psychologen und Pädagogen. Reinhardt Verlag, München, 2003
- [SS06] SIMPSON, J. J.; SIMPSON, M. J.: Foundational Systems Engineering (SE) Patterns for a SE Pattern Language. In: Proceedings of 16th Annual INCOSE Symposium, Orlando, USA, Juli, 2006
- [SS10] SCHMELZER, H. J.; SESSELMANN, W.: Geschäftsprozessmanagement in der Praxis. Carl Hanser Verlag, München, 7. Auflage, 2010
- [Sta73] STACHOWIAK, H.: Allgemeine Modelltheorie. Springer Verlag, Wien, 1973
- [Sta04] STAHL, W. H.: Fest-Flüssig-Trennung – Industrie Zentrifugen, Band 2, Maschinen und Verfahrenstechnik. DrM Press, Männedorf, 2004
- [Str96] STRUBE, G.: Wörterbuch der Kognitionswissenschaft. Klett-Cotta, 1996
- [Str98] STRUBE, G.: Modelling Motivation and Action Control in Cognitive Systems. In: SCHMID, U.; KREMS, J. F.; WYSOCKI, F.: Mind Modelling. Pabst, Berlin, 1998
- [Suh93] SUHM, A.: Produktmodellierung in wissensbasierten Konstruktionssystemen auf Basis von Lösungsmustern. Dissertation, Fakultät für Maschinenbau, Universität Karlsruhe, Reihe Konstruktionstechnik, Verlag Shaker, Aachen, 1993
- [SZ03] SANZ, R.; ZALEWSKI, J.: Pattern Based Control Systems Engineering – Using Design Patterns to Document, Transfer and Exploit Design Knowledge. IEEE Control Systems Magazine, 2003, S. 46-60
- [Szy02] SZYPERSKI, C.: Component Software – Beyond Object-Oriented Programming. Addison-Wesley Longman, Amsterdam, 2. Auflage, 2002
- [Ulr95] ULRICH, K.: The Role of Product Architecture in the Manufacturing Firm. In: Research Policy, 24, 1995, S. 419-440
- [Wag00] WAGNER, S. M.: Strategisches Lieferantenmanagement in Industrieunternehmen – Eine empirische Untersuchung von Gestaltungskonzepten. Dissertation, Europäische Hochschulschriften, Reihe 5, Volks- und Betriebswirtschaft, Band 2734, 2000
- [Web10] WEBER, H.: Erstellung nutzerindividueller Dokumente für die Vermittlung von Produktentwicklungswissen durch den Einsatz von Topic Maps. Dissertation, Fachgebiet Produktentwicklung, TU Darmstadt, VDI-Verlag, Düsseldorf, 2010
- [Wei06] WEILKIENS, T.: Systems Engineering mit SysML/UML – Modellierung, Analyse, Design. Dpunkt-Verlag, Heidelberg, 2006
- [Wei14-ol] WEILKINS, T.: Two kinds of Patterns. Unter: <http://model-based-systems-engineering.com/2013/02/07/two-kinds-of-patterns>, 15.10. 2014
- [ZG04] ZIMBARDO, P. G.; GERRIG, R. J.: Psychologie. Pearson Studium, München, 2004

## Normen und Richtlinien

- [DIN19226] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): Leittechnik – Regelungstechnik und Steuerungstechnik – Allgemeine Grundbegriffe. DIN 19226 Teil 1, Beuth-Verlag, Berlin, 1995
- [IEEE1076.1] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE): IEEE Standard VHDL Language Reference Manual (Integrated with VHDL-AMS changes). IEEE 1076.1-1999, The Institute of Electrical and Electronics Engineers, New York, USA, 1999
- [INC10] INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE): Systems Engineering Handbook – A Guide for System live cycle processes and activities. International Council on Systems Engineering (INCOSE), Version 3.2, 2010
- [INC12] INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE): INCOSE Systems Engineering Handbuch – Deutsche Übersetzung. GfSE, 2012
- [ISO19505-1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO); INTERNATIONAL ELECTRO-TECHNICAL COMMISSION (IEC): Information technology – Object Management Group Unified Modeling Language (OMG UML), Infrastructure, Version 2.4.1. ISO/IEC 19505-1:2012(E), ISO copyright office, Geneva, 2012
- [ISO19505-2] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO); INTERNATIONAL ELECTRO-TECHNICAL COMMISSION (IEC): Information technology – Object Management Group Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1. ISO/IEC 19505-2:2012(E), ISO copyright office, Geneva, 2012
- [ISO25010] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE). ISO 25010:2011, ISO copyright office, Geneva 2011
- [VDI2206] VEREIN DEUTSCHER INGENIEURE (VDI): Entwicklungsmethodik für mechatronische Systeme. VDI-Richtlinie 2206, Beuth-Verlag, Berlin, 2004
- [VDI2218] VEREIN DEUTSCHER INGENIEURE (VDI): Informationsverarbeitung in der Produktentwicklung – Feature- Technologie. VDI-Richtlinie 2218, Beuth-Verlag, Berlin, März 2003
- [VDI2221] VEREIN DEUTSCHER INGENIEURE (VDI): Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. VDI-Richtlinie 2221, Beuth-Verlag, Berlin, 1993
- [VDI2727] VEREIN DEUTSCHER INGENIEURE (VDI): Konstruktionskataloge – Lösung von Bewegungsaufgaben mit Getrieben – Grundlagen. VDI-Richtlinie 2727, Blatt 1, Beuth-Verlag, Berlin, 2007
- [VDI3320-2] VEREIN DEUTSCHER INGENIEURE (VDI): Werkzeugnummerung - Werkzeugordnung; Werkzeuge zum Urformen, Stoffbereiten; Umformen. VDI-Richtlinie 3320, Blatt 2, 1978

# Anhang

Inhaltsverzeichnis	Seite
A1 Ergänzungen zum Stand der Technik .....	A-1
A1.1 Beispiele für Lösungsmuster nach SIMPSON .....	A-1
A1.2 Beispiel für ein Lösungsmuster nach CLOUTIER .....	A-3
A1.3 Beispiele für Lösungsmuster nach GAUSEMEIER ET AL. ....	A-5
A2 Ergänzungen zu Kapitel 4.2 – Das Paradigma der Lösungsmuster in der PE .....	A-7
A2.1 Beispiele für Lösungsmuster der strategischen Produktplanung .....	A-7
A2.2 Beispiele für Lösungsmuster der Produktentwicklung .....	A-9
A2.3 Beispiele für Lösungsmuster der Produktionssystementwicklung..	A-10
A3 Ergänzungen zu Kapitel 4.3 – Lösungswissen für den Systementwurf ..	A-13



## A1 Ergänzungen zum Stand der Technik

### A1.1 Beispiele für Lösungsmuster nach SIMPSON

„Problem vs. Lösung“	
Problem:	Das Fehlen eines eindeutigen Verständnisses des Problems führt zu vorgefassten Meinungen hinsichtlich der Lösungsfindung
Herausforderung:	Unterschiedliche Expertise kann zu großen Differenzen führen in Bezug auf die notwendige Lösung sowie das rechtliche Umfeld
Kontext:	Mehrere Beteiligte müssen sich über ein Problem und eine zugehörige Lösung verständigen und einigen
Verwandte Muster:	„mindestens drei Systeme“
Lösung:	Mehrere Aspekte des Problems mit allen Beteiligten diskutieren und erst anschließend potentielle Designlösungen vorschlagen

Bild A-1: „Problem vs. Lösung“-Muster [vgl. SS06]

„mindestens drei Systeme“	
Problem:	Das fehlen dreier klar von einander abgegrenzter Systeme schafft unnötige Komplexität in den Zusammenhängen zwischen Problem und Lösung
Herausforderungen:	Unterschiedliche Standpunkte und Interaktionen von Systemnutzern und -entwicklern müssen festgestellt werden
Kontext:	Das „mindestens drei Systeme“-Muster ist auf alle Problemstellungen im Systementwurf anzuwenden
Verwandte Muster:	„Problem vs. Lösung“-Muster
Lösung:	Es sind explizite Systeme zu definieren, um das Endprodukt, den Entwicklungs- und Produktionsprozess, und die Umwelt zu berücksichtigen

Bild A-2: „mindestens drei Systeme“-Muster [vgl. SS06]

<b>„FRAT- Functions-Requirements-Architecture-Test“</b>	
Problem:	Adäquate Identifizierung von Problemen, deren Beschreibung sowie die Erarbeitung zugehöriger Lösungen auf unterschiedlichen Konkretisierungsstufen im Entwurf
Herausforderung:	Die Berücksichtigung der Umwelt sorgt für unterschiedliche Perspektiven und Gewichtungen möglicher Systemlösungen
Kontext:	Das FRAT-Muster wird innerhalb der Grenzen des Musters „mindestens drei Systeme“ angewendet
Verwandte Muster:	“Problem vs. Lösung”, “mindestens drei Systeme”
Lösung:	Probleme sind explizit zu definieren, um diese anschließend in der Entwicklung lösen und gegenüber der Problembeschreibung absichern zu können

Bild A-3: *FRAT – Functions-Requirements-Architecture-Test – Muster [vgl. SS06]*

## A1.2 Beispiel für ein Lösungsmuster nach CLOUTIER

Kategorie	Erklärung zu den Kategorien
Name	Ausführen C <sup>2</sup> , Command and Control (Kommandieren und Prüfen)
Aliasnamen	Nicht bekannt
Schlüsselwörter	Planen, Detektieren, Prüfen, Ausführen, C <sup>2</sup>
Problem-Kontext	Bezieht sich weder auf die "Vorbereitung" von Voraussetzungen noch auf die "Bewertung" der Nachbedingungen
Problem-beschreibung	In diesem Muster wird das Vorgehen für ein Management unterschiedlicher Situationen in einzelne Phasen unterteilt. Es ist darauf hinzuweisen, dass Rücksprünge in bereits durchlaufene Phasen oder Vorsprünge in noch nicht durchlaufene Phasen, möglich sind. Die Phasen lauten: Planen, Detektieren, Kontrollieren, Ausüben
Herausforderungen	Terminologien variieren zwischen den einzelnen Disziplinen sehr stark, daher sollten diese bei der Anwendung des Musters angepasst werden
Lösung	Dieses Muster ist die Basis für das Kommandieren und Prüfen von Schnittstellen und Informationen entlang der Stufen des Vorgehensmodells C <sup>2</sup> .
Diagramme	s. Bild A-5
Schnittstellen	Informationsflüsse zwischen den Phasen dieses Musters sowie Rückkopplungsschleifen. Einige Informationen werden nur in einer bestimmten Stufe erzeugt und dann in Form von Berichten ausgegeben. Die Terminologie, in der die Informationen übermittelt werden, kann je nach Verwendung in bestimmten Disziplinen eine Änderung erfordern.
Resultierender Kontext	Weitere Arbeiten sind erforderlich, um die Aufgaben, die in jeder Phase durchgeführt werden, zu definieren, und um diese Aufgaben auf Systeme, Hardware, Software oder Menschen zu verteilen
Beispiel(e)	Dieses Muster kann in der Modellierung eines Systems für militärische oder paramilitärische Operationen (wie beispielsweise Polizei oder der Landesverteidigung) verwendet werden. Es kann sogar in der Koordination von einem Kraftfahrzeugflottenbetrieb angewendet werden.
Übertragungsmöglichkeiten	Dies ist eine erprobte Vorgehensweise vom Militär, die jedoch auch auf weitere Disziplinen übertragen werden kann
Verwandte Muster	OODA (Observieren, Orientieren, Entscheiden, Handeln)
Referenzen	MCDP 6 Marine Corps Command and Control Handbook
Autoren	Harry Johnson Ph.D., Ken Hartnett, Satya Moorthy, Robert Cloutier, 2006.

Bild A-4: Systemarchitekturmuster Perform C<sup>2</sup> nach CLOUTIER [Clo06, S. 44]

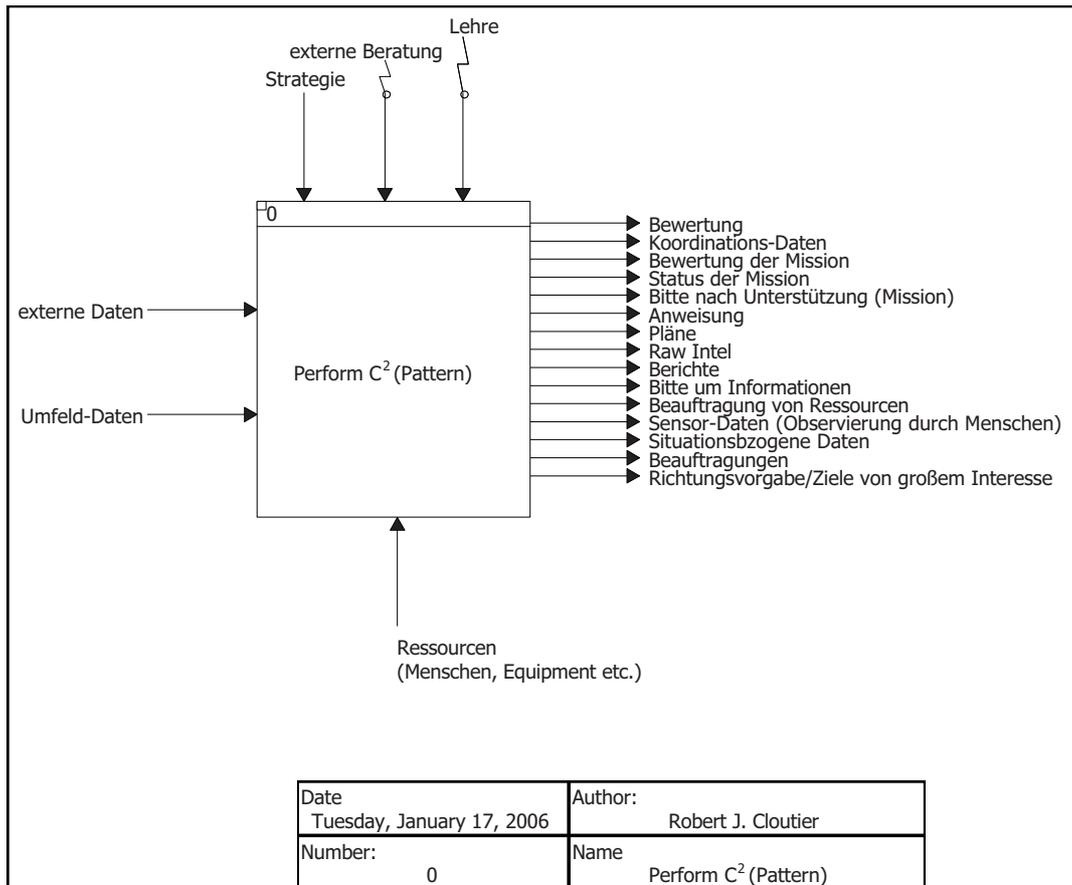


Bild A-5: UML Diagram des Systemarchitekturmusters Perform C<sup>2</sup> [Clo06, S. 45]

### A1.3 Beispiele für Lösungsmuster nach GAUSEMEIER ET AL.

Teilfunktionen	Physikal. Effekte (lösungsneutral)	Wirkprinzipien für eine Teilfunktion (Phys. Effekte sowie geometrische und stoffliche Merkmale)	Lösungselemente (Basieren auf Wirkprinzipien, aber die Merkmale sind mit konkreten Werten belegt)
<p><b>Drehmoment übertragen</b></p>	<p><b>Reibungseffekt</b></p> $F_R = \mu \cdot F_N$		<p>Werkstoff: St 37-2</p>
<p><b>Handkraft vergrößern</b></p>	<p><b>Hebeleffekt</b></p> $F_a \cdot a = F_b \cdot b$		<p>Seillänge: 10 m max. Zugbelastung: 360 kg kleinstes Hakenmass: 50mm</p>
<p><b>Signal geben wenn <math>g \geq g_a</math></b></p>	<p><b>Ausdehnungseffekt</b></p> $\Delta l = \alpha \cdot l \cdot \Delta \vartheta$	<p>Bimetall</p>	<p>Überstrom-bimetallrelais mit Nennstrom 300 A</p>

Bild A-6: Beispiele für Wirkprinzipien des Maschinenbaus [GHK+06, S. 374] (in Anlehnung an [PBF+07, S. 53])

Teilfunktionen	Physikal. Effekte (lösungsneutral)	Wirkprinzipien für eine Teilfunktion (Phys. Effekte sowie geometrische und stoffliche Merkmale)	Lösungselemente (Basieren auf Wirkprinzipien, aber die Merkmale sind mit konkreten Werten belegt)
<p><b>Energie speichern</b></p>	<p><b>Elektrisches Feld</b></p>	$W_{\text{Kond}} = \frac{1}{2} C \cdot U^2$ $C = \epsilon_r \cdot \epsilon_0 \cdot \frac{A}{d}$	<p>Elna-Kondensator ROA 851626 C=100µF U=25V</p>
<p><b>Energie wandeln</b></p>	<p><b>Lorentzkraft</b></p>	$\vec{F} = I \cdot (\vec{i} \times \vec{B})$	<p>Elektromotor C40 7000U/min.</p>
<p><b>Signal schalten</b></p>	<p><b>pn-Übergang</b></p> <p>(Transistor → Gatter → FPGA)</p>	<p>ACT Logikmodul</p>	<p>XILINX XCS10XL (FPGA) 10K Gates PLCC Package 18\$</p>

Bild A-7: Beispiele für Wirkprinzipien der Elektrotechnik [GHK+06, S. 374]

Teilfunktionen	Softwaremuster / Pattern für eine Teilfunktion	Lösungselemente / Softwarekomponenten (Basieren auf Mustern, aber die Merkmale sind konkret ausgeprägt)										
<p><b>Passe die Schnittstelle einer Klasse an eine andere an</b>, die von einem Klienten erwartet wird, ohne die Klasse selbst zu verändern</p>	<p><b>Adapter</b></p> <ul style="list-style-type: none"> <li>• Delegation</li> <li>• Datenkapselung</li> </ul>	<p><b>Datenbankzugriff</b></p>										
<p><b>Übertrage Anwendungsdaten</b> zwischen zwei Kommunikationsteilnehmern, unabhängig von der Art der physikalischen Netzwerkverbindung</p>	<p><b>Abstraktion über Schichten</b></p> <ul style="list-style-type: none"> <li>• Kommuniziere nur zwischen benachbarten Schichten</li> <li>• Entkopple nicht direkt benachbartes Verhalten</li> </ul>	<p><b>TCP/IP-Protokollstapel (Linux)</b></p> <table border="1"> <tr> <td>Transport:</td> <td>TCP / IP:</td> </tr> <tr> <td>Network:</td> <td></td> </tr> <tr> <td>LogicalLink:</td> <td>IEEE802.2</td> </tr> <tr> <td>MediaAccess:</td> <td>Ethernet (CSMA/CD)</td> </tr> <tr> <td>Physical:</td> <td>Twisted Pair</td> </tr> </table>	Transport:	TCP / IP:	Network:		LogicalLink:	IEEE802.2	MediaAccess:	Ethernet (CSMA/CD)	Physical:	Twisted Pair
Transport:	TCP / IP:											
Network:												
LogicalLink:	IEEE802.2											
MediaAccess:	Ethernet (CSMA/CD)											
Physical:	Twisted Pair											
<p><b>Vermeide die Kopplung</b> des Auslösers einer Anfrage an seinen Empfänger, in dem mehr als ein Objekt die Möglichkeit zur Erledigung erhält</p>	<p><b>Zuständigkeitskette</b></p> <ul style="list-style-type: none"> <li>• Verkette die Empfänger; leite Anfragen weiter</li> </ul>	<p><b>Ereignisverarbeitung (Windows)</b></p>										

Bild A-8: Beispiele für Wirkmuster der Softwaretechnik [GHK+06, S. 375]

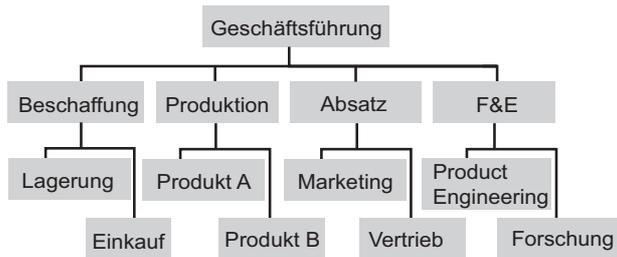
Teilfunktionen	Wirkmuster	Lösungselemente
Größe beeinflussen	<p><b>Steuerung</b></p>	<ul style="list-style-type: none"> <li>• Inverse Strecke</li> <li>• Optimale Steuerungsfunktion</li> <li>• Kennfeld</li> </ul>
	<p><b>Regelung</b></p>	<p>Eingrößenregler</p> <p>Mehrgrößenregler</p> <ul style="list-style-type: none"> <li>• Kaskadenregelung</li> <li>• Zustandsvektorrückführung</li> <li>• ...</li> </ul>
Größe bestimmen	<p><b>Beobachter</b></p>	<ul style="list-style-type: none"> <li>• Luenberg-Beobachter</li> <li>• Kalman-Filter</li> </ul>
	<p><b>Nachbilden</b></p>	<ul style="list-style-type: none"> <li>• Modell der Strecke, z.B. PT1,</li> </ul>
	<p><b>Messen/Erfassen</b></p>	<p>...</p>

Bild A-9: Beispiele für Wirkmuster der Softwaretechnik [GHK+06, S. 375]

## A2 Ergänzungen zu Kapitel 4.2 – Das Paradigma der Lösungsmuster in der PE

### A2.1 Beispiele für Lösungsmuster der strategischen Produktplanung

#### Funktionale Organisation



#### Divisionale Organisation

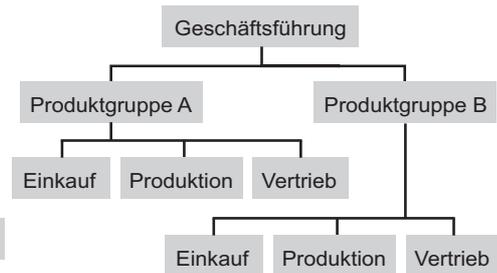
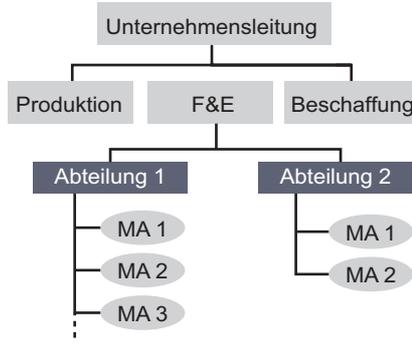
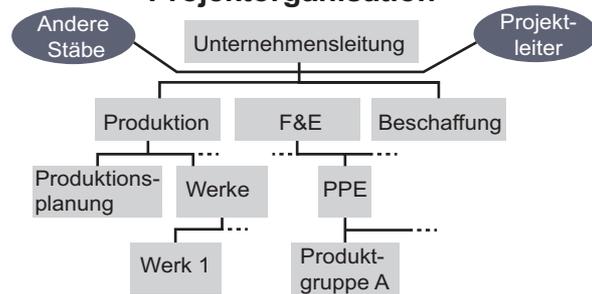


Bild A-10: Gegenüberstellung von einer funktionalen und divisionalen Organisation

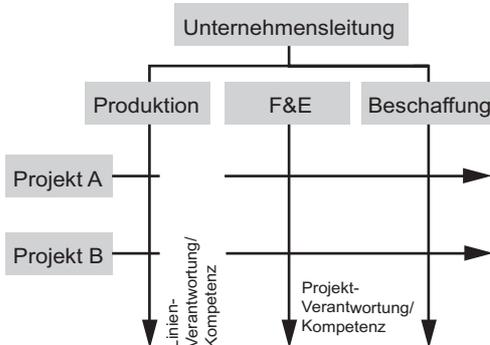
#### Projektmanagement in der Linie



#### Einfluss- Projektorganisation



#### Matrix- Projektorganisation



Legende

■ : Projektleiter    ● : Projektgruppenmitglied

#### Reine Projektorganisation

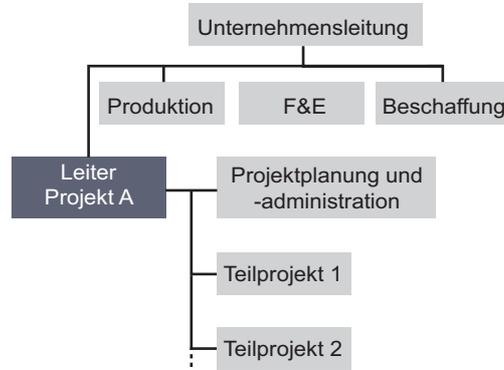


Bild A-11: Projektorganisationsformen des Innovationsbereiches [Sch12, S. 38]

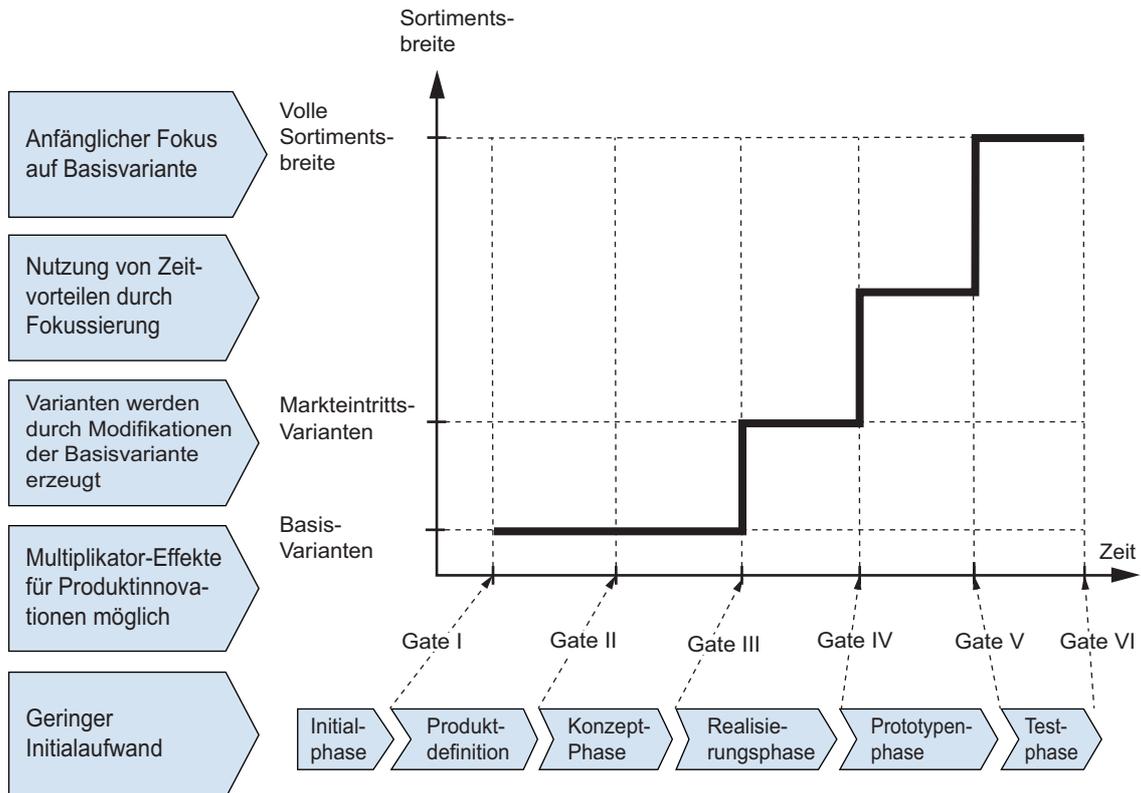


Bild A-12: Sortimentsstrategie Add-on-Engineering [Sch12, S. 110]

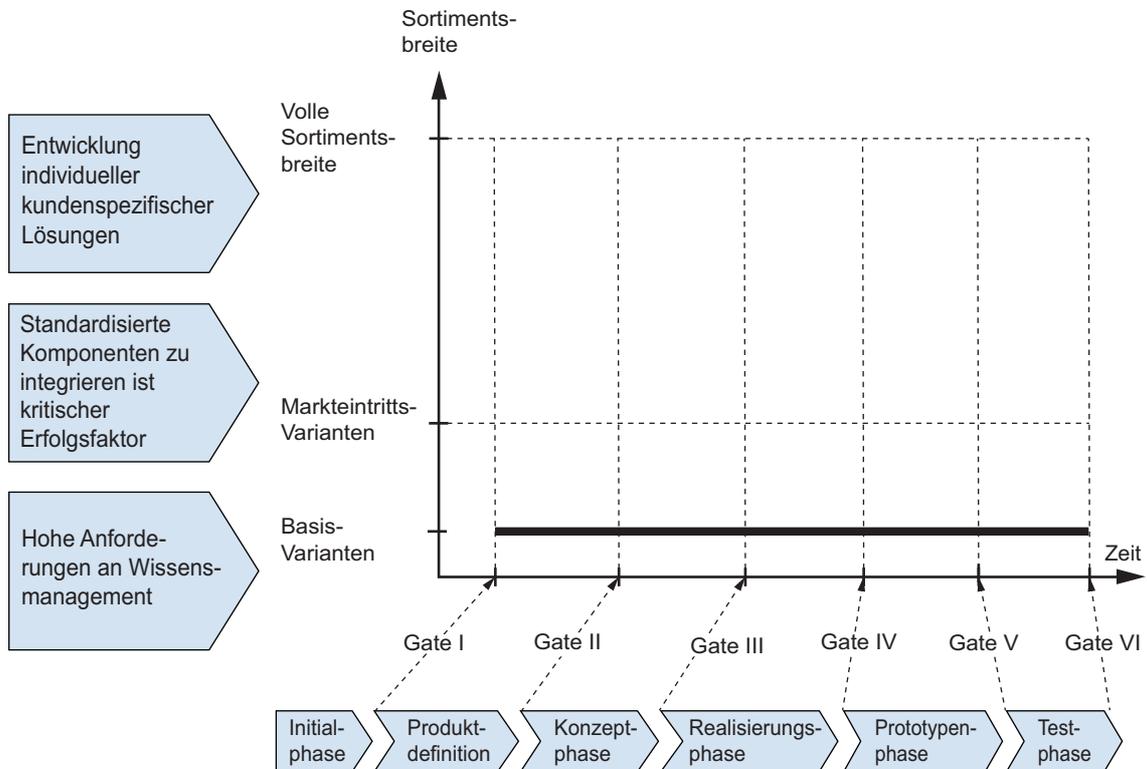


Bild A-13: Sortimentsstrategie Non-variant-Engineering [Sch12, S. 110]

## A2.2 Beispiele für Lösungsmuster der Produktentwicklung

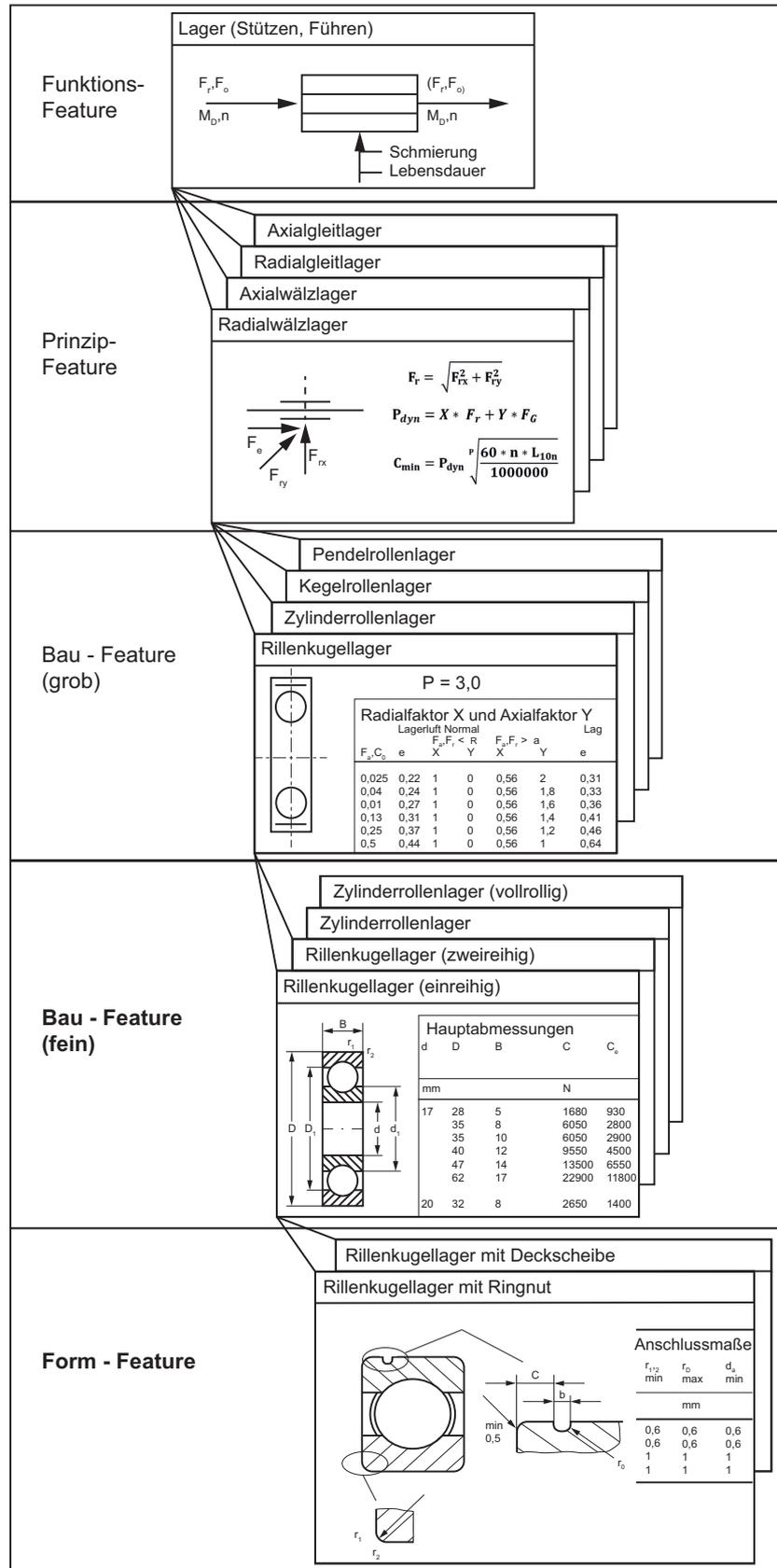


Bild A-14: Feature-Klassen am Beispiel von Wälzlagern [PBF+07, S. 758]

## A2.3 Beispiele für Lösungsmuster der Produktionssystementwicklung

### MID-Entwurfsmuster: Stacking

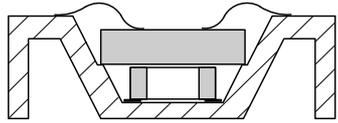
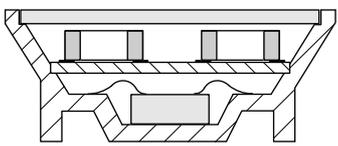
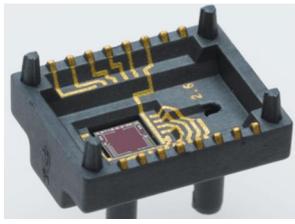
MID-Muster: Stacking		
<p style="text-align: center;"><b>Merkmale</b></p> <p><b>Geometrische Klassifizierung:</b> 2 ½ D</p> <p><b>Produktanforderungen:</b></p> <ul style="list-style-type: none"> <li>- Bauteile kompakt anordnen</li> <li>- Bauteile stapeln</li> </ul> <p><b>Grundkörperherstellung:</b></p> <ul style="list-style-type: none"> <li>- 2K-Spritzguss</li> <li>- Spritzguss und Laserdirektstrukturierung</li> </ul> <p><b>Aufbau- und Verbindungstechnik:</b></p> <ul style="list-style-type: none"> <li>- Drahtbonden</li> <li>- flip chip</li> </ul> <p><b>Anlagentechnik für 3D-Montage:</b></p> <ul style="list-style-type: none"> <li>- 2D/3D Dispensersystem</li> <li>- 2D/3D Substrathalter</li> <li>- 2D/3D Montagesystem</li> </ul>	<p style="text-align: center;"><b>Gestaltung</b></p> <div style="text-align: center;">  </div> <p style="text-align: center;">Oben liegendes Bauteil wird an den Kanten abgestützt und über Bonddrähte zum Layout verbunde</p> <div style="text-align: center;">  </div> <p style="text-align: center;">SMD-Bauteile auf Zwischenebene platziert. Anbindung erfordert definierte Kontaktstellen.</p>	<p style="text-align: center;"><b>Anwendungsbeispiel</b></p> <div style="text-align: center;">  </div> <p style="text-align: center;">Gehäuse eines Drucksensors.  <a href="http://www.harting-mitronics.ch/produkte/anwendungen/sensorik">http://www.harting-mitronics.ch/produkte/anwendungen/sensorik</a></p>
<b>Elektrische MID-Funktionen</b>		
<b>Mechanische MID-Funktionen</b>		
<b>Fertigungsrestriktionen</b>		
<ul style="list-style-type: none"> <li>- Spritzguss: Große Materialanhäufungen vermeiden; Rippenstrukturen oder freie Räume in die Bauteilgeometrie einbringen.</li> <li>- Strukturierung: Scharfkantige Leiterbahnübergänge an Kanten vermeiden (Radius vorsehen).</li> <li>- Metallisierung: Abflussmöglichkeiten auf Grund der wannenförmigen Gehäusegeometrie vorsehen.</li> <li>- Bestückung: Freie Höhe des Bestückungssystems beachten.</li> <li>- ...</li> </ul>		

Bild A-15: Entwurfsmuster Stacking [GJD+13]

### Simulationsmuster für die Technologie MID

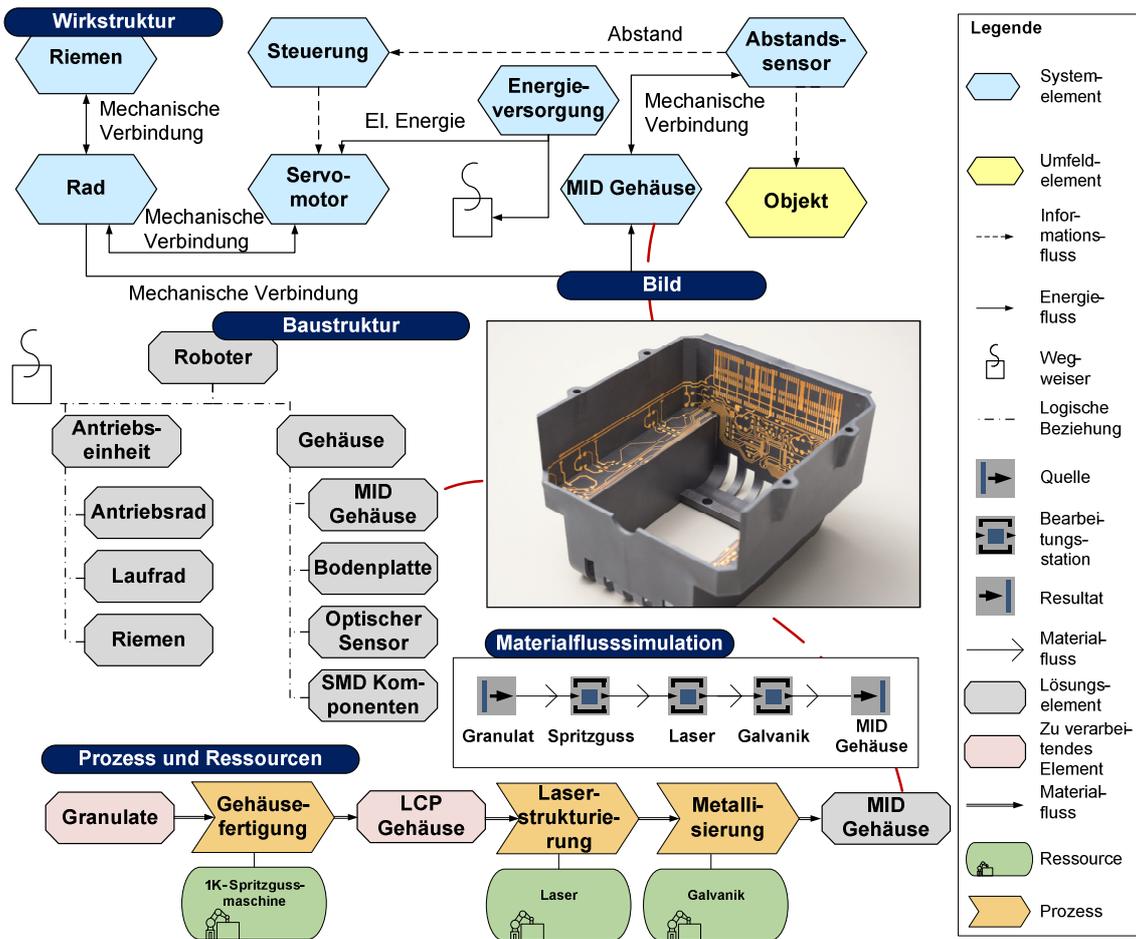
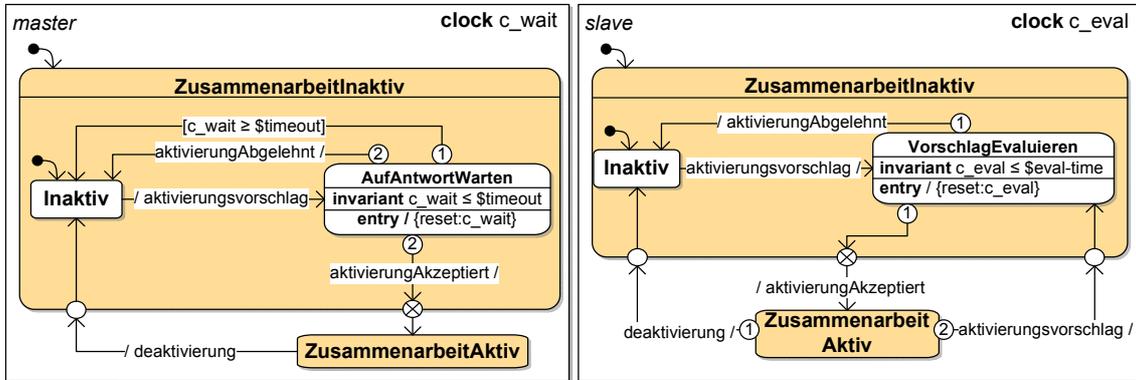


Bild A-16: Aspekte eines Simulationsmusters für die Technologie MID<sup>69</sup>

<sup>69</sup>Für weitere Informationen sei an dieser Stelle auf [ADG13] aus der *Liste der eigenen Publikationen* verwiesen.



### A3 Ergänzungen zu Kapitel 4.3 – Lösungswissen für den Systementwurf



**Legende**

- Real-Time Statechart      □ Zustand      → Transition      ○ Eintrittspunkt
- Rollenkonnetktor      ● Startzustand      ⊙→ Transition mit Priorität      ⊗ Austrittspunkt

Bild A-17: Verhalten des Koordinationsmusters Echtzeit-Zusammenarbeit aushandeln, formalisiert mit Real-Time Statecharts [GTS14, S. 122]

