

**Rahmenwerk zur Modellierung
einer plausiblen Systemstruktur
mechatronischer Systeme**

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

genehmigte
DISSERTATION

von
M.Sc. Lydia Kaiser
aus *Alma-Ata*

Tag des Kolloquiums: 19. Dezember 2013
Referent: Prof. Dr.-Ing. Jürgen Gausemeier
Korreferent: Prof. Dr. rer. nat. Wilhelm Schäfer

Zusammenfassung

Die Entwicklung mechatronischer Systeme ist eine Herausforderung. Die steigende Interdisziplinarität und die daraus resultierende Komplexität der Systeme erfordert von Beginn an eine enge Zusammenarbeit und ein einheitliches Systemverständnis aller Beteiligten. Die disziplinübergreifende Systembeschreibung mit einem Systemmodell bietet das Potential, dieser Herausforderung zu begegnen. Ein zentraler Aspekt des Systemmodells ist die Systemstruktur. Sie beschreibt die Elemente und ihre Beziehungen im System. Die erfolgreiche disziplinübergreifende Arbeitsweise erfordert eine vergleichbare, vollständige und richtige Systemstruktur. Der Begriff „Plausibilität“ bringt dies zum Ausdruck.

Zur Modellierung einer plausiblen Systemstruktur wird ein Rahmenwerk erarbeitet. Den Kern bilden Vorgaben zur Beschreibung der Systemstruktur. Darüber hinaus bietet das Rahmenwerk eine Plausibilitätsprüfung, ein Konzept für eine Werkzeugunterstützung sowie ein Vorgehensmodell. Dabei geht das Vorgehensmodell auf die disziplinübergreifende Erstellung, die formalisierte rechnerbasierte Modellierung und Prüfung der Systemstruktur ein.

Das Rahmenwerk wird an zwei Produkten beispielhaft angewendet: einer Tretkraftunterstützung und einer Sortieranlage. Während bei der Tretkraftunterstützung die Systemstruktur in der frühen Phase erstellt wird, findet bei der Sortieranlage eine nachträgliche Modellierung statt.

Summary

The development of mechatronic systems can be challenging. The growing complexity and interdisciplinary dependence of systems require close collaboration and uniform system understanding by all the parties involved. The cross-disciplinary system description in the form of a system model can potentially face this challenge. A key aspect is the system structure, which describes the elements and their relationships in the system. The cross-disciplinary way of working requires a comparable, complete and correct description of the system structure, which is expressed by using the word “plausibility”.

A framework is developed to model a plausible system structure. The main focus lies on guidelines for the description of the system structure. In addition, the framework consists of a plausibility check, a concept for a software tool and a process model. Thereby the process model describes the cross-disciplinary creation, the formalized computer-based modeling and checking of the system structure.

The framework is applied on two products: a pedal-assist and a sorting system. While the pedal-assist system structure is created in the early phase, the sorting system is modeled subsequently.

Liste der veröffentlichten Teilergebnisse

- [DGL+08] DEYTER, S.; GAUSEMEIER, J.; LACKMANN, L.; STEFFEN, D.: InZuMech – Instrumentarium für die frühzeitige Zuverlässigkeitsanalyse mechatronischer Systeme. wt Werkstattstechnik online (7/8)/2008, Springer-VDI Verlag, Düsseldorf, S. 607–614
- [DLH+09] DEYTER, S.; LACKMANN, L.; HOLST, C.; THESING, W.; MIDDENDORF, A.; STEFFEN, D.: Frühzeitige Zuverlässigkeitsbewertung miniaturisierter mechatronischer Robotermodule. In: Gausemeier, J.; Rammig, F.J.; Trächtler, A.; Schäfer, W. (Hrsg.): 6. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 250, Paderborn, 2009
- [GKP09] GAUSEMEIER, J.; KAISER, L.; POOK, S.: FMEA von komplexen mechatronischen Systemen auf Basis der Spezifikation der Prinziplösung. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 11/2009, Carl Hanser Verlag, München, S. 1011–1017
- [GPD+09] GAUSEMEIER, J.; PÖSCHEL, M.; DEYTER, S.; KAISER, L.: Modeling and Analyzing Fault-Tolerant Mechatronic Systems. In: Proceedings of International Conference on Engineering Design, Stanford, 2009
- [KNT09] KAISER, L.; NORDSIEK, D.; TERFLOTH, A.: Softwaregestützte Konzipierung komplexer mechatronischer Systeme und der zugehörigen Produktionssysteme. ATZ Elektronik 5/2009, GWV Fachverlag GmbH, Wiesbaden
- [GBK10] GAUSEMEIER, J.; BRANDIS, R.; KAISER, L.: Auswahl von Montageverfahren auf Basis der Produktkonzeption. In: Gausemeier, J.; Rammig, F.J.; Schäfer, W.; Trächtler, A. (Hrsg.): 7. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 272, Paderborn, 2010
- [GDK10] GAUSEMEIER, J.; DOROCIAC, R.; KAISER, L.: Computer-Aided Modeling of the Principle Solution of Mechatronic Systems: A Domain-Spanning Methodology for the Conceptual Design of Mechatronic Systems. In: Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Montreal, Canada, 2010
- [GKP+10] GAUSEMEIER, J.; KAISER, L.; POOK, S.; NYSSSEN, A.; TERFLOTH, A.: Rechnerunterstützte Modellierung der Prinziplösung mechatronischer Systeme. In: Gausemeier, J.; Rammig, F.J.; Schäfer, W.; Trächtler, A. (Hrsg.): 7. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 272, Paderborn, 2010
- [GBK12] GAUSEMEIER, J.; BRANDIS, R.; KAISER, L.: Integrative Conceptual Design of Products and Production Systems for Mechatronic Systems. In: Proceedings of 13th International Workshop on Research and Education in Mechatronics, Paris, 2012
- [KAG+12] KAISER, L.; ANACKER, H.; GAUSEMEIER, J.; DUMITRESCU, R.: Plausibilitätsanalyse der Wirkstruktur am Beispiel einer Sortieranlage. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [KDB+13] KAISER, L.; DUMITRESCU, R.; BREMER, C.; GAUSEMEIER, J.: Sichten der Systemstruktur im Model-Based Systems Engineering für mechatronische Systeme. In: Spath, D.; Bertsche, B.; Binz, H. (Hrsg.): Stuttgarter Symposium für Produktentwicklung SSP 2013, Stuttgart, 2013

- [KDH+13] KAISER, L.; DUMITRESCU, R.; HOLTSMANN, J.; MEYER, M.: Automatic Verification of Modeling Rules in Systems Engineering for Mechatronic Systems. In: Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Portland, Oregon, 2013
- [IKD+13] IWANEK, P.; KAISER, L.; DUMITRESCU, R.; NYSSSEN, A.: Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2013
- [SKD+13] SCHMITT, N.; KAISER, L.; DUMITRESCU, R.; HOFMANN, M.: Von der Anforderungserfassung bis zur Funktionsstruktur – Ein Systems Engineering-Vorgehen für die industrielle Praxis. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2013

Inhaltsverzeichnis	Seite
1 Einleitung	1
1.1 Problematik.....	1
1.2 Zielsetzung	3
1.3 Vorgehensweise	4
2 Problemanalyse	7
2.1 Begriffsdefinitionen	7
2.2 Mechatronische Systeme	10
2.2.1 Klassen mechatronischer Systeme	11
2.2.2 Komplexität der Systeme.....	13
2.3 Entwicklungsmethodik für mechatronische Systeme nach VDI 2206 ...	15
2.4 Systems Engineering.....	17
2.4.1 SE-Bestandteile nach HABERFELLNER et al.....	17
2.4.2 Einsatz und Nutzen von Systems Engineering	18
2.4.3 Rollen im Systems Engineering.....	21
2.5 Model-Based Systems Engineering.....	22
2.5.1 Modellbildung in der Produktentwicklung	23
2.5.2 Handlungsfeld MBSE	24
2.5.3 Systemmodell	26
2.5.4 Partizipation der SE-Rollen am Systemmodell	33
2.5.5 Modellierungssprache	34
2.5.6 Komplexität im Systemmodell.....	37
2.6 Grundsätze ordnungsmäßiger Modellierung	38
2.7 Problemabgrenzung	40
2.8 Anforderungen.....	43
3 Stand der Technik	45
3.1 Modellierungssprachen	45
3.1.1 METUS - Sprache	45
3.1.2 CONSENS - Sprache	46
3.1.3 OPM	48
3.1.4 SysML	49
3.2 Modellierungsmethoden	51
3.2.1 METUS - Methode.....	51

3.2.2	CONSENS - Methode.....	53
3.2.3	FAS	54
3.2.4	SYSMOD.....	55
3.2.5	OOSEM	57
3.2.6	Funktionale und technische Entwicklung nach ALT	58
3.3	Modellierungswerkzeuge.....	59
3.3.1	METUS - Software.....	59
3.3.2	Mechatronic Modeller	60
3.3.3	Enterprise Architect – Systems Engineering Edition.....	61
3.3.4	CATIA V6 Systems.....	63
3.3.5	Mechatronic Concept Designer	64
3.4	Modellierungsrichtlinien	65
3.4.1	Kochbuch für MBSE mit SysML.....	65
3.4.2	Richtlinien nach ALT.....	66
3.5	Funktionsmodellierung	66
3.6	Zusammenfassende Betrachtung der Strukturmodellierung.....	68
3.7	Handlungsbedarf	70
4	Rahmenwerk zur Modellierung einer plausiblen Systemstruktur.....	73
4.1	Definition und Klassifikation der Modellkonstrukte.....	74
4.1.1	Beziehungen.....	76
4.1.1.1	Energie	77
4.1.1.2	Information.....	78
4.1.1.3	Stoff	79
4.1.1.4	Mechanische Verbindung	80
4.1.2	Technische Elemente	81
4.1.2.1	Energieumsetzendes Element.....	83
4.1.2.2	Informationsumsetzendes Element	84
4.1.2.3	Stoffumsetzendes Element.....	86
4.1.2.4	Tragstruktur	88
4.1.3	Nicht-technische Elemente	89
4.1.3.1	Lebewesen	90
4.1.3.2	Stoff-Objekt.....	91
4.1.3.3	Umgebung.....	91
4.2	Richtlinien und Bedingungen der Modellierung	92
4.2.1	Vergleichbarkeit.....	92
4.2.2	Vollständigkeit	94
4.2.3	Richtigkeit.....	100
4.3	Plausibilitätsprüfung	101
4.3.1	Matrix der energieumsetzenden Elemente	104
4.3.2	Matrix der Sensoren	106

4.4	Sichten auf die Systemstruktur	107
4.5	Werkzeugunterstützung.....	110
4.5.1	Konzept der Werkzeugunterstützung	110
4.5.1.1	Arbeitsweise und Werkzeugfunktionen.....	111
4.5.1.2	Automatische Plausibilitätsprüfung.....	115
4.5.2	Prototypische Implementierung der Plausibilitätsprüfung	118
4.6	Vorgehensmodell zur Erstellung plausibler Systemstruktur.....	121
4.6.1	Phase 1: Modellierungs-Workshops	122
4.6.2	Phase 2: Formalisierung des Modells.....	124
4.6.3	Phase 3: Modell-Review	125
4.6.4	Phase 4: Modellvervollständigung	126
4.6.5	Phase 5: Modellfreigabe	127
5	Anwendung und Bewertung	129
5.1	Anwendungsbeispiel: Tretkraftunterstützung für ein Fahrrad	129
5.1.1	Phase 1: Modellierungs-Workshops	130
5.1.2	Phase 2: Formalisierung des Modells.....	133
5.1.3	Phase 3: Modell-Review	136
5.1.4	Phase 4: Modellvervollständigung	137
5.1.5	Phase 5: Modellfreigabe	138
5.2	Anwendungsbeispiel: Sortieranlage	139
5.2.1	Oberste Hierarchieebene der Systemstruktur.....	140
5.2.2	Sichten auf die Systemstruktur der Sortieranlage.....	141
5.2.2.1	Stoffspezifische Sicht	141
5.2.2.2	Energiespezifische Sicht	142
5.2.2.3	Informationsspezifische Sicht	143
5.2.2.4	Ergänzungen der Systemstruktur	144
5.3	Bewertung der Arbeit an den Anforderungen	145
6	Resümee und Ausblick	147
7	Abkürzungsverzeichnis	151
8	Literaturverzeichnis	153

Anhang

A1	Steckbriefe der Elementklassen	A-1
A2	Liste der Richtlinien und Bedingungen	A-9
A3	Matrix der energieumsetzenden Elemente	A-12
A4	Matrix der Sensoren	A-13
A5	Aufgaben und Fähigkeiten eines Systems Engineers	A-14
A6	Systemstruktur der Sortieranlage	A-15

1 Einleitung

Die vorliegende Arbeit entstand in Rahmen der Grundlagenforschung am Heinz Nixdorf Institut sowie der anwendungsorientierten Forschung am Fraunhofer-Institut für Produktionstechnologie. Zahlreiche Grundlagen der disziplinübergreifenden Konzipierung mechatronischer Systeme sowie Erfahrungen der werkzeugtechnischen Umsetzung wurden am Heinz Nixdorf Institut in Rahmen des Verbundprojekts VireS¹ erarbeitet. Ziel des Projekts VireS war ein Instrumentarium für die integrative Entwicklung von Produkt und Produktionssystem. In diesem Zusammenhang wurde in Kooperation mit einem Softwareunternehmen unter anderem das Modellierungswerkzeug Mechatronic Modeller umgesetzt.

Die Grundlagen zur disziplinübergreifenden Beschreibung mechatronischer Systeme wurden in der Zeit am Fraunhofer-Institut in der Projektgruppe Entwurfstechnik Mechatronik vertieft und in der Industrie eingesetzt. Die Anwendung erfolgte in Rahmen von zahlreichen Workshops mit Unternehmen der Branchen Automotive, Automatisierungstechnik sowie Maschinen- und Anlagenbau. Die daraus gewonnen Handlungsfelder und Erkenntnisse flossen in die Arbeit ein. Diese sind im Rahmen des Querschnittsprojekts Systems Engineering² des Spitzenclusters it's OWL³ wissenschaftlich aufgearbeitet worden. Ziel des Projekts ist eine ganzheitliche disziplinübergreifende Systematik zur Entwicklung Intelligenter Technischer Systeme. Mit den Ansätzen des Model-Based Systems Engineering wird der fachdisziplinübergreifende Systementwurf unterstützt. Die vorliegende Arbeit ist ein Bestandteil dieser Systematik. Damit ordnet sie sich in das Themenfeld Model-Based Systems Engineering (MBSE) ein und beschreibt ein *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*.

1.1 Problematik

Moderne Erzeugnisse des Maschinenbaus beruhen auf einem engen Zusammenwirken von Mechanik, Elektrotechnik, Regelungstechnik und Softwaretechnik. Der Begriff Mechatronik bringt dies zum Ausdruck. Mechatronische Systeme sind aus dem alltäglichen Leben und in der Wirtschaft nicht mehr wegzudenken. Sie unterstützen den Menschen beim Transport, der Produktion und Kommunikation. Aktuell ist ein starker Anstieg des Softwareanteils in den Systemen zu beobachten. Durch den Fortschritt in der

¹ Virtuelle Synchronisation von Produktentwicklung und Produktionssystementwicklung (VireS), gefördert vom Bundesministerium für Bildung und Forschung (BMBF) im Rahmenkonzept „Forschung für die Produktion von morgen“. Projektlaufzeit: Juli 2008 bis Juni 2011

² Gefördert vom BMBF im Rahmen des Spitzenclusters it's OWL. Projektstart: Juni 2012

³ Intelligente Technische Systeme OstWestfalenLippe (it's OWL)

Kommunikations- und Nachrichtentechnik können bestehende Funktionen durch Software realisiert, aber auch komplett neue Funktionen umgesetzt werden. Der Trend geht zu Intelligenten Technischen Systemen, die sich durch die Begriffe adaptiv, robust, vorausschauend und benutzungsfreundlich charakterisieren lassen [aca09], [Dum11], [GAC+13].

Mechatronische Systeme zeigen eine hohe Interdisziplinarität auf, so dass an der Entwicklung unterschiedliche Fachdisziplinen beteiligt sind. Durch diese Interdisziplinarität, aber auch durch die Komplexität der Systeme selbst, wächst die Herausforderung einer effektiven und effizienten Entwicklung [BHL07-ol], [GLL12]. Die unterschiedlichen Fachdisziplinen müssen von der ersten Produktidee an miteinander das System konzipieren und entwickeln. Ein integrativer Ansatz zur frühzeitigen Berücksichtigung der Produktionssysteme ist unabdingbar, um sämtliche Lösungsvarianten ins Kalkül zu ziehen [ES05], [GLL12]. Es herrscht ein hoher Bedarf an Abstimmung und verzahnter Zusammenarbeit. Dies muss bereits bei den Anforderungen beginnen und sich über den gesamten Produktentstehungsprozess fortsetzen. Hier zeigen sich jedoch Defizite. Dies liegt unter anderem daran, dass die Entwickler⁴, bedingt durch ihren fachlichen Hintergrund, unterschiedliche Denk- und Arbeitsweisen haben. Dies führt häufig zu Missverständnissen, da teilweise gleiche Begriffe unterschiedlich genutzt werden oder unterschiedlich verwendete Begriffe das Gleiche meinen. Jedoch liegen die Schwierigkeiten in der Zusammenarbeit nicht nur an den verwendeten Begriffen. Es fehlen Ausdrucksmittel, die losgelöst von fachdisziplinspezifischen Beschreibungsmöglichkeiten die Kommunikation unterstützen. Diese Notwendigkeit zeigt sich auch durch ein fehlendes einheitliches Verständnis des Systems bei den beteiligten Entwicklern. Die Entwicklungstätigkeiten erfolgen in Teillösungen, so dass ein Zusammenwirken erst in der Integration ersichtlich wird. Änderungen in der späten Phase der Entwicklung sind kosten- und zeitintensiv [INC12], [Fra06], [BGJ+09].

Dieser Herausforderung stellt sich das Systems Engineering. Mit dem Systemdenken im Vordergrund wird bei allen beteiligten Disziplinen ein einheitliches und ganzheitliches Systemverständnis erreicht [INC12]. Dabei sollen Modelle unterstützen, dies zu erreichen. Mit diesen Ansätzen beschäftigt sich der Bereich Model-Based Systems Engineering. Ein sogenanntes Systemmodell steht im Mittelpunkt der Entwicklung. Dieses beschreibt das System interdisziplinär und bietet damit eine Plattform zur Kommunikation und Kooperation der beteiligten Disziplinen [INC07-ol], [GFD+09], [FMS12]. Die Modellierung erfolgt mit graphischen Modellierungssprachen, so dass die damit erstellten Diagramme Beziehungen visualisieren, die sonst aufwendig in Textform umschrieben werden. Neben den Diagrammen wird das Repository (Datenmodell) erstellt, das die Modellinhalte beinhaltet. Dieses muss softwaretechnisch verwaltet und gepflegt werden

⁴ Die Inhalte der vorliegenden Arbeit beziehen sich in gleichem Maße sowohl auf Frauen als auf Männer. Aus Gründen der besseren Lesbarkeit wird jedoch die männliche Form (Entwickler, Konstrukteur etc.) für alle Personenbezeichnungen gewählt. Die weibliche Form wird dabei stets mitgedacht.

[Alt12b]. Es existieren Bestrebungen, das Repository mit allen anderen im Produktlebenszyklus anfallenden Produktdaten zu verbinden. Hier lassen sich zwei alternative Ansätze beobachten: die Anbindung des Systemmodells an PDM-Systeme⁵ oder eine direkte Kopplung der verwendeten Softwarewerkzeuge [Sen13], [GSG+09], [Alt12b], [FMS12].

Das angestrebte Systemmodell besteht aus verschiedenen Aspekten, die das System disziplinübergreifend beschreiben. Diese lassen sich in die Kategorien Anforderungen, Verhalten und Struktur einteilen. Dabei nimmt die Struktur eine wesentliche Rolle ein, da diese die Bestandteile des Systems und deren Beziehung beschreibt. Auf Basis der Systemstruktur kann die interdisziplinäre Zusammenarbeit maßgeblich unterstützt und das einheitliche Systemverständnis erreicht werden. Verschiedene Modellierungssprachen, Methoden und Werkzeuge ermöglichen die Beschreibung der Struktur. Eine Analyse von bereits modellierten Systemstrukturen zeigt jedoch erhebliche Unterschiede in der Art und Weise der Modellierung. Auch innerhalb eines Ansatzes finden sich Strukturen, die einen gleichen Sachverhalt auf unterschiedliche Weise beschreiben. Dies liegt an einer mangelnden Definition der zu beschreibenden Elemente und Beziehungen. Darüber hinaus fehlt es an Richtlinien, die den Anwender unterstützen, die Modelle vergleichbar, vollständig und richtig zu erstellen. Erst wenn diese Anforderungen erfüllt sind, kann auf Basis der Systemstruktur eine interdisziplinäre Zusammenarbeit effektiv erfolgen.

Die Erstellung eines Systemmodells und damit auch der Systemstruktur muss dabei zwei entgegengesetzten Anforderungen gerecht werden. Die Modelle müssen von vielen Personen mit unterschiedlichem fachlichem Hintergrund erstellt, aber vor allem gelesen werden können. Da die Systemmodellierung die Entwicklungstätigkeiten unterstützt, jedoch die disziplinspezifischen Vorgehensweisen nicht ersetzt, bedarf es einer intuitiven, leicht erlernbaren Anwendung [aca09]. An die rechnerbasierte Modellierung werden hingegen andere Anforderungen gestellt. Hier wird zur Erstellung eines Repository, das zudem auswertbar ist, ein hoher Grad an Formalisierung benötigt. Damit bedarf es einer systematischen Vorgehensweise, die den Kompromiss zwischen einer intuitiven, einfachen Anwendung und der formalen, rechnerinternen Modellierung schafft [SMM+12], [ALR12].

1.2 Zielsetzung

Ziel der Arbeit ist ein *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*. Dabei ist eine Systemstruktur dann als plausibel zu bewerten, wenn sie vergleichbar, vollständig und richtig modelliert wird. Das Rahmenwerk soll so konzipiert sein, dass es aus Bestandteilen besteht, die im Einzelnen oder als Gesamtheit eingesetzt werden können.

⁵ PDM – Produktdaten-Management

Mit der Ausrichtung auf mechatronische Systeme soll das Rahmenwerk eine klare Vorgabe geben, wie die Systemstruktur aufgebaut ist und welche Elemente und Beziehungen wie zu modellieren sind. Die Vorgaben sollen in Form von Richtlinien aufbereitet werden, so dass der Anwender bei der Erstellung der Systemstruktur auf Plausibilität achtet. Darüber hinaus soll auch die Möglichkeit geben werden, eine Systemstruktur im Nachhinein zu prüfen und damit eine Aussage über die Vergleichbarkeit, Vollständigkeit und Richtigkeit zu erhalten. Hierzu soll ein Konzept einer Werkzeugunterstützung aufzeigen, wie eine effiziente Modellierung computerbasiert erfolgen kann.

Ein Vorgehensmodell soll die Erstellung einer plausiblen Systemstruktur und damit die Anwendung der Bestandteile aufzeigen. Dabei soll auf die disziplinübergreifende Arbeitsweise bei der Modellierung eingegangen werden. Hierzu ist eine Trennung zwischen der disziplinübergreifenden Erarbeitung einer Systemstruktur und der rechnerinternen Abbildung vorzunehmen.

Die Anwendbarkeit der Bestandteile soll an zwei Systemen nachgewiesen werden. Dabei soll sich ein System in der Neuentwicklung befinden und wird damit disziplinübergreifend erarbeitet. Hingegen soll das zweite System bereits realisiert vorliegen. Dieser Fall soll die Modellierung der Struktur zur Systemanalyse beschreiben (z.B. für das Reverse Engineering).

1.3 Vorgehensweise

Nach der Einleitung in diesem Kapitel findet in **Kapitel 2** eine ausführliche **Problem-analyse** statt. Dazu wird das Forschungsfeld der Arbeit abgesteckt und Grundlagen beschrieben. Die Arbeit befasst sich mit mechatronischen Systemen, die zu Beginn eingeführt werden. Diese Systeme zeichnen sich durch hohe Interdisziplinarität und Komplexität aus. Dies wirkt sich auch auf die Entwicklung der Systeme aus. Hierzu wird die VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ vorgestellt sowie die Grundlagen und Handlungsfelder des Systems Engineering. Der Schwerpunkt von Kapitel 2 liegt auf Model-Based Systems Engineering (MBSE). Das Ziel von MBSE ist ein interdisziplinäres Systemmodell. Die Einsatzbereiche des Modells sowie die Voraussetzungen zur Modellierung werden dabei näher beleuchtet. Zwischen der Systemmodellierung und der Geschäftsprozessmodellierung lassen sich Parallelen erkennen. Die im Rahmen der Prozessmodellierung aufgestellten Grundsätze ordnungsmäßiger Modellierung werden eingeführt und können auf die Systemmodellierung übertragen werden. Aus der Problemabgrenzung werden Anforderungen an das Rahmenwerk abgeleitet.

Die Ergebnisse der Literaturrecherche zum aktuellen **Stand der Technik** werden in **Kapitel 3** vorgestellt. Hierzu findet eine Trennung zwischen den drei Bestandteilen zur Beschreibung eines Systemmodells statt: Modellierungssprache, Vorgehensmethode und Softwarewerkzeuge. Bei den Modellierungssprachen werden nur solche betrachtet, die die Beschreibung der Systemstruktur (als Bestandteil des Systemmodells) adres-

sieren. Die beschriebenen Vorgehensmodelle sind größtenteils auf einzelne Sprachen ausgelegt, wobei in dem Zusammenhang die Unabhängigkeit einer spezifischen Modellierungssprache betont wird. Abschließend werden Softwarewerkzeuge vorgestellt, die zur Modellierung der Systemstruktur verwendet werden. Neben den Sprachen, Methoden und Werkzeugen werden Modellierungsrichtlinien untersucht. Die Funktionsmodellierung ist ein wesentlicher Bestandteil der Beschreibung von Systemen und wird daher ebenfalls näher beleuchtet. Eine zusammenfassende Betrachtung der Strukturmodellierung zeigt die Unterschiede des Struktur-, bzw. Architekturverständnisses in der Literatur. Der Stand der Technik wird hinsichtlich der in Kapitel 2 aufgestellten Anforderungen bewertet und ein Handlungsbedarf aufgezeigt.

Kapitel 4 bildet den Kern der vorliegenden Arbeit mit den Bestandteilen des **Rahmenwerks zur Modellierung einer plausiblen Systemstruktur**. Den Ausgangspunkt des Rahmenwerks bildet die Definition und Klassifikation der zu beschreibenden Elemente und Beziehungen. Die Vorgaben zur Erstellung einer plausiblen Systemstruktur werden in Form von Richtlinien und Bedingungen aufbereitet. Diese bilden die Basis zur Prüfung der Plausibilität. Ausgehend von den Element- und Beziehungsklassen werden Sichten auf die Systemstruktur vorgestellt. Das Konzept einer Werkzeugunterstützung greift die Bestandteile auf und zeigt die computerbasierte Modellierung. Abschließend wird ein Vorgehensmodell zur Erstellung einer plausiblen Systemstruktur dargelegt.

An zwei Beispielen erfolgt in **Kapitel 5** die **Anwendung der Bestandteile** des Rahmenwerks. Das erste Beispiel ist eine Tretkraftunterstützung, die in einem Unternehmen neu entwickelt werden soll, während das zweite Beispiel bereits realisiert ist und im Nachhinein modelliert wird. Die Bewertung der Arbeit erfolgt an den in Kapitel 2 aufgestellten Anforderungen.

Kapitel 6 besteht aus einem **Resümee** der Arbeit sowie einem **Ausblick** auf zukünftige Forschungsfelder. Der **Anhang** enthält ergänzende Informationen zum Rahmenwerk und stellt Hilfsmittel bereit, die in Kapitel 4 vorgestellt wurden.

2 Problemanalyse

Ziel der Problemanalyse sind Anforderungen an ein *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*. Hierzu ist Kapitel 2 wie folgt aufgebaut. Für ein einheitliches Verständnis über verwendete Begrifflichkeiten werden diese in Kapitel 2.1 definiert. Die hier fokussierten mechatronischen Systeme werden in Kapitel 2.2 eingeführt. Wie das Vorgehen der Entwicklung mechatronischer Systeme aussieht und welche Ansätze das Systems Engineering liefert, werden in Kapitel 2.3 und 2.4 vorgestellt. Anschließend wird das Themenfeld Model-Based Systems Engineering intensiv betrachtet (Kapitel 2.5). Übertragen aus der effizienten und effektiven Prozessmodellierung werden Grundsätze ordnungsmäßiger Modellierung vorgestellt (2.6). Abschließend erfolgt eine Problemabgrenzung (Kapitel 2.7) aus der die Anforderungen abgeleitet werden (Kapitel 2.8).

2.1 Begriffsdefinitionen

Die vorliegende Arbeit befasst sich mit der disziplinübergreifenden Beschreibung von Systemen. Der Begriff **System** wird im alltäglichen Sprachgebrauch häufig verwendet und bezieht sich auf ein zusammengesetztes Ganzes: z.B. Wirtschaftssystem, biologisches System, IT-System oder Transportsystem. Im Lexikon finden sich dazu unterschiedliche Bedeutungen. Unter anderem wird ein System verstanden als eine

„Einheit aus technischen Anlagen, Bauelementen, die eine gemeinsame Funktion haben“ [Dud13-ol].

Etwas allgemeiner wird der Begriff System in der Naturwissenschaft genutzt:

„Gesamtheit von Objekten, die sich in einem ganzheitlichen Zusammenhang befinden und durch die Wechselbeziehungen untereinander gegenüber ihrer Umgebung abzugrenzen sind“ [Dud13-ol].

Damit besteht ein System aus einer Ansammlung miteinander in Beziehungen stehender Teile [Pat82]. Es ist durch eine willkürlich gesetzte Grenze (Systemgrenze) von seiner Umgebung getrennt, wobei die Elemente mit der Umgebung in Beziehung stehen. Jedes Element kann wiederum für sich betrachtet ein System darstellen und damit in Subelemente unterteilt werden [HWF+12] (Bild 2-1).

Im Rahmen dieser Arbeit wird der Begriff System synonym für das technische Produkt – im speziellen mechatronische Produkt – verwendet. Auf die Besonderheiten mechatronischer Systeme wird im anschließenden Kapitel eingegangen.

- **Produktiv:** Das Modell soll dem Modellierungszweck entsprechen. Damit wird vom Modell verlangt, brauchbare Antworten zu liefern.
- **Handhabbar:** Es muss eine leichte Anwendbarkeit und Interpretierbarkeit des Modells gegeben sein.
- **Nicht aufwendig:** Der Aufwand für die Erstellung, Anwendung und Pflege des Modells soll möglichst gering sein.

Da die ersten drei und die letzten beiden Anforderungen gegensätzlich sind, muss ein Kompromiss gefunden werden. Dies führt häufig zu Missverständnissen mit Modellen, da die erreichten Ergebnisse häufig nicht den subjektiven Vorstellungen entsprechen [Pat82].

Das Modell kann rechnerintern erstellt werden. Hierzu werden Diagramme verwendet, die eine Oberfläche zur graphischen Modellierung bereitstellen. Die spezifizierten Modellinhalte werden dabei formalisiert abgespeichert (Bild 2-2). Daraus ergibt sich eine Datenstruktur, die in der Literatur auch als Modell/Datenmodell bezeichnet wird [Alt12b]. Zur besseren Trennung dieser Begrifflichkeiten, wird in dieser Arbeit für das Datenmodell die Bezeichnung **Repository**, bzw. **Datenbank** verwendet.

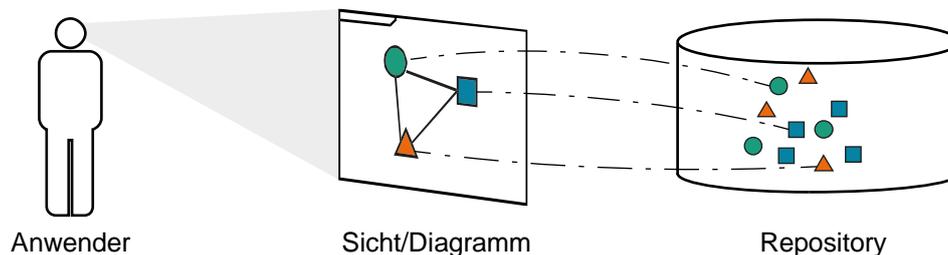


Bild 2-2: Trennung zwischen Diagramm und Repository (nach [Alt12b])

Mit Modellen wird das System aus verschiedenen Betrachtungsweisen beschrieben (vgl. Kap. 2.4.1). Jede Betrachtungsweise ist ein **Aspekt** des Systems, wie z.B. Struktur oder Verhalten. Für jeden Aspekt kann ein eigenes Modell erstellt werden, das als **Partialmodell** bezeichnet wird. Die Summe aller Partialmodelle mit den aspektübergreifenden Verlinkungen zwischen den Modellelementen entspricht dem **Systemmodell**. Die formalisierte Spezifikation des Systemmodells erfolgt im Repository.

Die Modellinhalte eines Partialmodells können dem Anwender vollständig oder über definierte **Filter** angezeigt werden [Neg06]. Die so gefilterten Inhalte entsprechen einer **Sicht** des Aspekts. So stellt die Hierarchie einen Filter dar. Dieser definiert die Hierarchieebenen und die je Ebene enthaltenen Elemente. Die Betrachtung einer Ebene ist damit eine Sicht innerhalb eines Aspekts. Es sind auch aspektübergreifende Filter denkbar. Diese stellen dem Anwender ausgewählte Inhalte des Systemmodells dar, die zu

mehr als einem Aspekt gehören. Im Folgenden wird in diesem Fall das Adjektiv „aspektübergreifend“ verwendet.

2.2 Mechatronische Systeme

Der Begriff Mechatronik wurde in den 70er Jahren stark geprägt [Ise02]. Das Kunstwort setzt sich zusammen aus **Mechanik** (Maschinenbau) und **Elektronik** (Elektrotechnik). Durch den Begriff wird damit bereits das interdisziplinäre Zusammenwirken klassischer Ingenieurwissenschaften zum Ausdruck gebracht [Jan10], [Rod12]. Die vielen bestehenden Definitionen des Begriffs unterstreichen die Synergie der Fachdisziplinen, erweitern diese jedoch um die Informationstechnik. In der VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ wird die Definition von HARASHIMA, TOMIZUKA und FUKUDA zu Grunde gelegt [HTF96] und aus dem englischen Original übersetzt:

„Mechatronik bezeichnet das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung.“ [VDI2206]

Mechatronische Systeme bestehen in der einfachsten Form aus Sensoren, Aktoren, Informationsverarbeitung und Grundsystem. Diese vier Bestandteile wirken miteinander in einem Regelkreis, der im Bild 2-3 dargestellt ist [VDI2206].

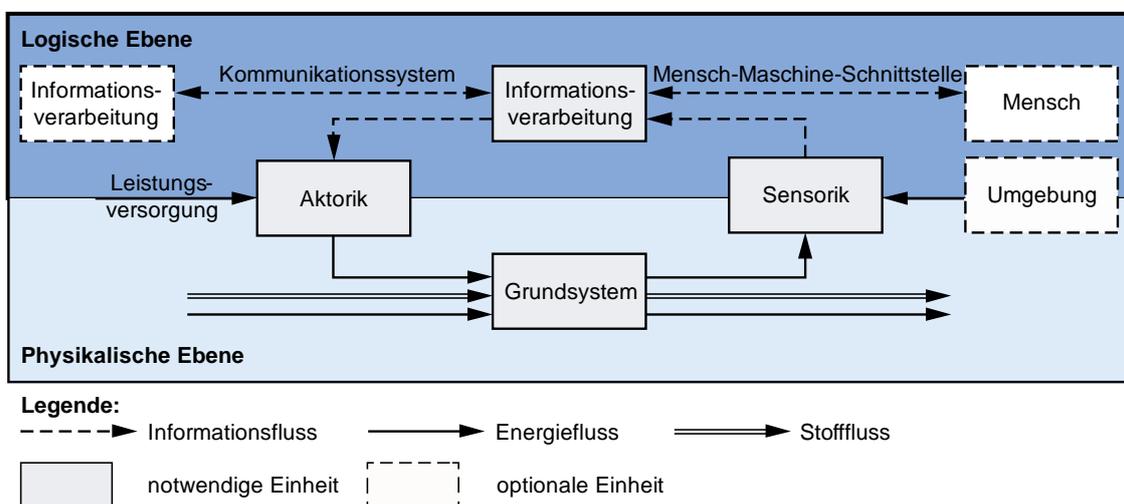


Bild 2-3: Grundstruktur eines mechatronischen Systems [VDI2206]

Zu dem **Grundsystem** zählen alle mechanischen, elektromechanischen, hydraulischen und pneumatischen Strukturen. Durch technologische Weiterentwicklung sind auch Kombinationen aus diesen Strukturen möglich. Zur Bestimmung von physikalischen Größen oder deren Änderungen werden **Sensoren** benötigt. Sie arbeiten als Messwert-aufnehmer und wandeln beliebige physikalische/chemische Größen in elektrische Grö-

Ben. Daraus können Informationen aus der Umgebung und dem inneren Zustand des Systems erfasst werden [Rod12]. Die Sensorsignale sind Eingangsgrößen der **Informationsverarbeitung**. Die Einwirkungen auf das System werden verarbeitet und die Stellgrößen bestimmt, die das Grundsystem auf gewünschte Weise beeinflussen. Die Einwirkung auf das System erfolgt jedoch nicht ausschließlich über Sensoren. Eine Beeinflussung durch den Benutzer kann über die Mensch-Maschine-Schnittstelle erfolgen. Aufgabe der **Aktoren** ist das Wandeln der Stellgröße in eine physikalische Größe zur Erzeugung von Bewegung, Ausüben von Kräften oder Leisten von Arbeit [Czi08], [VDI2206].

Mechatronische Systeme kennzeichnet die Eigenschaft, Stoffe, Energie und/oder Information zu wandeln, zu transportieren und/oder zu speichern [Czi08]. Dies wird durch die Wechselwirkung des Grundsystems mit den Sensoren, Aktoren und Informationsverarbeitung realisiert. Mittels Flussbeziehungen (Stoff, Energie, Information) kann die Interaktion dargestellt werden [PBF+07].

2.2.1 Klassen mechatronischer Systeme

Erzeugnisse des Maschinenbaus und verwandter Branchen wie der Automobiltechnik, Bahntechnik und Medizintechnik waren lange Zeit durch mechanische Systeme geprägt. Die Anzahl an mechanischen Komponenten überwog bei diesen Systemen, so dass die Entwicklung auch von einer Disziplin ausschlaggebend beeinflusst wurde: Konstruktion. Die Integration von elektronischen und softwaretechnischen Elementen nimmt stetig zu (Bild 2-4). Der Wandel zu mechatronischen Systemen ist bereits vollzogen, auch wenn die Entwicklung mechatronischer Systeme nach wie vor eine Herausforderung darstellt [Ben05].

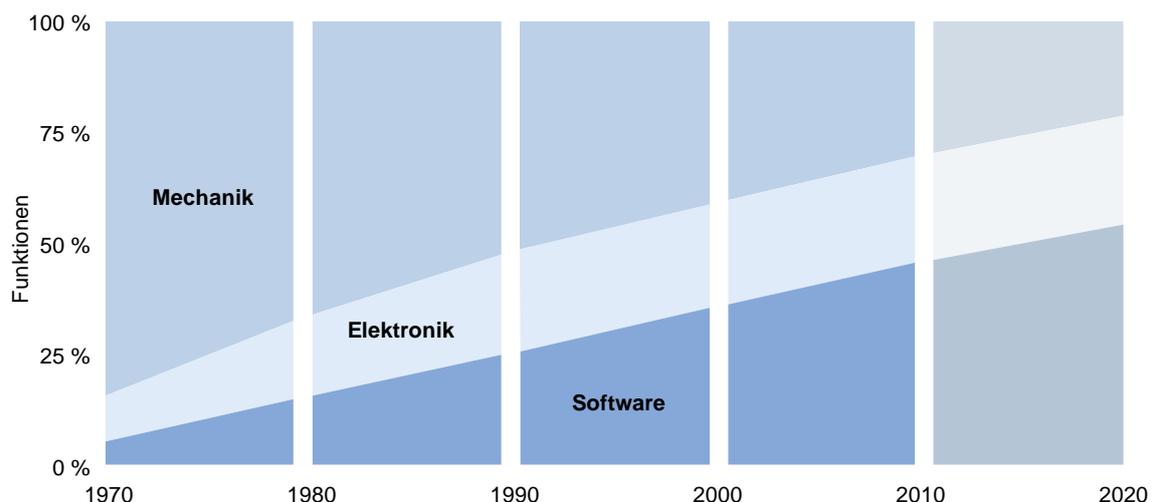


Bild 2-4: Zunahme der Elektronik- und Software-Funktionen im technischen Produkt (nach [SB09])

Die Entwicklung in den Bereichen der Informations- und Kommunikationstechnik deutet darüber hinaus einen weiteren Wandel an. Die mechatronischen Systeme erhalten eine inhärente Teilintelligenz, mit der neue Funktionen realisiert werden können. Der Weg führt zu Intelligenten Technischen Systemen, die durch vier Eigenschaften charakterisiert werden [Dum11], [GAC+13]:

- **Adaptiv:** In der Entwicklung werden Situationen und Änderungen des Systems vorausgedacht und in der Umsetzung berücksichtigt. Dadurch kann sich das System zur Laufzeit auf gegebene Umgebungsbedingungen autonom anpassen und sich weiterentwickeln.
- **Robust:** In Situationen, die vom Entwickler nicht berücksichtigt wurden, kann das System im gewissen Rahmen weiter agieren. Dabei werden Unsicherheiten und fehlende Informationen ausgeglichen.
- **Vorausschauend:** Einflüsse werden analysiert und aus Erfahrungswissen können künftige Wirkungen antizipiert werden. Damit können auch Gefahren frühzeitig erkannt werden. Strategien zur Bewältigung von Gefahren wurden im Entwicklungsprozess implementiert und werden vom System auf Basis der erkannten Situation autonom ausgewählt.
- **Benutzungsfreundlich:** Die Systeme stehen in einer Interaktion mit dem Benutzer und passen sich dem Benutzerverhalten an. Dabei ist deren Verhalten für den Benutzer stets nachvollziehbar.

Die Rechenleistung der Intelligenten Technischen Systeme nimmt stetig zu und ermöglicht eine Vernetzung der Systeme über globale Netze wie das Internet. Hierzu müssen sich die Systeme in der Kommunikation öffnen, um so gegenseitig Daten und Dienste auszutauschen. Diese Systeme werden unter dem Begriff „Cyber-Physical Systems“ (CPS) geführt. Daraus lassen sich vielfältige Anwendungsfelder generieren, da neue Funktionen realisiert oder Funktionen verbessert werden können, die bislang ein einziges System für sich nicht leisten konnte. Die Herausforderungen in der Entwicklung dieser Systeme liegen unter anderem in den Echtzeitanforderungen der technischen Systeme und Offenheit der Kommunikationssysteme. Das intelligente Stromnetz, das sogenannte *Smart Grid*, ist ein Beispiel für derartige Systeme. Durch die Vernetzung von Verbrauchern und Betriebsmitteln der Elektrizitätsversorgung kann eine Optimierung und Überwachung der Energieversorgung realisiert werden. Ziel ist Ressourceneinsparung und effizienter Systembetrieb [aca09].

Mit den Intelligenten Technischen Systemen ergibt sich eine Bandbreite mechatronischer Systeme, die in drei Klassen unterteilt werden (Bild 2-5). Systeme der **Klasse 1** sind mechanisch-elektronische Baugruppen, die auf räumlicher Integration von Mechanik und Elektronik beruhen. Mit diesen Systemen werden Ziele wie die Miniaturisierung sowie Funktionsintegration verfolgt. Schwerpunkte in der Entwicklung liegen in der Aufbau- und Verbindungstechnik. Fokus der **Klasse 2** liegt bei der Optimierung und

Erweiterung des kontrollierten Bewegungsverhaltens von Mehrkörpersystemen. Im Vordergrund steht der Entwurf der Regelung als Basis für das kontrollierte Bewegungsverhalten [Gau10]. Durch die zunehmende Vernetzung der Systeme zeichnet sich eine weitere Klasse ab: **Klasse 3** – Intelligente, vernetzte Systeme. Diese sind in der Lage, sich auf ihre Umgebung und die Wünsche der Anwender flexibel anzupassen. Größtenteils sind sie geografisch verteilt und agieren im Verbund. Das System als Ganzes aufzufassen und die Komplexität zu beherrschen sind die Treiber der Entwicklung dieser Systeme. Daher liegen die vorrangigen Aufgaben im Systems Engineering.



Bild 2-5: Klassen mechatronischer Systeme [GAC+13]

2.2.2 Komplexität der Systeme

Der Begriff Komplexität wird in Zusammenhang mit modernen technischen Systemen häufig genannt. Dabei ist zwischen komplex und kompliziert [Pat82], [HWF+12] sowie zwischen der Produkt- und Prozesskomplexität zu unterscheiden [BHL07-01]. Nach Bild 2-6 lässt sich die Unterscheidung kompliziert und komplex auf Basis der Anzahl an Elementen und deren Beziehungen sowie der Veränderbarkeit dieser festlegen. Demnach sind Systeme einfach, massiv vernetzt, dynamisch oder komplex [Pat82], [UP95], [HWF+12].

Einfache Systeme bestehen aus einer geringen Anzahl an Elementen und Beziehungen. Dabei sind die Elemente fest und permanent miteinander verbunden. Steigt die Anzahl an Elementen und Beziehungen, werden die Systeme kompliziert und zeichnen sich durch ihre **massive Vernetzung** aus. Die Beziehungen sind jedoch statisch. **Dynamisch, komplizierte Systeme** bestehen aus Verbindungen, die sich zeitlich verändern. Die Veränderung ist meist nicht linear und äußert sich z.B. in der Intensität, Struktur, oder der Art der Wechselwirkung. Von **komplexen Systemen** wird gesprochen, wenn

eine hohe Anzahl, Vielfalt sowie Größe der Elementen und Beziehungen vorliegt und deren Verbindungen dynamisch sind [HWF+12].

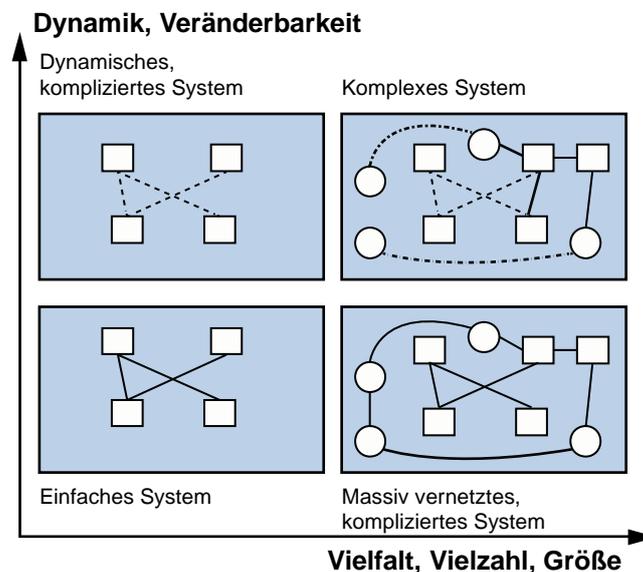


Bild 2-6: Systemtypen nach [HWF+12] und [UP95]

Mechatronische Systeme zeichnen sich durch ihre Vielfalt der einzelnen Elemente (elektrische, softwaretechnische, mechanische Elemente) und Beziehungen aus, die wiederum eine Dynamik besitzen. Demnach fallen mechatronische Produkte in die Kategorie der komplexen Systeme. Nach einer Studie der TU München, Lehrstuhl Prof. Lindemann, wurde Produktkomplexität in den befragten Unternehmen durch folgende Merkmale charakterisiert [BHL07-01]:

- steigende und zeitlich verändernde Anforderungen,
- Vielzahl an Funktionen und Elementen,
- große Anzahl an Abhängigkeiten und Schnittstellen sowie
- Zusammenwirken von Elementen der verschiedenen Fachdisziplinen.

In der Studie wird zudem deutlich, dass die Produktkomplexität in den folgenden Jahren weiter ansteigen wird. Diese Einschätzung wurde von den befragten Personen durchgehend bestätigt. Dies ist mit dem Trend zu intelligenten, vernetzten Systemen angedeutet (Kapitel 2.2.1). Durch die Vernetzung steigt die Anzahl an Elementen, die dynamisch in Interaktion stehen.

Die Produktkomplexität führt direkt zu einer Komplexität in der Entwicklung dieser Systeme und äußert sich in einer Prozesskomplexität. Die Vielfalt an beteiligten Disziplinen und die Dynamik in der Interaktion der Entwickler zeigt die Eigenschaften eines komplexen Entwicklungsprozesses [BHL07-01].

2.3 Entwicklungsmethodik für mechatronische Systeme nach VDI 2206

Wie im vorherigen Kapitel eingeführt, beruhen mechatronische Systeme aus der Symbiose der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik. Jede Disziplin für sich deckt ein breites Spektrum ab: Maschinenbau umfasst z.B. die Konstruktion, Dynamik sowie Kinematik. Aber auch Hydraulik und Pneumatik werden in diesen Bereich eingeordnet. Die Elektrotechnik hingegen deckt unter anderem die Bereiche Antriebstechnik, Leistungselektronik, Mikroelektronik sowie Energietechnik ab. Softwaregestaltung und Automatisierungstechnik sind zwei Bereiche der Informationstechnik [Ise02].

Heutige technische Produkte fallen in die Kategorie mechatronische Systeme und lassen sich damit nicht mehr nur einer Disziplin zuordnen. Dies wirkt sich direkt auf die Vorgehensweise in der Entwicklung aus. Alle beteiligten Fachdisziplinen müssen von Beginn an interdisziplinär zusammenarbeiten. Das erfordert jedoch ein Denken und Verstehen des gesamten Systems über alle Fachdisziplinen hinweg [Jan10].

Das methodische Vorgehen bei der disziplinübergreifenden Entwicklung mechatronischer Systeme wird in der VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ beschreiben [VDI2206]. Dieses orientiert sich an dem sogenannten V-Modell (Bild 2-7).

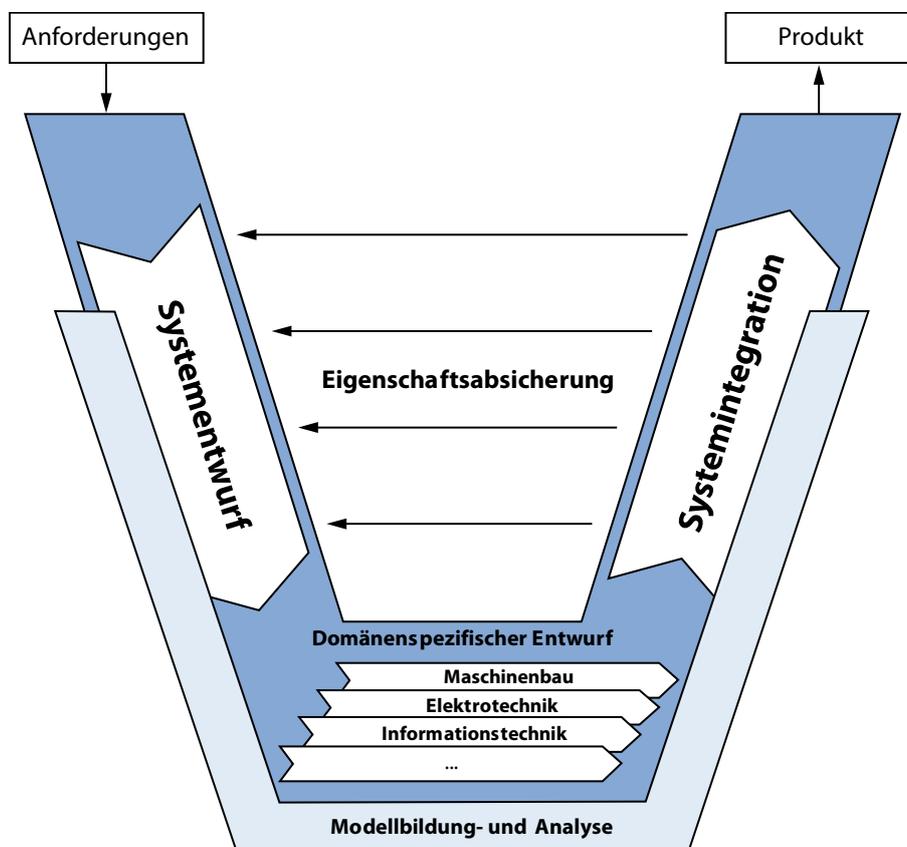


Bild 2-7: V-Modell (nach [VDI2206])

Die Entwicklungsaufgabe beginnt mit einem konkreten Entwicklungsauftrag. Diese wird in Form von **Anforderungen** beschrieben. In jedem anschließenden Prozessschritt ist die Erfüllung der Anforderungen anzustreben und die Ergebnisse dahingehend zu bewerten. Das spätere Produkt muss den Anforderungen im vollen Umfang genügen.

Ausgehend von den Anforderungen wird ein **Systementwurf** durchgeführt. Hierzu erfolgt die Betrachtung der Gesamtfunktion des Systems. Aufgrund der hohen Komplexität der Entwicklungsaufgabe ist eine Lösungsfindung auf Basis der Gesamtfunktion nicht möglich – diese ist in Teilfunktionen zu unterteilen. Die Teilfunktionen werden durch Flussbeziehungen in einer Funktionsstruktur zusammengefasst. Die Richtlinie empfiehlt dabei, sogenannte kanonischen Funktionen (siehe [Hua02]) zu verwenden. Im nächsten Schritt werden Wirkprinzipien gesucht. Diese müssen die Teilfunktionen erfüllen. In der Wirkstruktur werden Wirkprinzipien über Flussbeziehungen mit dem Ziel verknüpft, physikalische Verträglichkeit zwischen den Wirkprinzipien sicherzustellen. Die Baustruktur erweitert die Verträglichkeitsbetrachtung unter Berücksichtigung der Gestalt. Auf Basis dieser Informationen kann jedoch noch keine Bewertung der Lösungsvarianten erfolgen. Hierzu sind Konkretisierungen in entsprechenden Bereichen vorzunehmen, wie z.B. die Analyse von Mehrkörpersystemen oder der Bau von Anschauungsmodellen. Die Lösungsvarianten sind abschließend nach technischen und wirtschaftlichen Kriterien zu bewerten. Das Ergebnis der Phase Systementwurf ist ein disziplinübergreifendes Lösungskonzept.

Auf Basis des Lösungskonzepts erarbeiten die Fachdisziplinen in der Phase **Domänen-spezifischer Entwurf** die Teillösungen. Dies geschieht mit disziplinspezifischen Entwicklungsmethodiken. Die Teillösungen werden anschließend in der **Systemintegration** zum Ganzen zusammengeführt.

Eine Verifikation und Validierung findet durchgehend durch die Absicherung der Eigenschaften statt. Dabei ist die Verifikation das Überprüfen eines Sachverhalts gegenüber Annahmen oder Behauptungen. Im Bereich der Produktentwicklung wird der Begriff Verifikation in zwei verschiedenen Zusammenhängen genutzt. Einerseits kann der Sachverhalt das Produkt sein, so dass das Produkt hinsichtlich der Erfüllung von Anforderungen geprüft wird. Dies kann durch Qualitätsmethoden, mit Simulationsmodellen oder Prüfständen erfolgen. Dabei wird angenommen, dass die aufgestellten Anforderungen (die Spezifikation) richtig sind. Mit der Verifikation wird die Frage beantwortet: „Bauen wir das Produkt richtig?“ [GG05]. Darüber hinaus kann auch ein Modell verifiziert werden. Das Modell entspricht in diesem Fall dem „Sachverhalt“. Dabei wird ein Modell (z.B. ein Simulationsmodell) auf seine Gültigkeit überprüft. Dies geschieht meistens mit Experimenten. Die Messwerte des Experiments werden mit den simulierten Ergebnissen verglichen.

Die Validierung ist die Prüfung der Anforderungen auf ihre Richtigkeit. Damit wird die Frage beantwortet „Bauen wir das richtige Produkt?“. Die Unterteilung zeigt, dass auch bei einem positiven Ergebnis der Verifikation, die Validierung negativ ausfallen kann.

Erst bei der Validierung wird aufgezeigt, dass evtl. Anforderungen die Erwartungen nicht erfüllen. In den meisten Fällen wird Verifikation und Validierung nicht getrennt betrachtet, so dass diese Begriffe stets gemeinsam genannt werden (Verifikation/Validierung oder V&V). Im Bereich der Softwareentwicklung wird der Begriff Testen verwendet, der beides abdeckt [GG05].

2.4 Systems Engineering

Systems Engineering (SE) wird als Sichtweise bzw. als Ansatz verstanden, der Methoden und Prozesse zur Realisierung von erfolgreichen Systemen bereitstellt [INC12]. Die Ursprünge liegen in der Systemtheorie, Modelltheorie und der Kybernetik, die das ganzheitliche Systemdenken und das Modelldenken vorangetrieben haben [Rop09]. Die wesentliche Basis des Ansatzes ist das Systemdenken, mit dem das Verständnis und die Beherrschung komplexer Systeme ermöglicht werden [HWF+12], [Rop09], [INC12]. Eine allgemeingültige Definition, die die wesentliche Kernphilosophie hinter Systems Engineering zusammenfasst, beschreibt HINTCHINS mit

„Systems Engineering is the art and science of creating whole solutions to complex problems” [Hit07, S.91].

Die Anfänge zum Einsatz von Systems Engineering fanden in den 40er Jahren in den Bell Telephone Laboratories statt. Bei der Entwicklung der Telekommunikationssysteme spielten ganzheitliches Systemdenken und ausführliche Analyse der Anforderungen eine wichtige Rolle. Die Konzepte wurden im späteren Verlauf im Wesentlichen im Bereich der Luft- und Raumfahrt sowie dem Militärbereich eingesetzt und weiterentwickelt. Das NASA Apollo Programm beispielsweise zeigte nicht nur durch die Komplexität des Systems die Notwendigkeit auf, sondern vor allem durch die Unkenntnis und Unvorhersehbarkeit des Start- und Landeprozesses [Hit07]. Auch die historischen Ereignisse, wie das Wettrüsten im Kalten Krieg forderten systematisches Vorgehen und schnelle Realisierungen. Diese Einsatzgebiete führen dazu, dass mit Systems Engineering oft große Projekte und Produkte assoziiert werden. Verschiedene Organisationen forcieren die Verbreitung des Ansatzes und sind überzeugt, dass Methoden und Prozesse auf beliebige Größenordnung von Projekten und Produkten übertragen werden können [INC07-ol], [GAC+13], [FK13-ol], [KWH+11-ol].

2.4.1 SE-Bestandteile nach HABERFELLNER et al.

HABERFELLNER et al. betrachtet SE als eine methodische Komponente bei der Problemlösung auf [HWF+12] und definiert Konzepte, die sich in der Struktur nach Bild 2-8 darstellen lassen. Zur SE-Philosophie gehören das SE-Vorgehensmodell und das Systemdenken. Die eigentliche konstruktive Tätigkeit zur Lösungsfindung erfolgt in der Systemgestaltung. Das Projektmanagement stellt das Rahmenwerk für die Organisation und Koordination des Problemlösungsprozesses.

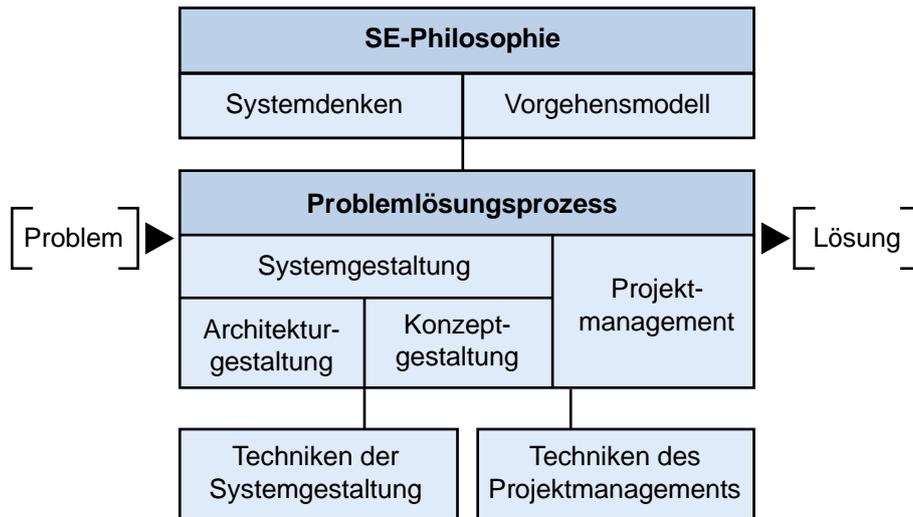


Bild 2-8: Das Systems Engineering-Konzept nach HABERFELLNER et al. [HWF+12]

Das **Systemdenken** liefert Denkansätze, die es ermöglichen, komplexe Erscheinungen (Systeme) besser zu verstehen und gestalten zu können. Dabei ist das System ganzheitlich zu betrachten. Durch unterschiedliche Blickwinkel wird das System im Kontext einer Problemstellung oder Situation beleuchtet und die dafür notwendigen Zusammenhänge aufgezeigt. HABERFELLNER et al. schlägt hierzu die drei Betrachtungen umgebungsorientiert, wirkungsorientiert und struktuorientiert vor [HWF+12].

Die Prinzipien des Systemdenkens werden durch modellhafte Abbildungen unterstützt. Dieser Ansatz wird im Model-Based Systems Engineering verfolgt und wird im Kapitel 2.5 ausführlich dargestellt.

Der zweite Teil der SE-Philosophie – das **SE-Vorgehensmodell** – verfolgt vier Grundgedanken. Mit einem Top-Down-Vorgehen wird vom *Groben zum Detail* das Betrachtungsfeld (Problemfeld, Ausgangssituation, Entwurf von Lösungen) zu Beginn weiter gefasst und erst zum Schluss schrittweise eingengt. Durch das *Denken in Varianten* sollen verschiedene Konzepte als Lösungsmöglichkeit in Betracht gezogen werden. Ein *Phasenablauf* überführt die Systementwicklung in eine zeitliche Reihenfolge, während bei dem *Problemlösungszyklus* ein formaler Vorgehensleitfaden beim Lösen von Problemen unterstützt [HWF+12]. Das V-Modell der VDI-Richtlinie 2206 ordnet sich in den Bereich der SE-Vorgehensmodelle ein und stellt einen Standard in der Mechatronikentwicklung dar.

2.4.2 Einsatz und Nutzen von Systems Engineering

Vor dem Hintergrund steigender Qualitätsanforderungen sowie kürzeren Entwicklungszeiten bei geringem Kosteneinsatz wird das Potential des SE-Ansatzes in den frühen Entwicklungsphasen gesehen [INC12]. Bild 2-9 verdeutlicht die Motivation. In den frühen Phasen fallen nur geringe Kosten auf die Konzeption, Gestaltung und Entwicklung des Produkts an. Lediglich 20% der Lebensdauerkosten entstehen in dieser Phase.

Demgegenüber sind die Kosten aufgeführt, die zu den entsprechenden Phasen festgelegt werden. Damit wird ersichtlich, dass die meisten Kosten bereits im Produktkonzept entschieden werden.

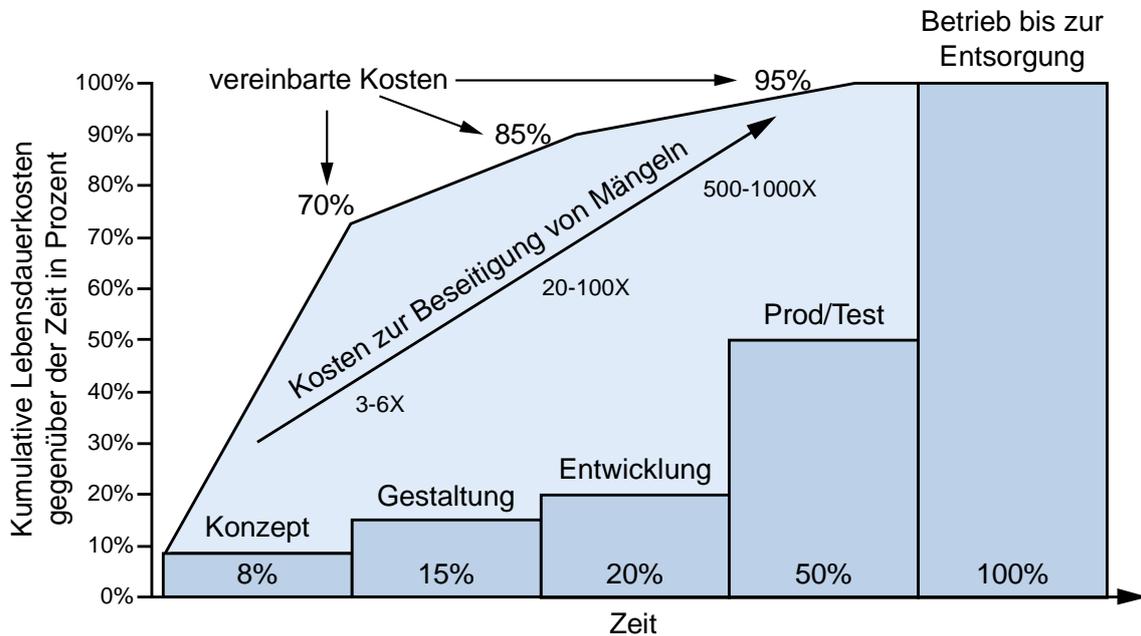


Bild 2-9: Kumulative Lebensdauerkosten über die Zeit [INC12]

Das Wissen über das Produkt ist in der Konzeptphase meistens sehr gering und steigt erst im Verlauf stetig durch Konkretisierung und Verifikation an. Demnach werden nach Bild 2-9 bereits 70% der späteren Kosten auf einer geringen Wissensgrundlage entschieden [INC12], [BVB12]. Fehler, die im späteren Verlauf identifiziert werden und damit Änderungen nach sich ziehen, sind meist aufwändig und kostenintensiv („rule of ten“) [BGJ+09].

Mit den Methoden des Systems Engineering soll die frühe Phase stärker unterstützt werden, um so eine bessere Basis zur Entscheidung von Konzepten zu erhalten [INC12], [Gau10]. Unterschiedliche Ansätze können hierzu genutzt werden, die teilweise unter dem Begriff „Frontloading“ geführt werden [Jür00]. Im Zusammenhang mit Systems Engineering werden folgende Aufgabenbereiche genannt [SA00], [INC12], [Wei06], [FMS12], [GCW+13]:

Anforderungsdefinition, -analyse und -management: Das breite Feld rund um die Anforderungen soll die erfolgreiche Realisierung des Qualitätsprodukts erreichen, das auf die Nutzerbedürfnisse zugeschnitten ist [INC12]. Bei der Anforderungsdefinition sollen alle Stakeholder berücksichtigt werden, um so den Nutzen des Produkts zu erfassen und bestmöglich umzusetzen. Zudem ist dabei der gesamte Produktlebenszyklus zu betrachten. Neben der eigentlichen Betriebsphase sind Anforderungen aus der Fertigung, Montage, Logistik, Vertrieb, Service und Wartung bis zur Entsorgung zu identifizieren und zu berücksichtigen.

Identifizierte Anforderungen sind im Entwicklungsprozess soweit zu pflegen, dass stets

eine Rückverfolgbarkeit möglich ist. Damit können Auswirkungen der Änderungen frühzeitig erkannt und der Aufwand abgeschätzt werden.

Systemdesign und -analyse: Für die Realisierung des Produkts sollen verschiedene Konzeptvarianten betrachtet werden. Dieser Ansatz findet sich auch im SE-Vorgehensmodell nach HABERFELLNER et al.: Denken in Varianten [HWF+12]. Dabei wird die Festlegung auf ein Konzept nach hinten verlagert, damit die bestmögliche Umsetzung auf fundierter Basis ausgewählt werden kann. Für die Konzipierung sind abstrahierte Beschreibungsmittel zu wählen, mit denen sich Anforderungen, Struktur und Verhalten abbilden lassen, jedoch der Aufwand zur Beschreibung gering gehalten werden kann. Die fachdisziplinübergreifende Systembetrachtung legt dabei großen Wert auf die Schnittstellenbetrachtung. Diese betreffen die Schnittstellen des Systems nach außen sowie die Schnittstellen der Subsysteme [BVB12]. Die Entwicklung von Subsystemen erfolgt teilweise disziplinspezifisch. Die ausführliche Schnittstellenbetrachtung und Zuweisung von Verantwortlichkeiten soll eine reibungslose Integration sicherstellen (der Subsysteme zum System als auch des Systems in seine Umgebung).

Die Konzepte sollen mit Methoden der Virtualisierung (Modellbildung und Simulation, vgl. Kap. 2.5.1) weiteren Aufschluss über die zu entwickelnden Produkte geben. Damit wird weiteres Wissen generiert und die Entscheidungsgrundlage verbessert.

Verifikation und Validierung: Mit der Verifikation wird eine frühzeitige Absicherung der gestellten Anforderungen angestrebt. Hierbei kommen Ansätze des Qualitätsmanagements zum Einsatz. Mit Methoden der Zuverlässigkeitsbewertung, wie z.B. der FMEA (Failure Mode and Effects Analysis) oder dem QFD (Quality-Function-Deployment) sollen Fehler frühzeitig identifiziert und behoben werden. Neben diesen qualitativen Methoden werden virtuelle Modelle zur Verifikation und Validierung genutzt. Mit dem virtuellen Testen können Erkenntnisse gewonnen und wiederum Fehler im Entwurf erkannt und beseitigt werden.

Organisation und Koordination: Dieses übergeordnete Thema adressiert alle Bereiche der Organisation des Entwicklungsprozesses. Durch die Interdisziplinarität müssen die Fachdisziplinen koordiniert werden. Dabei werden Ansätze generiert, die die Zusammenarbeit und die Kommunikation zwischen den Disziplinen verbessert. Zusätzlich werden typische Themen des Projektmanagements betrachtet, wie Qualität, Kosten und Zeit.

Der **Nutzen von Systems Engineering** ist schwer zu erfassen. Durch ein heterogenes Verständnis zum Themenfeld Systems Engineering fällt die Umsetzung in den Unternehmen schwer. In der Studie [GCW+13] von GAUSEMEIER et al. wurde der Leistungsstand des Systems Engineerings im deutschsprachigen Raum untersucht. Hierbei zeigten die geführten Interviews, dass in der Praxis das Verständnis für SE wenig ausgeprägt ist und teilweise als populärer Begriff abgetan wird. Dennoch konnten Nutzenpotentiale identifiziert werden, die SE zugeschrieben werden: Orchestrierung der disziplinübergreifenden Zusammenarbeit, Berücksichtigung der Bedürfnisse aller rele-

vanter Stakeholder, verbesserte Planungs- und Steuerungssicherheit, Qualitätssicherung und -steigerung sowie Wiederverwendung von Lösungswissen.

Qualitative Studien zur Nutzenbewertung von SE liegen im Bereich der Entwicklung softwareintensiver Systeme [Hon04]. Hier wurden Kosten- und Zeitüberschreitungen von Projekten in Korrelation mit SE-Aufwand untersucht. Dabei zeigte sich, dass eine Verringerung der Überschreitungen ersichtlich wurde, je mehr SE-Aufwand betrieben wurde. Als Resultat wird ein optimaler SE-Aufwand auf ca. 15-20% des Projektaufwands genannt.

2.4.3 Rollen im Systems Engineering

Die Umsetzung von SE-Prozessen verlangt neue Rollen. Eine entscheidende Rolle nimmt der sogenannte **Systems Engineer** ein. Dieser begleitet das Projekt durchgängig neben dem Projektmanager [Wei06]. Der Systems Engineer hat eine technische Sicht auf die Entwicklung, wobei dieser kein Spezialist in einer Disziplin ist. Vielmehr ist er der Vermittler zwischen den Fachdisziplinen, er hat stets das System als Ganzes und die Schnittstellen im Blick [BVB12]. Hierzu muss er ein technisches Grundwissen besitzen, sowie analytisch und systematisch arbeiten können. Nach KOSSIAKOFF et al. ist der Systems Engineer ein guter Problemlöser und Kommunikator [KS03]. Ähnliche Aufgaben und Anforderungen werden im SE-Kontext dem **Systemarchitekten** zugesprochen. Dieser wirkt jedoch nicht nur als Vermittler, sondern ist auch für die Architekturgestaltung verantwortlich. Damit muss er den Kundennutzen verstehen, diesen in Systemfunktionen übersetzen und daraus Konzepte erarbeiten [HWF+12].

SHEARD unterteilt die SE-Aufgaben granularer und ordnet sie den Rollen zu [She96]. Daraus sind 12 Rollen entstanden, die in den Aufgabenbereiche Anforderungen, Design, V&V und Organisation (vgl. Kap. 2.4.2) wie folgt verteilt sind.

Die Aufgabenbereiche rund um die Anforderungen übernimmt der **Requirements Owner (RO)**. Damit ist er verantwortlich für die Überführung von Kundenbedürfnissen in Anforderungen, prägnante Formulierung und Pflege der Anforderungen sowie Sicherstellung der Rückverfolgbarkeit bei Änderungen. Er kommuniziert anfallende Änderungen an entsprechende Verantwortliche und kann die Auswirkung von Änderungen auf Teilsysteme und Gesamtsystem abschätzen. Die Kundensicht vertritt dabei die **Customer Interface (CI) Rolle**. Sie stellt sicher, dass die Kundenwünsche bestmöglich umgesetzt werden.

Nach den Anforderungen wird eine Systemarchitektur erstellt, die das System mit seinen Subsystemen beschreibt. Dies wird von dem **System Designer (SD)** durchgeführt. Dabei setzt er auf der funktionalen Beschreibung des Systems an. Die Analyse der Systemeigenschaften übernimmt der **System Analyst (SA)**. Hierzu nutzt er Modelle und Simulationen zur Überprüfung. Die **Glue (G) Rolle** fungiert als Integrator und fokus-

siert hauptsächlich die Schnittstellen im System (außen und innen). Damit ist sie verantwortlich, dass die Subsysteme zum System integriert werden können. Hierzu muss sie die Schnittstellen kennen und Auswirkungen von Änderungen in einem Subsystem auf das nächste nachvollziehen und vermitteln können. Die **Logistics and Operations (LO) Rolle** betrachtet vor allem die Betriebs- und Servicephase des Systems. Damit ist sie verantwortlich für die Bedienungsanleitung und Schulungsmaterial. Die Rolle beantwortet alle Fragen zum System im Wartungsfall.

Die **Verification/Validation (VV) Rolle** verantwortet die Aufgaben zur Absicherung der Systemeigenschaften. Hierzu erstellt sie Tests und ist Ansprechpartner, wenn im Testfall ein von der Spezifikation abweichendes Systemverhalten auftritt.

Im Bereich des Projektmanagements werden vier verschiedene Rollen aufgezeigt. Die technischen Abläufe werden von der **Technical Manager (TM) Rolle** verwaltet. Ihre Aufgabe ist die Planung und Koordination der technischen Ressourcen (Rechner, Tools, Versuchsaufbau). Das Daten- und Informationsmanagement übernimmt die **Information Manager (IM) Rolle**. Die Dokumentation, Analyse und Verbesserung von Abläufen liegt bei dem **Process Engineer (PE)**, während die eigentliche Koordination der **Coordinator (CO)** übernimmt. Dieser greift ein, wenn es Meinungsverschiedenheiten zwischen den Disziplinen gibt. Er hat eine moderierende Funktion und soll die Gruppendynamik fördern.

Die letzte Rolle ist aus der Notwendigkeit entstanden, darzustellen, was unter Stellenanzeigen zum „Systems Engineer“ zu der Zeit verstanden wurde. Sie nennt die Rolle **Classified Ads Systems Engineering (CA)**. Diese Rolle entspricht viel mehr einem Software Engineer, der neben fundierten Programmierkenntnissen das gesamte System (Computersystem) überblicken kann.

Die verschiedenen Rollen zeigen auf, welche Aufgabenbereiche Systems Engineering abdeckt. Wie die Rollen im Unternehmen besetzt werden, d.h. ob jede Rolle von jeweils einer Person eingenommen wird oder mehrere Rollen zusammengefasst werden, hängt vom Unternehmen und der Projektgröße ab. MÖHRINGER zeigt beispielsweise in [Möh12] auf, wie in einem mittelständischen Unternehmen die Rollen auf bestehende Strukturen und Positionen verteilt werden.

2.5 Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE) liefert Werkzeug mit denen das Systemdenken ermöglicht wird. Innerhalb der Produktentwicklung werden viele Modelle erstellt, die einem unterschiedlichen Modellierungszweck dienen. Diese werden zu Beginn aufgeführt, um eine Abgrenzung zum MBSE herauszustellen. Anschließend wird das Handlungsfeld von MBSE aufgezeigt und auf die einzelnen Bestandteile eingegangen.

2.5.1 Modellbildung in der Produktentwicklung

Im Entwurf technischer Systeme werden Modelle vielseitig eingesetzt [Lin09], beginnend mit einfachen Handskizzen, die eine abstrahierte Abbildung darstellen. Innerhalb der einzelnen Fachdisziplinen werden Modelle genutzt, um komplexe Sachverhalte zu begreifen, darzustellen und diese zu analysieren. Hierzu werden weitestgehend graphische Modelle genutzt, da diese viele Informationen visuell greifbar machen, die sonst in Textform beschrieben werden müssten. Darüber hinaus bieten die graphischen Modelle eine Möglichkeit, einheitlich über den Sachverhalt zu kommunizieren. Beispiele einheitlicher Darstellungen sind die Schaltpläne im Bereich der Elektronik oder Hydraulik. Auch die 3D-Gestaltmodelle bilden das Produkt abstrahiert ab. Innerhalb dieser Disziplinen werden die Modelle nach festgelegten Regeln erstellt, so dass die Modelle einheitlich, eindeutig und gut verständlich sind. Es existieren graphische Elemente, die innerhalb einer Disziplin gleich interpretiert werden. Damit dient z.B. ein Schaltplan als Kommunikationsmittel zwischen den Elektrikern. Diesen auszutauschen und darüber zu kommunizieren ist einfacher, als den Sachverhalt in einem Text zu beschreiben, der wiederum unterschiedlich interpretiert werden kann.

Die graphische Darstellung der Modelle kann auf einem Blatt Papier erfolgen oder aber auch rechnerintern durch spezifische Softwarewerkzeuge (Bild 2-10). Den Vorteil der rechnerinternen Modellbildung kann am Beispiel der CAD⁶-Systeme gut veranschaulicht werden. Bevor CAD-Systeme vorhanden waren, wurden technische Zeichnungen auf dem Zeichenbrett, sogenannten Zeichenmaschinen erstellt. Erst die Entwicklung speziell angepasster Softwarewerkzeuge ermöglichte den Schritt zur rechnerunterstützten Konstruktion. Modelle konnten nun am Computer erstellt werden, wodurch weitere wesentliche Vorteile hinzukamen. Von den anfänglichen 2D-Darstellungen entwickelten sich die Softwarewerkzeuge weiter und ermöglichen heute drei dimensionale Darstellungen [GPW09]. Die üblichen technischen Zeichnungen können automatisch ausgeleitet werden. Die rechnerinternen Modelle können darüber hinaus für weitere Zwecke genutzt werden. Auf Basis der Modelle können Analysen durchgeführt werden. Hierzu ist in einigen Fällen ein Export der Daten (des Modells) in entsprechende Analysetools notwendig.

Neben den rechnerinternen Modellen werden in der Produktentwicklung Prototypen erstellt, die wiederum ein Modell, eine abstrahierte Form des Originals, darstellen. An den Prototypen werden z.B. Teilfunktionen umgesetzt sowie Analysen und Tests durchgeführt. Zum Beispiel kann am Prototyp eines Vier-Viertelfahrzeugs die Fahrdynamik eines Fahrzeugs getestet werden.

In der Produktentwicklung hat sich das modellbasierte Vorgehen etabliert – in vielen Fällen werden hierzu Begriffe, wie modellbasierter Systementwurf oder Model-Based Engineering verwendet [VDI2206]. Dabei liegt der Fokus auf den rechnerinternen Mo-

⁶ Computer Aided Design

dellen. Hierdurch soll es möglich werden, den aufwendigen Prototypenbau zu ersetzen und die damit verbundenen Versuche am Rechner abzudecken [GPW09].

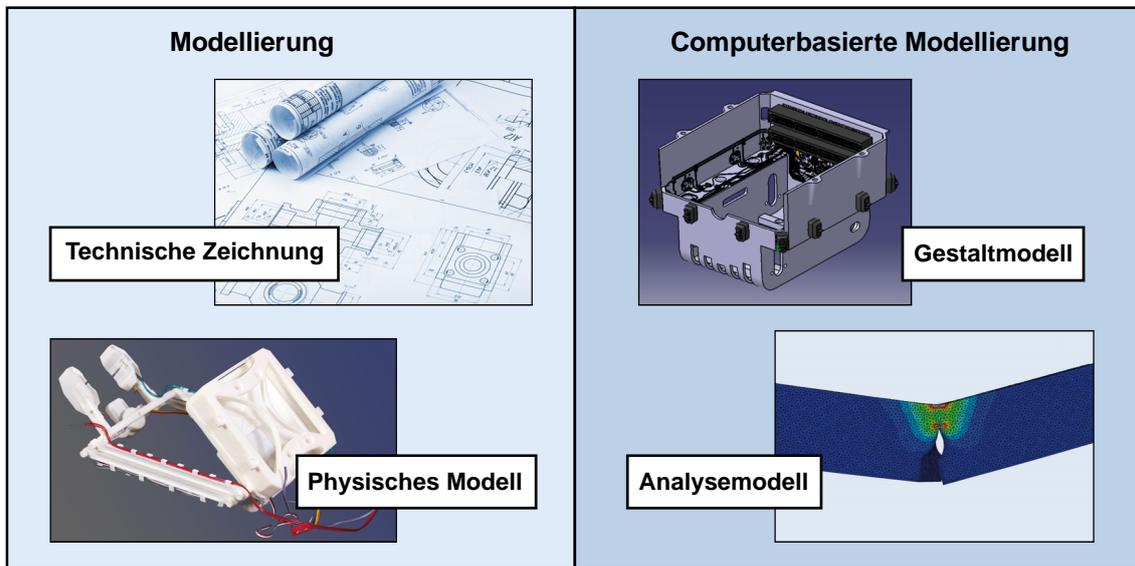


Bild 2-10: Modellbildung in der Konstruktion – von Papier und Prototypen zur computerbasierten Darstellung und Analyse

Bei der Modellbildung ist immer zu hinterfragen, zu welchem Zweck das Modell erstellt wird. Innerhalb einer Disziplin wird ein Subsystem konzipiert und ausgestaltet, wie z.B. das Gehäuse (Konstruktion), der Schaltplan (Elektrotechnik), das Platinen-Layout (Elektrotechnik), die Softwarearchitektur (Softwaretechnik). Darüber hinaus werden disziplinübergreifende Modellierungen mit dem Ziel einer Simulation, bzw. einer Analyse des Systems (z.B. Dynamik, Thermik, Regelung) eingesetzt. Hierzu wurden ebenfalls Modellierungsansätze entwickelt, wozu zum Beispiel Finite Elemente Methode (FEM) oder Mehrkörpersimulation (MKS) zählen [Lin09], [GPW09].

2.5.2 Handlungsfeld MBSE

Innerhalb des Systems Engineering wird das Systemdenken in den Vordergrund gestellt. Damit soll das technische System/das Produkt ganzheitlich und interdisziplinär betrachtet werden. Hier setzen die Konzepte des Model-Based Systems Engineerings an. Ziel ist ein Systemmodell, das in den Mittelpunkt der Entwicklung gestellt wird. Dieses beschreibt das System ganzheitlich in einer Art und Weise, die jede Fachdisziplin gleichermaßen lesen und nachvollziehen kann. Bild 2-11 zeigt die Abgrenzung des Systemmodells gegenüber den üblichen Modellen der Produktentwicklung.

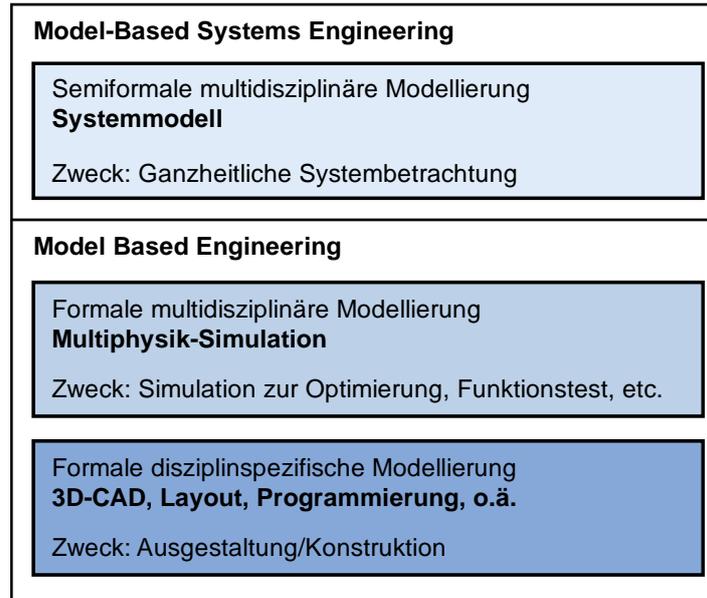


Bild 2-11: Abgrenzung des Systemmodells von den Modellen der Produktentwicklung

Mit dem Systemmodell wird ein Paradigmenwechsel angestrebt, der das weit verbreitete dokumentenbasierte Vorgehen durch modellbasiertes Vorgehen ersetzen soll (Bild 2-12). Dies bezieht sich im Speziellen auf die frühe Phase der Entwicklung. Die Systemspezifikation erfolgt über Anforderungen und textbasierten Beschreibungen, meist verbreitet über mehrere Dokumente. Texte haben den Nachteil, dass sie nicht eindeutig und damit unterschiedlich interpretierbar sind. Sie sind aufwendig in der Pflege und nicht schnell erfassbar. Zudem muss die Sprache, in der der Text formuliert ist, von allen Lesern beherrscht werden. Dies ist bei weltweit verteilten Entwicklungsstandorten oft nicht gegeben.

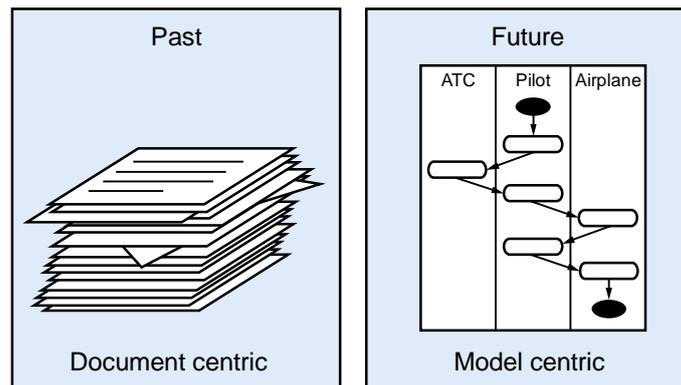


Bild 2-12: Übergang vom dokumentenzentrierten Vorgehen zu MBSE (nach [INC07-ol])

Die Vorteile der graphischen Modelle, die bereits disziplinspezifisch erkannt und gelebt werden (vgl. Kap. 2.5.1 am Beispiel der Schaltpläne), sollen auf Systemebene übertragen werden. Die Systemspezifikation, die sonst in Textform erfolgt ist, wird über graphische Modelle beschrieben. Durch die graphischen Modelle werden die Informationen

einheitlich dargestellt, sind mit dem Auge schnell erfassbar und einfach in der Pflege. Dies gilt jedoch nur im gewissen Maße, da Modelle ab einer bestimmten Anzahl an Elementen und Verbindungen nicht mehr als übersichtlich wahrgenommen werden. Zudem kann, je nachdem mit welchen Werkzeugen das Modell erstellt wird, die Pflege aufwendig sein.

Im Bereich des MBSE wird von einem Modell erst dann gesprochen, wenn dieses eine gewisse formale Form besitzt und damit rechnerunterstützt abgebildet wird [Alt12a]. Durch die rechnerinterne formalisierte Modellierung können die Inhalte vom Rechner automatisch weiterverarbeitet werden.

Zur Beschreibung eines Systemmodells werden daher, neben einer graphischen Modellierungssprache, ein Softwarewerkzeug und eine Methode benötigt (Bild 2-13) [FMS12]. Erst eine auf einander abgestimmte Kombination ermöglicht den effizienten Einsatz der Systemmodellierung in einem Unternehmen. Dabei ist die Modellierungssprache isoliert betrachtet nur ein Ausdrucksmittel. Wie und zu welchem Zweck diese Sprache angewandt wird, wird durch eine Methode festgelegt. Die Methode gibt damit vor, was spezifiziert werden muss und in welcher Reihenfolge die Informationen entstehen [Est08], [RFB12]. Welche Aspekte eines Systems ein Systemmodell abbilden muss, um das System interdisziplinär zu beschreiben, wird im anschließenden Kapitel betrachtet.

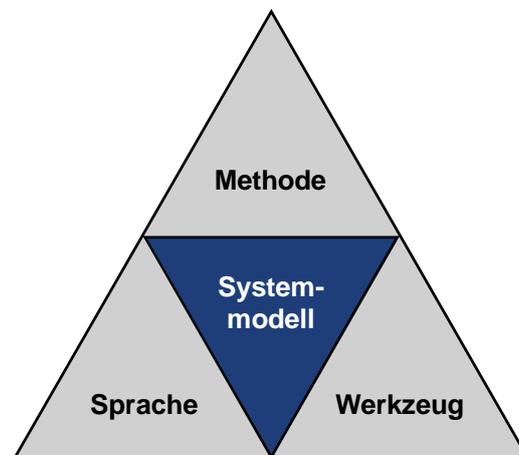


Bild 2-13: Zur Beschreibung eines Systemmodells wird eine geeignete Kombination aus graphischer Modellierungssprache, einem Softwarewerkzeug und einer Methode benötigt

2.5.3 Systemmodell

Mit dem MBSE-Ansatz und der damit verbundenen Systembeschreibung entstehen mehrere Bereiche, in denen das Systemmodell unterstützend eingesetzt werden kann (Bild 2-14). Es ermöglicht eine **ganzheitliche interdisziplinäre Betrachtung des Systems** und fördert damit das Systemdenken. Darüber hinaus wird es als Plattform zur **Kommunikation und Kooperation der beteiligten Fachdisziplinen** gesehen. Auch

eine Koordination der Arbeiten fällt in diesen Bereich. Durch die gemeinsame Erstellung des Systemmodells in der frühen Phase dient das Systemmodell als **Basis für die Konkretisierung** der Arbeiten innerhalb der einzelnen Fachdisziplinen. Die **Dokumentation** der disziplinübergreifenden Informationen erfolgt im Systemmodell. Durch die interdisziplinären Informationen über das System kann es bei der **Verifikation und Validierung** unterstützend eingesetzt werden. Durch die rechnerinterne Beschreibung des Systemmodells kann dieses als **Bindeglied zwischen allen Produktdaten** fungieren.

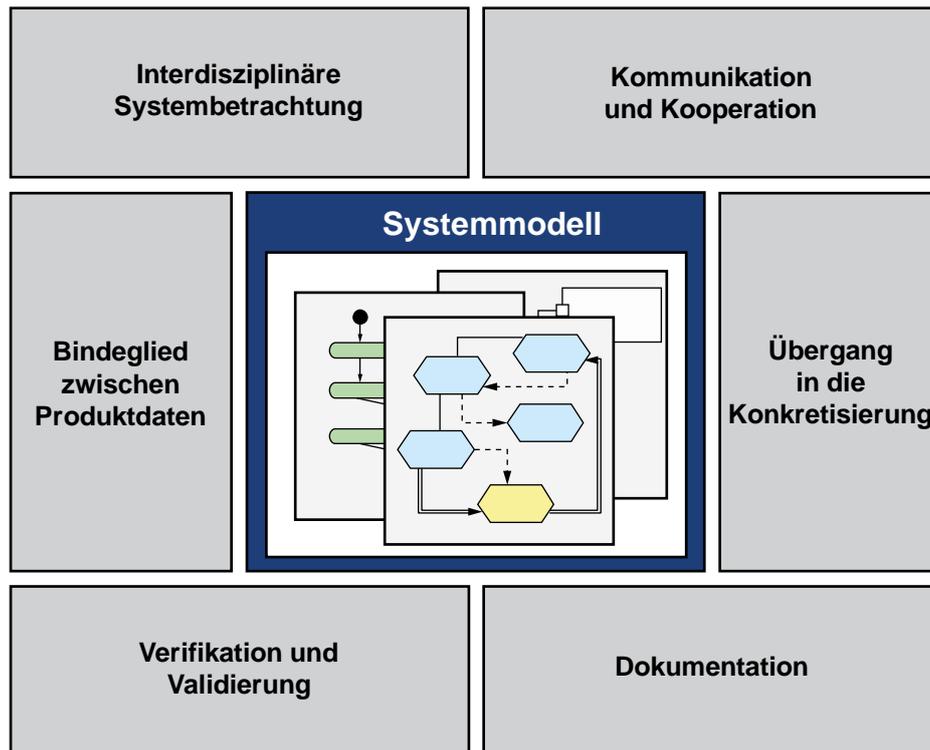


Bild 2-14: Einsatzbereiche des Systemmodells

Interdisziplinäre Systembetrachtung

Mit dem Systemmodell kann eine ganzheitliche Betrachtung des Systems ermöglicht werden. Aus dem technischen Standpunkt heraus ergibt sich die Fragestellung, welche Informationen über das System interdisziplinär sind und zum einheitlichen Verständnis bei den beteiligten Disziplinen führen. Dazu existieren im Wesentlichen drei Aspekte, die in den MBSE-Ansätzen in der Literatur übereinstimmend genannt werden: **Anforderungen, Systemstruktur** und **Verhalten** [Alt12b], [GFD+08], [Kle13].

Anforderungen bilden die Grundlage für jede Entwicklungsaufgabe. Sie definieren die Entwicklungsziele sowie gewünschte Produkteigenschaften aus Kunden- und Unternehmenssicht. Anhand der Anforderungen werden die Teillösungen und auch das fertige Produkt bewertet – sie bilden einen Maßstab für die Entwicklung des Systems [VDI2206]. Anforderungen werden meist in Textform definiert. Dabei wird zwischen Wunsch-/Festforderung bzw. funktionalen/nichtfunktionalen Anforderungen unter-

schieden [PBF+07], [Alt12b]. Die erste Kategorie priorisiert Anforderungen. Wird eine Festforderung nicht erfüllt, kann dies das Aus für die entsprechende Lösungsidee bzw. das Entwicklungsprojekt bedeuten [Lin09]. Wünsche hingegen sollten nach Möglichkeit berücksichtigt werden. Ein Zugeständnis des Mehraufwands wird dabei in Kauf genommen [PBF+07].

Die Unterteilung der Anforderungen in funktional und nichtfunktional hat den Ursprung in der Softwareentwicklung [BBK+09], [SBD+10]. Funktionale Anforderungen beziehen sich auf das Systemverhalten. Sie geben vor, welche Funktionen das System realisieren muss (z.B. die Fensterscheibe muss automatisiert heruntergefahren werden). Alle anderen Anforderungen fallen unter den Bereich nichtfunktional und stellen u.a. Qualitätsanforderungen dar. Nach der Norm ISO/IEC 25010 zählen hierzu Eigenschaften, wie z.B. Zuverlässigkeit, Effizienz oder Benutzbarkeit [ISO25010]. Weitere Beispiele für nichtfunktionale Anforderungen sind Gestaltungsrichtlinien/Normen sowie Vorgaben bezüglich Preis, Herstellkosten, Entwicklungszeit oder Design [Alt12b].

Bezogen auf die Beschreibung von Anforderungen herrscht weitestgehend Einigkeit, während differenzierte Sichtweisen bei der Struktur und dem Verhalten vorhanden sind. Synonym zur **Struktur** werden Begriffe, wie Wirkstruktur [GFD+08], Wirkmodell [Lin09], logische oder funktionale (System-)struktur/Architektur [EGG+12], [Kle13], [Alt12b], [FMS12] verwendet. Grundsätzlich ist damit die Definition der Elemente gemeint, aus denen das System besteht. Systemelemente repräsentieren dabei Softwarekomponenten, Module/Baugruppen oder Bauteile. In der Systemstruktur wird ebenfalls festgelegt, wie die Systemelemente miteinander in Beziehung stehen und damit welche Schnittstellen sie haben. Die Struktur bildet eine statische Betrachtung des Systems, ohne Berücksichtigung ihrer zeitlichen Veränderung, z.B. durch das Bewegungsverhalten oder dem zeitlichen Zusammenhang zwischen ausgetauschten Informationen [Alt12b].

In einigen Fällen – im speziellen aus dem Mechatronik-Bereich – wird die Gestalt ebenfalls als eine Form der Strukturbeschreibung verstanden [Kle13], [GFD+08]. Die Modellierung der Gestalt erfolgt dabei in 3D Werkzeugen, die geometrische Daten beinhalten. Darüber hinaus beschreibt die Gestalt die Auslegung von Bauräumen, Wirkflächen und die Lage der Bauteile. Diese Betrachtungsweise ist bei mechatronischen Systemen sehr wichtig, da die Gestalt, anders als bei softwareintensiven Systemen, einen großen Einfluss auf die Auslegung des Systems und die Wirkungsweise hat.

Das **Verhalten** wird auf vielfältige Weise beschrieben. Funktionen stellen die einfachste Art der Verhaltensbeschreibung dar. Nach PAHL/BEITZ beschreibt eine Funktion

„... den gewollten Zusammenhang zwischen Eingang und Ausgang eines Systems mit dem Ziel, eine Aufgabe zu erfüllen.“ [PBF+07, S.44]

Funktionen können orientiert an den Nutzern (Anwender, Service, Fertigung), an der Umsatzart (Stoff, Energie oder Signal), an der Art der Relation (z.B. eine Funktion ver-

ursacht eine schädliche Funktion) oder hierarchisch betrachtet werden [Lin09]. Dabei finden sich auch Ansätze, die die Funktionsbeschreibung der Struktur zuordnen (vgl. Kapitel 3.6). Neben den Funktionen werden weitere Verhaltensbeschreibungen eingesetzt, die weiterhin abstrahiert das Verhalten des Systems beschreiben, jedoch spezifischer als Funktionen sind. Hierzu zählen Zustände und Zustandsübergänge im System. Aktivitäten zeigen eine Abfolge von Aktionen des Systems. Die Verknüpfung von Aktionen erfolgt über Objekt- oder Kontrollflüsse. Die chronologische Abfolge der Kommunikation zwischen Elementen wird mit Sequenzen betrachtet. Diese stellen jedoch stets eine Abfolge dar, z.B. je Anwendungsfall (Use Case) [GFD+08], [Wei06], [Alt12b].

Kommunikation und Kooperation

Der Erfolg der Entwicklung mechatronischer Systeme hängt von der Kommunikation und Kooperation der beteiligten Disziplinen ab. Die Abstimmung muss weitaus häufiger stattfinden, als es bislang bei technischen Systemen der Fall war [BHL07-ol]. Hierzu ist ein einheitliches Verständnis über das System notwendig. Die Fragestellung ergibt sich daher aus Sicht der beteiligten Disziplinen: Wie ist das Systemmodell abzubilden, damit ein einheitliches Verständnis entsteht und die Kommunikation sowie Kooperation der beteiligten Disziplinen darüber erfolgen kann?

Die Entwickler der einzelnen Disziplinen sind Experten auf ihrem Gebiet. Bei der Abstimmung über Schnittstellen entstehen dabei häufig Missverständnisse. Ein Grund hierfür sind die Unterschiede in den Denk- und Begriffswelten der Disziplinen [VDI2206], [Fra06]. Gleiche Begriffe haben in den verschiedenen Disziplinen unterschiedliche Bedeutung. Beispielsweise wird in der Informatik unter Komponente ein Teil der Software verstanden. In der Konstruktion oder Elektronik entsprechen Komponenten physischen Elementen, wie Bauteilen oder Modulen. Dies wird zum Missverständnis, wenn eine Funktion sowohl durch Software als auch Hardware realisiert werden kann.

Entwickler verwenden fachspezifische Modelle zum Ausdruck ihres Standpunktes. Daraus ergeben sich zwei Gefahrenpotentiale. Dabei setzt der Entwickler voraus, dass der fachfremde Entwickler die Modelle lesen, verstehen und gleich interpretieren kann, wie er selbst. Dies ist in den meisten Fällen nicht gegeben. Ein Konstrukteur wird UML-Modelle der Softwaretechnik nicht verstehen können, ohne sich vorher mit der Sprache auseinander zu setzen. Bei dem Einsatz von disziplinspezifischen Sprach- und Ausdrucksmitteln besteht auch die Gefahr, dass die Entwickler sich schnell in Details verlieren. Sind die Details für die Abstimmung nicht relevant, kann dies von dem fachfremden Entwickler nicht erkannt werden. Zeitintensive und häufige Abstimmungen sind die Folge.

Ein weiterer Punkt, der häufig zu Missverständnissen und Abstimmungsaufwänden führt, ist die Bezeichnung der Systemelemente im Verlauf der Entwicklung. Jede Fachdisziplin bezeichnet Elemente, sofern sie nicht im Vorfeld vorgegeben wurden, auf ihre eigene Weise. Meist erfolgt eine einfache Aufzählung (z.B. Sensor 1, Sensor 2) oder in

der Fachdisziplin üblichen Codierungen (z.B. S3_VL für Sensor 3 vorne links). Diese Kennzeichnung ist notwendig, um die Ausarbeitung des Teilsystems voranzutreiben und innerhalb der Fachdisziplin Einigkeit zu erstellen. In der Softwaretechnik werden diese Begriffe beispielsweise im Quellcode verwendet. Solange die Bezeichnung der Elemente nur eine Disziplin betrifft, entstehen keine weiteren Probleme. Jedoch ist dies nur in seltensten Fällen gegeben. Bei der disziplinübergreifenden Abstimmung wird evtl. nicht sofort erkannt, dass vom selben Element gesprochen wird, weil die Bezeichnungen unterschiedlich vergeben werden (z.B. Sensor 1 und S3_VL für denselben Sensor). Ähnliches ergibt sich bei der Bezeichnung von Parametern und deren Einheiten. Oft treten Probleme auf, weil die Einheit nicht richtig angegeben war (z.B. Inch statt mm).

MBSE gibt den Entwicklern ein Ausdrucksmittel vor, die jede Disziplin gleichermaßen versteht. Die angestrebte Abstraktionsebene ermöglicht, über Sachverhalte zu diskutieren, ohne sich in Details zu verlieren. Mit der gemeinsamen Sprache wird ein Systemmodell erstellt, das ein einheitliches Verständnis des Systems darstellt. Die Bezeichnungen der Elemente, der Parameter und der Einheiten werden disziplinübergreifend festgelegt und im Systemmodell dokumentiert.

Das Modell wird in der frühen Phase der Entwicklung erstellt und bildet die Basis für die Konkretisierung in den entsprechenden Disziplinen. Während des gesamten Entwicklungsprozesses erfolgt die Abstimmung zwischen den Disziplinen über das Systemmodell. Dieses ist damit über die Konkretisierung zu pflegen und bei Änderungen anzupassen. Da im Systemmodell die einzelnen Modellinhalte mit den Anforderungen verknüpft sind, können Auswirkungen von vorzunehmenden Änderungen rückverfolgt und bewertet werden [JLS11].

Übergang in die Konkretisierung

Das Systemmodell wird in Kooperation aller beteiligten Disziplinen erstellt und beinhaltet bereits viele Informationen über das System. Auf Basis des Systemmodells beginnt die Konkretisierung der Teillösungen bzw. die Verifikation von Subsystemen. Eine Verifikation kann zum Beispiel die Simulation von Bewegungsverhalten in Modelica sein. Dies kann in der frühen Phase auf ersten abstrakten Informationen erfolgen. Beim Übergang vom Systemmodell in die Ebenen der formalen (multidisziplinären und disziplinspezifischen) Modelle sind bereits dokumentierte Informationen des Systemmodells zu nutzen. Dies kann einerseits manuell durchgeführt werden. Hierzu greift jede Fachdisziplin die spezifischen Informationen heraus. Der Abgleich der Simulationsmodelle, bzw. der disziplinspezifischen Modelle mit dem Systemmodell erfolgt in diesem Fall ebenfalls manuell. Einen automatischen Weg für das Ableiten der Informationen und Sicherung der Konsistenz wurde in [GSG+09] für die Fachdisziplin Softwaretechnik dargestellt.

Dokumentation

MBSE verfolgt den Ansatz, von der dokumentenzentrierten hin zur modellbasierten Arbeitsweise zu gelangen [INC07-ol]. Ein weiterer Punkt dabei ist die Dokumentation disziplinübergreifender Abstimmungen und Entscheidungen. Derzeit entstehen viele Modelle der zweiten und dritten Ebene (Bild 2-11). Sind Abstimmungen notwendig, wird dieses in den Modellen eingepflegt, jedoch der Grund für die Änderung nicht protokolliert. Im Verlauf der Entwicklung bzw. auch bei ähnlichen Entwicklungsaufgaben kann diese Entscheidung nicht mehr nachvollzogen werden. Wurde eine Abstimmung über einen direkten Zusammenhang getroffen, besteht die Gefahr, dass indirekte Zusammenhänge nicht erkannt und damit Dritte über die Änderung nicht informiert wurden. Je nach Schweregrad der Auswirkung führt dies wiederum zu zeit- und kostenintensiven Iterationen.

Mit dem MBSE Ansatz werden diese Änderungen in dem Systemmodell gespeichert, so dass dieses die notwendigen disziplinübergreifenden Informationen enthält. Die indirekten Zusammenhänge werden aus dem Systemmodell ersichtlich, bzw. werden die indirekt betroffenen Disziplinen über die Änderung informiert und können darauf frühzeitig reagieren.

Verifikation und Validierung

Das Systemmodell soll bei der Verifikation und Validierung unterstützen. Unter Berücksichtigung der im Systemmodell enthaltenen Informationen kann eine Verifikation stattfinden. Die Anforderungen sind mit der Struktur und dem Verhalten verlinkt. Qualitative Methoden, wie FMEA können auf Basis des Systemmodells durchgeführt werden [GKP09], [Alt12b]. Damit werden jedoch nicht alle Anforderungen abgedeckt. Die Anforderungen auf Subsystemebene, die teilweise ins Detail gehen (Design und Haptik von Elementen oder Strömungsverhalten), können erst durch ergänzende Modelle (CAD- oder Simulationsmodelle) überprüft werden. Für die Validierung wird das Systemmodell genutzt, um Testfälle abzuleiten und zu dokumentieren. Die Validierung selbst erfolgt jedoch durch spezifische Tests, wie z.B. virtuelle Inbetriebnahme, X-in-the-Loop. In der Literatur existieren ebenfalls Ansätze, die ein ausführbares Systemmodell adressieren. Damit soll über die Verknüpfung der Anforderungen mit der Analyse des Modells eine Aussage über die Verletzung der Anforderung getroffen werden können [JLS11].

Bindeglied zwischen Produktdaten

Das Systemmodell stellt eine abstrahierte Beschreibung über das zu entwickelnde System dar. Es enthält viele Informationen, die in den Fachdisziplinen weiterverwendet und auch innerhalb der Disziplin geändert werden. Die Konsistenz der Daten muss dabei sowohl horizontal (über alle Aspekte im Systemmodell) als auch vertikal (zwischen dem Systemmodell und spezifischen Modellen) sichergestellt sein. Je mehr Modelle diese Daten jedoch benötigen und damit Änderungen entstehen können, umso auf-

wendiger ist es, diese konsistent zu halten. Im Idealfall werden Änderungen, ausgehend vom Systemmodell, vorgenommen und zu den einzelnen Fachdisziplinen propagiert. Dies kann auf unterschiedliche Weise geschehen. Zum einen können hierzu aspektübergreifende Sichten gebildet werden, wenn beispielsweise eine Dokumentation der Systemspezifikation entstehen soll. Darüber hinaus gibt es Ansätze, die das Systemmodell als Bindeglied zwischen allen anderen Modellen der Produktentwicklung sehen (Bild 2-15). Die Umsetzung kann als direkte Toolkopplung oder durch einfache Verlinkung von Modellelementen des Systemmodells mit den Artefakten⁷ anderer Modelle realisiert werden [FMS12], [GSG+09], [RPC+12], [Sen13].

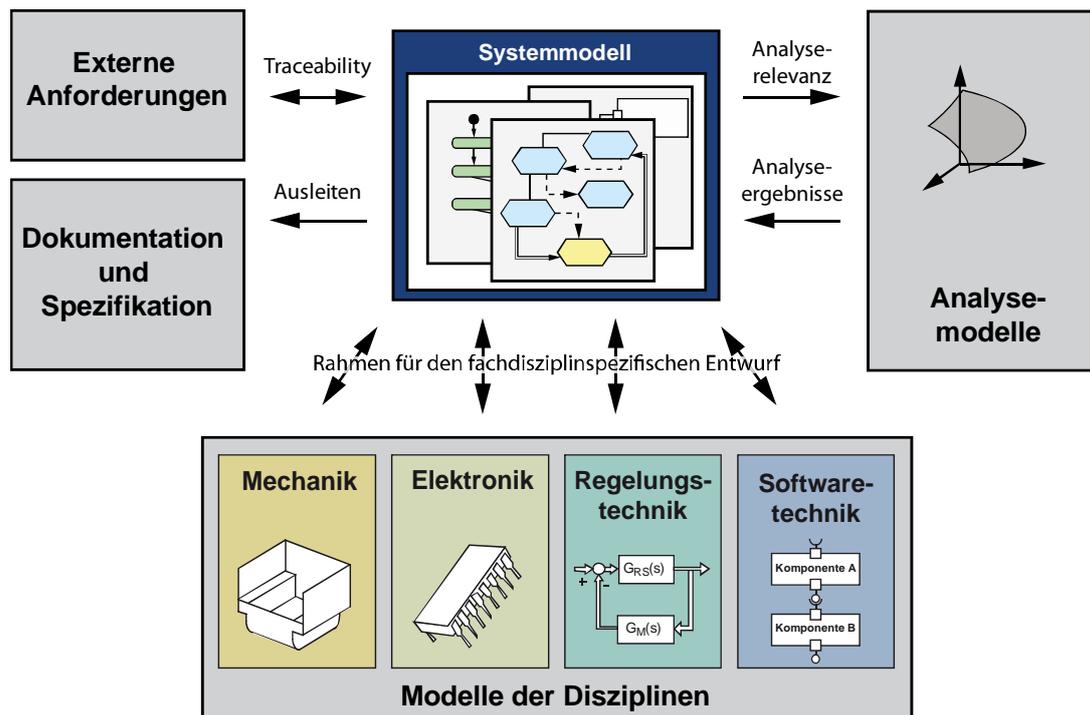


Bild 2-15: Systemmodell bildet einen Rahmen für Analysen und Rückverfolgung (nach [FMS12])

Weitere Ansätze verstehen das Systemmodell als Bestandteil eines PLM⁸-Konzeptes [Sen13]. PLM ist eine Strategie, die aus der Ausweitung der PDM⁹-Ansätze auf den gesamten Produktlebenszyklus entstanden ist. Ziel ist die Integration und Verwaltung der im Produktlebenszyklus anfallenden Daten (auch Produktmodell genannt). Hierzu werden Prozesse definiert, wie z.B. Prüf-, Freigabe- und Änderungsprozesse [Abr05], [ES05], [ES09]. Da das Systemmodell ebenfalls Informationen über das Produkt ent-

⁷ Unter Artefakt wird ein Zwischen- oder Endergebnis einer Entwicklungstätigkeit gesehen. Beispiele für Artefakte sind eine technische Zeichnung, ein Schaltplan, Simulationsmodell oder der Quellcode einer Software.

⁸ PLM – Product Lifecycle Management

⁹ PDM – Product Data Management

hält, ist die Integration in PLM-Lösungen ein notwendiger Schritt. Neben der Verwaltung der im Systemmodell anfallenden Daten bietet die Integration die Möglichkeit, über das Systemmodell alle anderen Daten zu verknüpfen (Bild 2-16).

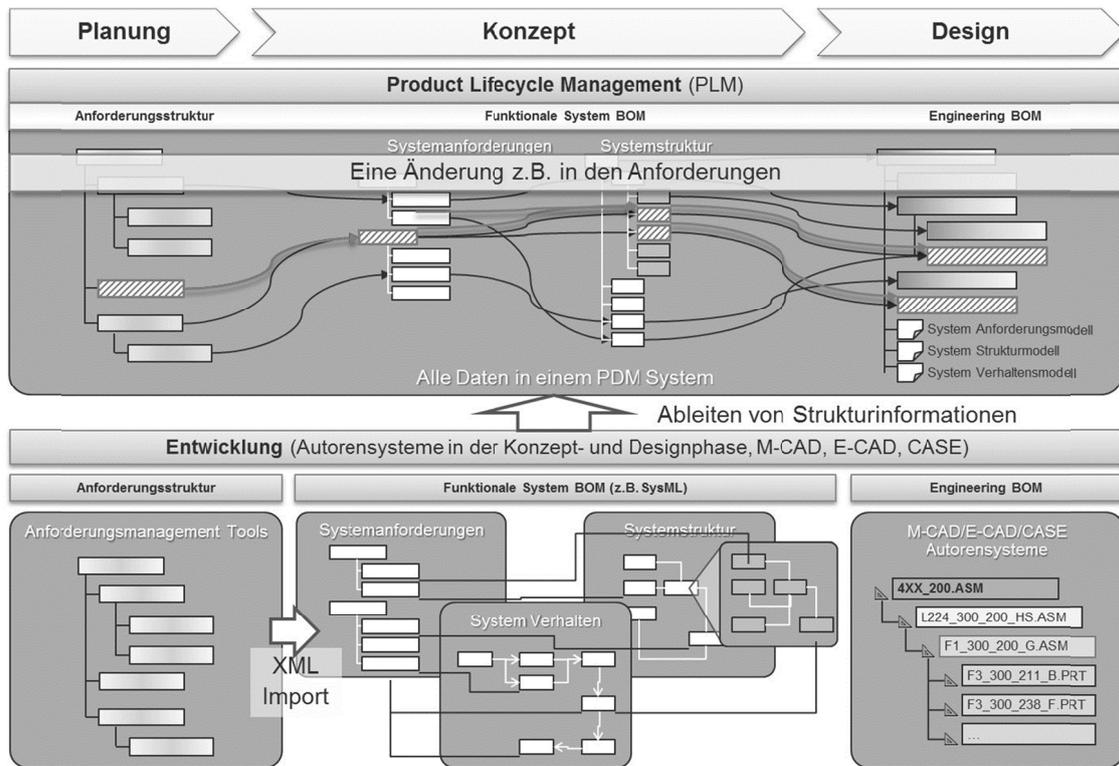


Bild 2-16: Systemmodell als Teil des PLM-Konzepts [Sen13]

2.5.4 Partizipation der SE-Rollen am Systemmodell

In Kapitel 2.4.3 wurden Rollen des SE eingeführt. Nach der Erläuterung des Systemmodells wird aufgezeigt, wie die Rollen am Systemmodell partizipieren. Dabei generieren die Rollen Inhalte und benötigen zur Erfüllung ihrer Aufgaben wiederum Inhalte aus dem Systemmodell.

Die Informationen für und vom Requirements Owner (RO) sind auf die Anforderungen festgelegt. Diese werden von ihm beschrieben und verwaltet. Die Verknüpfung der Anforderungen mit den anderen Modellinhalten benötigt der RO für seine Arbeit zur Rückverfolgung von Änderungen. Die Rolle der Kundenschnittstelle (CI) liefert die Informationen über die Kundenwünsche und verfolgt die technische Umsetzung. Hierzu greift er Informationen aus der Systemstruktur und dem -verhalten heraus.

Die Systemstruktur enthält die Subsysteme mit den Schnittstellen. Diese Informationen kommen vom System Designer (SD). Die Glue Rolle (G) nutzt die Schnittstellenbeschreibung, um seine Aufgabe als Integrator zu erfüllen. Falls Schnittstellen sich ändern oder Änderungen auf Subsystemebene entstehen, kann diese Rolle am Systemmodell

die Auswirkungen erkennen. Der Systemanalyst (SA) benötigt die Systemstruktur und das Verhalten sowie die entsprechende Verknüpfung zu den Anforderungen.

Bei der Erfüllung der Arbeit der Rollen Verification/Validation (VV) und Logistics and Operations (LO) stellt das Systemmodell wichtige Informationen bereit. Aus den Anforderungen und funktionaler Beschreibung des Systems können Testfälle abgeleitet werden. Darüber hinaus können auf Basis von Verhaltensbeschreibungen und Struktur die Lernmaterialien und Betriebsbedingungen erstellt werden.

Die Rollen aus dem Aufgabenbereich Organisation und Koordination nutzen das Systemmodell als Kommunikationsmittel. Zum Beispiel kann auf Basis der Strukturbeschreibung die Meinungsverschiedenheiten der Disziplinen diskutiert werden. Der Koordinator (CO) nutzt das graphische Modell und kann gezielt mit den Verantwortlichen über die Inhalte diskutieren. Ziel ist ein Konsens bei den beteiligten Disziplinen zu erreichen.

Neben den von SHEARD eingeführten Rollen gibt es weitere Stakeholder des Systemmodells. Die abstrahierte Darstellung des gesamten Systems kann als Kommunikationsmittel für die nicht-technischen Bereiche, wie beispielsweise Management, Vertrieb oder Einkauf genutzt werden. Am Systemmodell können Sachverhalte erläutert und Informationen ausgetauscht werden. Zusätzlich kann das Systemmodell für die Abstimmung zwischen Unternehmen dienen, die die Entwicklung von Subsystemen an externe Partner vergeben. So wird eine einheitliche Beschreibung der Schnittstellen erreicht und die Rückverfolgung von Anforderungen über Unternehmensgrenzen bereitgestellt.

2.5.5 Modellierungssprache

Das Systemmodell beinhaltet die Informationen aller beteiligten Disziplinen in einer abstrakten Form mit dem Ziel, dass jede Disziplin das Modell gleichermaßen lesen und verstehen kann. Darüber hinaus müssen die Inhalte formalisiert vorhanden sein, damit sie rechnerintern verarbeitet werden können. Zur Modellierung des Systemmodells wird eine Sprache benötigt, die alle Disziplinen gemeinsam nutzen [OMG12], [Wei06]. Eine Modellierungssprache wird durch die Syntax und Semantik definiert. Die Syntax unterteilt sich in abstrakte und konkrete Syntax.

*„Die **abstrakte Syntax** (Grammatik) definiert die syntaktischen Elemente oder Konstrukte (z.B. Buchstaben) und regelt, wie Konstrukte (z.B. Wörter) aus anderen gebildet werden, während die **konkrete Syntax** die Ausdrucksmittel (Darstellungsform, Notation) festlegt...“ [PM06].*

Die graphische Notation entspricht der konkreten Syntax und wird anschließend näher betrachtet.

Mit der Semantik wird festgelegt, wie Modellkonstrukte miteinander verknüpft werden müssen, um eine Bedeutung zu haben (statische Semantik). Dies wird in Form von Bedingungen gegen die abstrakte Syntax definiert [SV07]. Welche Bedeutung die Modellkonstrukte und damit auch welche Bedeutung die Verknüpfung von Modellkonstrukten haben, beschreibt die dynamische Semantik.

Die abstrakte Syntax und die statische Semantik werden in einem sogenannten Metamodell definiert [SV07]. Die konkrete Syntax wird separat festgelegt und den Modellkonstrukten, beschrieben in der abstrakten Syntax, zugeordnet. Auch die dynamische Semantik wird nicht im Metamodell festgehalten. Sie kann in Textform erfolgen oder streng mathematisch durch eine Grammatik. Die Spezifikation der Modellierungssprache beinhaltet das Metamodell (abstrakte Syntax, statische Semantik), die konkrete Syntax (graphische Notation) sowie die Beschreibung der dynamischen Semantik.

Die Unterscheidung zwischen formalen, semiformalen und informalen Sprachen bezieht sich auf den Formalisierungsgrad der Syntax und der Semantik. Erst wenn beide, jedoch im speziellen die dynamische Semantik, formal durch eine Grammatik beschrieben werden, handelt es sich um eine formale Sprache. Dies ist selbst in der Informatik nur in Sonderfällen gegeben [BBK+09].

Zur interdisziplinären Beschreibung des Systemmodells wird eine semiformale Modellierungssprache bevorzugt. Formale Modelle hemmen die Kreativität der Entwickler, erzeugen Akzeptanzhürden bei den Beteiligten und Barrieren der freien und flexiblen Modellierung. Diese ist notwendig, um die verschiedenen Aspekte des Systems aus den Blickwinkeln der Entwickler darstellen zu können.

Graphische Notation

Zur Darstellung des Systemmodells werden überwiegend graphische Modellierungssprachen verwendet. Die Vorteile der graphischen Modellierung liegen in der Effektivität und Effizienz in Bezug auf Bearbeitung, Wahrnehmung und Pflege der Modelle durch den Benutzer [SFP+09]. Diese Vorteile werden jedoch nicht vollständig genutzt, da bei der Sprachdefinition die Wichtigkeit der Notation unterschätzt wird [JWK+08], [Moo09]. Während die abstrakte Syntax und die Semantik sorgfältig definiert und durchdacht sind, wird die visuelle Repräsentation als trivial angesehen. Die graphische Notation erhält dadurch fälschlicherweise einen ästhetischen Charakter, so dass die Auswahl der Ausdrucksmittel meist auf dem persönlichen Geschmack basiert und weniger auf wissenschaftlich fundierten Erkenntnissen [Moo07]. Im Bereich des Software Engineerings wurde in Studien bestätigt, dass die visuelle Darstellung zum Verständnis des Modells wesentlich beiträgt [Moo09].

Mit den Symbolen der graphischen Notation wird beabsichtigt, eine Nachricht zu übermitteln. Hierzu wird die Nachricht in Form des verwendeten Symbols codiert. Der Mo-

der Nutzer muss diese erst wieder decodieren, um die gesendete Nachricht zu erhalten. Dies ist in Bild 2-17 veranschaulicht.

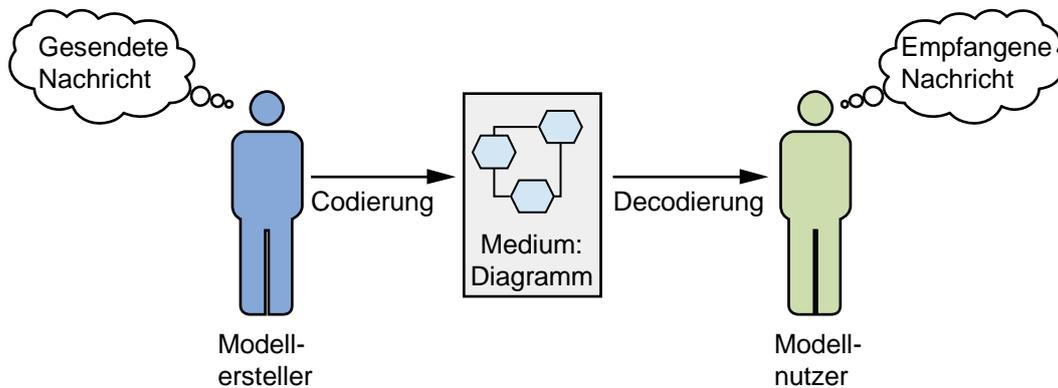


Bild 2-17: Theorie der Kommunikation unter Einsatz graphischer Notation nach [Moo09]

Die Nachricht kann jedoch nur eindeutig übermittelt werden, wenn die Symbolauswahl eindeutig gewählt wird. Folgende Fehler treten häufig bei der Definition der graphischen Notation auf, so dass die Übermittlung der Nachricht gestört ist (Bild 2-18).

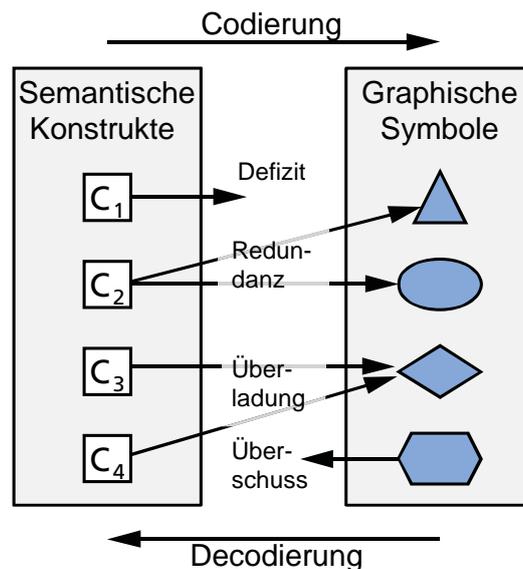


Bild 2-18: Potentielle Fehler bei der Zuordnung von Symbolen zu semantischen Konstrukten [Moo09]

- **Redundanz der Symbole:** Mehrere graphische Symbole werden zur Präsentation desselben semantischen Konstrukts verwendet (Dreieck und Oval für C2).
- **Überladung der Symbole:** Gleiches Symbol wird für zwei unterschiedliche semantische Konstrukte verwendet (Raute für C3 und C4).
- **Symbol-Überschuss:** Ein Symbol hat kein semantisches Konstrukt, das es präsentiert (Sechseck).

- **Symbol-Defizit:** Ein semantisches Konstrukt hat kein Symbol zur Präsentation (C1).

Auswahl der Modellierungssprache

Die Entscheidung für oder gegen eine Modellierungssprache wird unter anderem von der Usability der Sprache beeinflusst. REBSTOCK et al. bewertet die Usability mit den Attributen Erlernbarkeit, Einprägsamkeit, Effektivität, Effizienz, visuelle Wahrnehmbarkeit sowie Benutzerzufriedenheit [SRC10]. Der Benutzer muss die Sprache schnell erlernen und nach längerer Zeit ohne Spracheinsatz sich wieder an die verschiedenen Elemente und Regelungen erinnern können. Effizient ist eine Modellierungssprache dann, wenn der Benutzer unterstützt wird, das Modell fehlerfrei und schnell zu erstellen. Der Gesamteindruck des Modells, der definierten Elemente (Formen, Farben) fallen unter die visuelle Wahrnehmbarkeit. Alles zusammen wirkt sich auf die Benutzerzufriedenheit aus. Diese wird an individueller Wahrnehmung beim Modellieren und Interpretieren gemessen [SRC10], [HN09].

2.5.6 Komplexität im Systemmodell

Mit dem Systemmodell soll die Komplexität der Systeme beherrscht werden. Durch abstrahierte modellhafte Abbildungen werden Zusammenhänge veranschaulicht und der Fokus auf einen Aspekt des Systems gelegt (z.B. Struktur). Je Aspekt entsteht ein Teilmodell (Partialmodell) im Systemmodell. Diese unterschiedlichen Aspekte beschreiben dabei stets dasselbe System und enthalten zum Teil redundante Informationen. Dies wird vor allem bei dem Aspekt Anforderungen deutlich. Anforderungen beschreiben Bedingungen an das zu entwickelnde System. Die Umsetzung wird in den anderen Aspekten spezifiziert. Beispielsweise beschreibt eine funktionale Anforderung ein gewünschtes Verhalten des Systems: „Das System muss um eine Kurve fahren können“. In der Realisierung werden hierzu ein Lenker und bewegbare Achsen eingesetzt. Diese finden sich in der Strukturbeschreibung in Form von Systemelementen wieder. Das zugehörige Verhalten wird wiederum im Aspekt Verhalten, z.B. durch eine Aktivitätsfolge dargestellt. Die einzelnen Modellelemente müssen dabei auch aspektübergreifend in Beziehung gesetzt werden. Daraus können Änderungen nachvollzogen und rückverfolgt werden. Darüber hinaus bietet es die Möglichkeit, die Konsistenz im Systemmodell sicherzustellen.

Unter Berücksichtigung der Definition für Komplexität, wird die Systemkomplexität zwar durch die Abstraktion reduziert, jedoch eine Modellkomplexität durch die verschiedenen Aspekte generiert [LSO06]. Es entstehen viele Modellelemente, die durch viele Beziehungen miteinander verbunden sind. Da der Entwicklungsprozess an sich dynamisch ist, ergeben sich Änderungen der Elemente und Beziehungen. Somit ändern sich diese über die Zeit, was aus einem komplizierten Modell ein komplexes Modell macht.

Die Handhabung der Modellinhalte (sie entsprechen den Daten im Repository) kann durch ein Softwarewerkzeug unterstützt werden. Dies kann nur zu einem gewissen Grad geschehen, da das Tool die Bedeutung hinter den Modellelementen (Lenker) nur interpretieren kann, wenn diese sehr formal beschrieben werden.

Ein weiterer Punkt beeinflusst die Modellkomplexität: die graphische Darstellung. Die graphischen Modelle werden aus Gründen der Übersichtlichkeit und Anschaulichkeit genutzt. Werden jedoch viele Inhalte auf einmal dargestellt, sinkt die Anschaulichkeit mit der Anzahl an Elementen und Beziehungen.

2.6 Grundsätze ordnungsmäßiger Modellierung

Im Bereich des Prozessmanagements werden ebenfalls Modelle zur Beschreibung der Prozesse eingesetzt. Dieses Themenfeld befasst sich mit analogen Fragenstellungen, wie diese im MBSE derzeit diskutiert werden. Zum Beispiel drohen die Modelle bei komplexen Projekten unübersichtlich zu werden oder redundante Informationen werden in Teilmodellen erstellt, die wiederum konsistent gehalten werden müssen. Diese Fragestellungen hat BECKER aufgegriffen und sechs Grundsätze ordnungsmäßiger Modellierung formuliert [BPV12].

Mit den Grundsätzen soll eine Vergleichbarkeit von Modellen und Bewertbarkeit hinsichtlich Qualität ermöglicht werden. Die Notwendigkeit ergibt sich aus der Tatsache, dass bei der Erstellung eines Modells für einen gegebenen Sachverhalt selten identische Modelle entstehen, selbst wenn die gleiche Modellierungsmethodik eingesetzt wurde. Durch die Modellierungssprache sind erste Regeln vorgegeben, die bei der Modellierung zu beachten sind. Jedoch reichen diese meist nicht aus, um zu vergleichbaren und zweckmäßigen Modellen zu gelangen [BPV12].

Die sechs Grundsätze sind aus der engen Anlehnung an die Grundsätze ordnungsmäßiger Buchführung definiert worden: Grundsatz der Richtigkeit, Grundsatz der Relevanz, Grundsatz der Wirtschaftlichkeit, Grundsatz der Klarheit, Grundsatz der Vergleichbarkeit und der Grundsatz des systematischen Aufbaus [BPV12].

Grundsatz der Richtigkeit: Bei der Richtigkeit wird zwischen syntaktischer und semantischer Richtigkeit unterschieden. Ein Modell ist hinsichtlich der Syntax dann richtig, wenn es konsistent zum Metamodell erstellt wurde und damit formal korrekt ist. Damit ist noch nicht sichergestellt, dass die dargestellten Inhalte dem entsprechen, was der Modellierer ausdrücken wollte und was für weitere Modellnutzer von Bedeutung ist. Dies festzulegen und zu überprüfen ist nicht möglich, da ein Modell zweckgebunden hinsichtlich Güte der Informationen und Qualität des Modells bewertet wird. Damit müssen die hierfür notwendigen Maßnahmen zweckorientiert formuliert werden. Hierzu kann zum Beispiel die Definition und Nutzung von Namenskonventionen gezählt werden.

Grundsatz der Relevanz: Der Grundsatz der Relevanz stellt sicher, dass bei der Modellierung nur die Sachverhalte dargestellt werden, die für den Modellierungszweck wesentlich sind. Damit geht der Abstraktionsgrad des Modells einher. Je nach Zweck des Modells, muss dieses Sachverhalte auf verschiedenen Abstraktionsebenen darstellen können. Dient das Informationsmodell bspw. als Basis für die Umsetzung eines Workflowmanagementsystems, muss es detailliert beschrieben werden. Anders ist es, wenn es zum Organisations- oder Kommunikationszweck genutzt wird. Damit setzt der Grundsatz der Relevanz voraus, dass zu Beginn der Modellierung das Ziel und damit Zweck der Modellierung definiert wird.

Grundsatz der Wirtschaftlichkeit: Die Modellierung ist unter wirtschaftlichen Gesichtspunkten zu betrachten, da diese mit Aufwand verbunden ist. Zielsetzung ist daher, das Modellierungsziel mit minimalem Aufwand zu erreichen. Dies kann auf verschiedene Arten geschehen. Ein Beispiel ist die Definition eines Modellierungsziels sowie Kriterien zur Überprüfung dessen Einhaltung. Einen weiteren Einfluss auf die Wirtschaftlichkeit hat die Auswahl der Modellierungssprache. Je nachdem wie die Sprache definiert ist und welchen Zweck sie verfolgt, kann dies Auswirkung auf den benötigten Aufwand haben. Zur Reduktion des Modellierungsaufwands wird unter dem Grundsatz der Wirtschaftlichkeit empfohlen, Referenzmodelle zu nutzen. Diese haben eine gewisse Allgemeingültigkeit für bestimmte Sektoren, Branchen oder betriebliche Funktionen. Der Modellierer beginnt damit nicht bei null und kann auf dokumentierte Best Practices zurückgreifen. Der Aufwand liegt dann in der unternehmensspezifischen Anpassung.

Grundsatz der Klarheit: Der Grundsatz der Klarheit adressiert die Leserlichkeit, Verständlichkeit und bestmögliche Anschaulichkeit von Modellen. Hierzu gehört unter anderem die Hierarchisierung, Layoutgestaltung und Filterung. Hierarchisierung soll zur Beherrschung der Komplexität und aus Übersichtsgründen genutzt werden. Prozesse, die aufeinander aufbauen, sind zu hierarchisieren. Damit kann eine einfache Navigation durch die Modelle erreicht werden. Die Layoutgestaltung bezieht sich auf die Anordnung der Modellelemente. Dabei sollen Konventionen eingehalten werden, die der Übersichtlichkeit dienen. Dazu zählt beispielsweise, dass Überschneidungen von Kanten vermieden werden sollen. Dies hängt jedoch stark von der Größe des Modells sowie dem Medium ab, mit dem es betrachtet wird (z.B. Bildschirm, Papier). Da nicht jede Information des Modells für jeden Modellbetrachter relevant ist, sind Filterfunktionen notwendig mit denen Sichten auf das Modell ermöglicht werden. Nicht relevante Elemente werden in dieser Sicht ausgeblendet.

Grundsatz der Vergleichbarkeit: Bei der Vergleichbarkeit werden zwei Anwendungsfälle unterschieden. Werden gleiche Sachverhalte (Abläufe) mit derselben Modellierungssprache und Methode erstellt, müssen diese weitestgehend identisch und damit vergleichbar sein. Der zweite Fall bezieht sich auf Modelle, die einen identischen Sachverhalt mit unterschiedlichen Modellierungssprachen dokumentieren. Dieser Grundsatz zielt darauf ab, die Modelle so zu erstellen, dass sie in einander überführt werden können.

Grundsatz des systematischen Aufbaus: Mit dem Grundsatz des systematischen Aufbaus wird ein konsistentes Gesamtmodell fokussiert. In den meisten Fällen wird ein Sachverhalt aus unterschiedlichen Gesichtspunkten beschrieben (z.B. Prozesssicht, Datensicht, Organisationssicht). Damit das Gesamtsystem konsistent ist, müssen die referenzierten Modellelemente der einen Darstellung dem identischen Modellelement der anderen Darstellung entsprechen.

Durch die allgemeingültige Formulierung der Grundsätze, sind diese für jede Art von Modellierung anwendbar und können auf die Systemmodellierung übertragen werden. Zur praktischen Umsetzung der Grundsätze sind operativ umsetzbare Handlungsempfehlungen in Form von Richtlinien zu definieren.

2.7 Problemabgrenzung

Die Entwicklung mechatronischer Systeme bedarf einer verzahnten Zusammenarbeit vieler unterschiedlicher Fachdisziplinen. Die Vorgehensmethoden der Disziplinen reichen nicht aus, die Komplexität der zu entwickelnden Systeme zu beherrschen und die interdisziplinäre Zusammenarbeit zu fördern. Dies kann durch ein gemeinsames Systemdenken und einheitliches Systemverständnis erreicht werden. Systems Engineering verfolgt dieses Ziel durch ganzheitliche und interdisziplinäre Denk- und Arbeitsweisen. Das Systemmodell ist ein Hilfsmittel, mit dem das System als Ganzes betrachtet und so das Systemdenken gefördert wird. Model-Based Systems Engineering befasst sich mit den Modellierungssprachen, Methoden und Werkzeugen zur Beschreibung des Systemmodells.

Das Systemmodell enthält unterschiedliche Aspekte des Systems. In den Methoden wird dabei festgelegt, was zu modellieren ist und wie diese Informationen entstehen. Eine Analyse der Methoden wird im Stand der Technik vorgenommen (vgl. Kap. 3.2). Bei allen Methoden wird die Systemstruktur als zentraler Aspekt gesehen. Es beschreibt das System in seiner Zusammensetzung, seiner Wirkungsweise sowie seiner Einbettung in dem Umfeld. Die Betrachtung der Systemstruktur weist wesentliche **Nutzenpotentiale** auf.

- **Ganzheitliches Systemverständnis:** Die einzelnen Fachdisziplinen haben ihre spezifischen Modelle und Dokumente zur Beschreibung ihrer Sicht auf das System. Die ganzheitliche Betrachtung des Systems entsteht erst in der Systemstruktur. Das System wird dabei durch seine Bestandteile (Elemente und Beziehungen) interdisziplinär beschrieben. Durch den gemeinsamen Aufbau der Systemstruktur wird ein einheitliches und ganzheitliches Verständnis erzeugt.
- **Interdisziplinäre Dokumentation und Traceability:** Das Systemmodell und damit auch die Systemstruktur bilden die Basis der interdisziplinären Zusammenarbeit. Bei Änderungen werden diese im Systemmodell dokumentiert. Durch die Verknüpfung der Elemente innerhalb der Systemstruktur und darüber hinaus mit

anderen Aspekten im Systemmodell, ist eine Rückverfolgbarkeit vorhanden. Bei Änderungen kann nachverfolgt und nachvollzogen werden, wie sich diese im System auswirken.

- **Transparenz durch Visualisierung:** Die graphische Darstellung der Elemente und ihrer Wirkbeziehung schafft Transparenz. In den disziplinspezifischen Modellen werden jeweils nur bestimmte Subsysteme oder ausgewählte Ausschnitte des Systems dargestellt. Diese können zudem nicht immer von Fachfremden nachvollzogen werden. Die Systemstruktur beschreibt interdisziplinär auf einfache Weise das Zusammenspiel der einzelnen Elemente. Damit enthält sie auch die Beschreibung der Schnittstellen, die sich aus der Interaktion ergeben. Mit der Darstellung wird den Entwicklern eine Basis geschaffen, die Elemente und Wirkbeziehungen einheitlich zu benennen und über Schnittstellen an einem graphischen Modell zu diskutieren. Die graphische Modellierung bringt einen weiteren Vorteil mit sich: Durch die formalisierte Beschreibung können Inhalte dargestellt und kommuniziert werden, die sonst in Textform aufwendig und nicht einheitlich gestaltet wurden.
- **Frühzeitige Fehlervermeidung:** Die Darstellung des Systems als Ganzes trägt zum gemeinsamen Verständnis bei. Durch die interdisziplinäre Arbeitsweise an der Systemstruktur werden darüber hinaus auch Fehler erkannt, die sonst erst in der Integration festgestellt werden würden. Diese Fehler entstehen z.B. an den Schnittstellen der Systemelemente. Durch softwareunterstützte Modellierung können geeignete Prüfmechanismen einen Beitrag zur Fehlervermeidung leisten.

Die Einführung der Systemmodellierung in einem Unternehmen ist verbunden mit einem Paradigmenwechsel: Die Arbeits- und Denkweise der Entwicklung wird geändert. Das Systemmodell soll von Beginn an erstellt, über die Projektlaufzeit gepflegt und damit Änderungen dokumentiert werden. Dies ist mit Aufwand verbunden, den die Entwickler als hinderlich betrachten. Zudem fördert das Modell zwar das Systemverständnis, es löst jedoch nicht die Entwicklungsaufgabe des einzelnen Fachexperten. Der Entwicklungsaufwand bleibt damit für den einzelnen Fachexperten weiterhin bestehen auch wenn der Abstimmungsaufwand in der Ausgestaltung reduziert wird. Damit sind **Herausforderungen** zu bewältigen, die sich bei einer Einführung der Systemmodellierung ergeben.

- **Aufwand-Nutzen-Verhältnis:** Die Einsatzmöglichkeiten der Systemmodellierung sind vielfältig. Neben der Schaffung von Transparenz durch Visualisierung kann das Modell genutzt werden, weitere Entwicklungsschritte zu unterstützen. Damit kann und sollte das Modell für weitere Zwecke genutzt werden. Unter anderem z.B. für das Qualitätsmanagement zur Durchführung einer FMEA [GKP09], [Alt12b]. Je nach Zweck werden unterschiedliche Inhalte aus dem Systemmodell benötigt. Der Aufwand der Modellierung muss dem Modellierungsziel angemessen sein. Dies ist wiederum verbunden mit der Kenntnis der Einsatzmöglichkeiten sowie der zur Verfügung stehenden Modellierungssprachen, Methoden und Werkzeuge.

- **Formalisierungsgrad des Modells:** Das Systemmodell kann unterschiedlich formal beschrieben werden. Dies hängt direkt mit dem Modellierungsziel zusammen. Für eine automatische Auswertung des Modells ist eine hohe Formalisierung notwendig. Damit eine Software die vorhandenen Informationen eindeutig bearbeiten und auswerten kann, müssen diese formal rechnerintern vorliegen. Mit dem Grad der Formalisierung steigt der Aufwand zum Erlernen und Anwenden der Modellierungssprache und -werkzeuge.

Für den interdisziplinären Einsatz der Systemmodellierung sollte gerade dieser Aufwand möglichst gering gehalten werden. An der Erstellung des Modells sind viele Wissensträger beteiligt, die unterschiedliche Voraussetzungen mitbringen [RFB12]. Hierzu zählt unter anderem der Bildungsstand, der Fachhintergrund sowie die Denk- und Arbeitsweise. Unter den Modellnutzern sind neben den Fachexperten noch weitere Unternehmensbereiche vertreten, wie bspw. der Einkauf, der Vertrieb oder die Fertigung. Der Formalisierungsgrad ist daher klein zu halten.

- **Plausibilität:** Ziel der Modellierung sind Systemmodelle, deren Modellinhalte sowohl vergleichbar als auch vollständig und richtig spezifiziert sind. Dies ist durch die semiformale Beschreibung des Systems häufig nicht gegeben. Modelle, die von verschiedenen Personen erstellt werden, unterscheiden sich in den Modellinhalten, in der Darstellung und in der Abstraktionstiefe. Werden darüber hinaus unterschiedliche Modellierungssprachen verwendet, werden gleiche Sachverhalte auf unterschiedliche Weise dargestellt. Die Grundsätze Vergleichbarkeit, Relevanz und Richtigkeit werden verletzt.

Die beschriebenen Herausforderungen gelten insbesondere für die Modellierung der Systemstruktur. Die Modellinhalte hängen von der verwendeten Modellierungssprache, -methode und -werkzeug ab. In der Beschreibung des gleichen Systems können in einer Systemstruktur, Elemente und Beziehungen vorhanden sein, die im nächsten Modell für nicht relevant gehalten wurden.

Die Nutzenpotenziale und die Herausforderungen zeigen den **Bedarf** auf für ein *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*. Die Bestandteile des Rahmenwerks sind dabei so zu erstellen, dass sie sprachunabhängig sind. Damit soll erreicht werden, dass Modellinhalte sowohl im Fall der Verwendung der gleichen Sprache als auch bei unterschiedlichen Sprachen vergleichbar sind. Das Rahmenwerk soll dabei aus den folgenden Bestandteilen bestehen:

- **Vorgaben zur Modellierung:** Kern des Rahmenwerks muss die Definition der zu spezifizierenden Inhalte einer plausiblen Systemstruktur sein. Ausgehend von der Definition des mechatronischen Systems sind Elemente und Beziehungen zu untersuchen, die in einer Systemstruktur dargestellt werden sollen. Diese sind nach ihrer Art der Darstellung in Klassen zu unterteilen. Mit der Klassifikation sind Bedingungen verbunden, die an die Beschreibung der Elemente und Beziehungen gelegt

werden. Darüber hinaus müssen weitere Bedingungen identifiziert werden, die einen Beitrag zur Vergleichbarkeit, Vollständigkeit sowie Richtigkeit der Systemstruktur leisten. Die Bedingungen sind in Form von Richtlinien zu beschreiben.

- **Überprüfung der Modellinhalte:** Die Einhaltung der aufgestellten Bedingungen und Richtlinien muss überprüft werden. Die Prüfung ist dabei so zu gestalten, dass sowohl eine manuelle als auch eine rechnerunterstützte Überprüfung möglich ist.
- **Konzept der softwaretechnischen Unterstützung:** Der Modellierungsaufwand kann softwaretechnisch reduziert werden. Hierzu soll ein Konzept beschrieben werden, wie Modellierungsschritte automatisiert werden können. Ebenfalls muss das IT-Konzept die softwaretechnische Überprüfung des Modells auf die Einhaltung der Vorgaben enthalten.
- **Strukturiertes Vorgehensmodell:** Abschließend muss das Rahmenwerk ein strukturiertes Vorgehen zur Erstellung einer plausiblen Systemstruktur beschreiben. Das Vorgehen soll die Bestandteile des Rahmenwerks integrieren und sich der Herausforderung der unterschiedlichen Anforderungen an die Formalisierung des Modells stellen.

2.8 Anforderungen

Aus der Problemanalyse resultieren folgende Anforderungen an ein *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*.

A1) Sprachenunabhängigkeit: Die Beschreibung der Systemstruktur erfolgt mit einer graphischen Modellierungssprache. Die Bestandteile des Rahmenwerks sind dabei so zu gestalten, dass sie mit den gängigen Modellierungssprachen beschrieben werden können. Dabei dürfen leichtgewichtige Anpassungen der Sprache vorgesehen werden.

A2) Systemstruktur als Basis für ein einheitliches Systemverständnis: Die Systemstruktur soll mechatronische Systeme in seinen Bestandteilen beschreiben können. Hierzu sollen Elemente und Beziehungen abgebildet werden, so dass die Systemstruktur das Systemdenken der Fachdisziplinen fördert und ein einheitliches Systemverständnis sicherstellt. Damit ist eine eindeutige Definition der Systemstruktur sowie der abzubildenden Elemente und Beziehungen verbunden. Die hierzu zu verwendenden Begrifflichkeiten sollen so gewählt sein, dass die verschiedenen Fachdisziplinen diese intuitiv nachvollziehen können.

A3) Vergleichbarkeit: Die Systemstruktur muss von allen Fachdisziplinen gelesen und vergleichbar interpretiert werden können. Dies setzt voraus, dass ein Sachverhalt in der Systemstruktur von allen Beteiligten gleich dargestellt wird. Durch den Fokus auf mechatronische Systeme sind die abzubildenden Sachverhalte vorgegeben. Die Definition der abzubildenden Elemente und Beziehungen muss sicherstellen, dass eine Vergleichbarkeit gegeben ist.

A4) Vollständigkeit: Die semiformalen Modellierungssprachen geben keine Aussage über die Modellierungstiefe. Damit kann bislang die Vollständigkeit einer Systemstruktur nicht überprüft werden. Durch den Fokus auf mechatronische Systeme sollen Annahmen getroffen werden, die Aussagen über die Vollständigkeit der Systemstruktur zulassen.

A5) Richtigkeit: Die richtige Darstellung des Systems soll sich auf die Syntax als auch auf die Semantik beziehen. Die Definition der Elemente und Beziehungen soll Bedingungen enthalten, so dass eine Überprüfung der Syntax möglich ist. Die semantische Richtigkeit ergibt sich erst aus dem Kontext des Systems und kann nur vom Anwender selbst bewertet werden. Hierzu sind Hilfsmittel zur Verfügung zu stellen, die bei der Bewertung der Richtigkeit unterstützen können.

A6) Klarheit: Die graphischen Modelle werden zur Kommunikation genutzt. Dies setzt voraus, dass die Modelle leserlich und anschaulich gestaltet werden (Grundsatz der Klarheit). Mit steigender Anzahl an Elementen und Beziehungen werden die Darstellungen komplex und unüberschaubar. Es bedarf Konzepte, die die Klarheit sicherstellen und damit die visuell wahrnehmbare Komplexität des Modells reduzieren.

A7) Vereinbarkeit von intuitiver Anwendung und formalisierter Beschreibung: Der Formalisierungsgrad der Systemstruktur muss zwei gegensätzlichen Ansprüchen gerecht werden. Zum einen wird eine geringe Formalisierung verlangt, so dass eine intuitive Anwendung möglich ist, zum anderen sollen die Daten rechnerintern auswertbar sein. Damit ist wiederum ein hoher Formalisierungsgrad notwendig. Es sollen Konzepte aufgezeigt werden, wie diese verschiedenen Formalisierungsgrade miteinander vereinbart werden können.

A8) Rechnerunterstützte Modellierung: Die Systemstruktur soll als Teil eines Systemmodells in einem Softwarewerkzeug modelliert werden. Der Aufwand der Modellierung soll dabei gering gehalten werden. Damit muss das Werkzeug bei der Modellierung sowie -pflege den Anwender unterstützen.

3 Stand der Technik

In diesem Kapitel wird ein Überblick zum aktuellen Stand der Technik gegeben. Dabei werden die drei Bestandteile zur Beschreibung des Systemmodells betrachtet: Modellierungssprachen, Methoden und Softwarewerkzeuge. In **Kapitel 3.1** werden Modellierungssprachen aufgezeigt, die für die Darstellung einer Systemstruktur geeignet sind. Es handelt sich hierbei stets um graphische Modellierungssprachen. Die Unterschiede in den Darstellungen sowie in den zu spezifizierenden Modellinhalten werden dabei aufgegriffen. Methoden bilden den Fokus in **Kapitel 3.2**. Dazu werden zwei Arten von Methoden vorgestellt. Methoden, die sich auf den Einsatz einer speziellen Modellierungssprache beziehen und speziell hierfür entwickelt wurden (METUS und CONSENS) und allgemeine Methoden der disziplinübergreifenden Produktentwicklung. Abschließend werden Softwarewerkzeuge in **Kapitel 3.3** vorgestellt, die den Systems Engineering-Ansatz verfolgen und damit die softwaretechnische Unterstützung der Modellerstellung darstellen.

Für die SysML sind Modellierungsrichtlinien vorhanden, auf die in **Kapitel 3.4** eingegangen wird. In **Kapitel 3.5** wurde zudem die Funktionsmodellierung untersucht, da diese teilweise als eine Form der Strukturbeschreibung aufgeführt wird. Eine abschließende Betrachtung fasst die verschiedenen Strukturbeschreibungen zusammen (**Kapitel 3.6**). Aus der Bewertung des Stands der Technik wird in **Kapitel 3.7** der Handlungsbedarf abgeleitet.

3.1 Modellierungssprachen

Im Folgenden werden graphische Modellierungssprachen beschrieben, die im Bereich des Model-Based Systems Engineering eingesetzt werden. Diese haben zum Ziel, das Systemdenken zu fördern und damit eine ganzheitliche Betrachtung des Systems zu ermöglichen. Daher bestehen die Sprachen meist aus vielen Modellkonstrukten, die unterschiedliche Aspekte des Systems adressieren. Hierzu zählen z.B. Aspekte wie Anforderungen oder Verhalten. Zu jeder Sprache werden diese kurz aufgeführt. Eine detailliertere Betrachtung wird auf die Darstellung der Systemstruktur gelegt.

3.1.1 METUS - Sprache

Die ID-Consult GmbH hat die Methode METUS und zugehörige gleichnamige Software entwickelt [TGH08]. Eine Modellierungssprache wird nicht explizit genannt. Die darzustellenden Elemente und deren graphische Notation ergeben sich aus der Methode und dem Werkzeug.

Wesentliche Aspekte der Sprache sind Anforderungen, Funktionen und die Systemstruktur. Anforderungen werden in einer Liste verwaltet und können bspw. in funktionale Anforderungen, Anforderungen an die Produkteigenschaft oder Umgebungsbe-

dingungen unterteilt werden [TG10]. Die Funktionsstruktur wird in einer Funktionshierarchie dargestellt (Bild 3-1). Systemelemente, die die untersten Funktionen erfüllen, werden erfasst und mit den entsprechenden Funktionen verknüpft. Die Elemente werden mit den Attributen Standard-, Varianz- oder optionales Element versehen. Zudem können Elementen Parameter, wie Gewicht oder Kosten, mitgegeben werden. Diese lassen sich in Ziel- und Ist-Werten unterteilen. Die Verknüpfung von Elementen zeigt die Module des Systems auf.

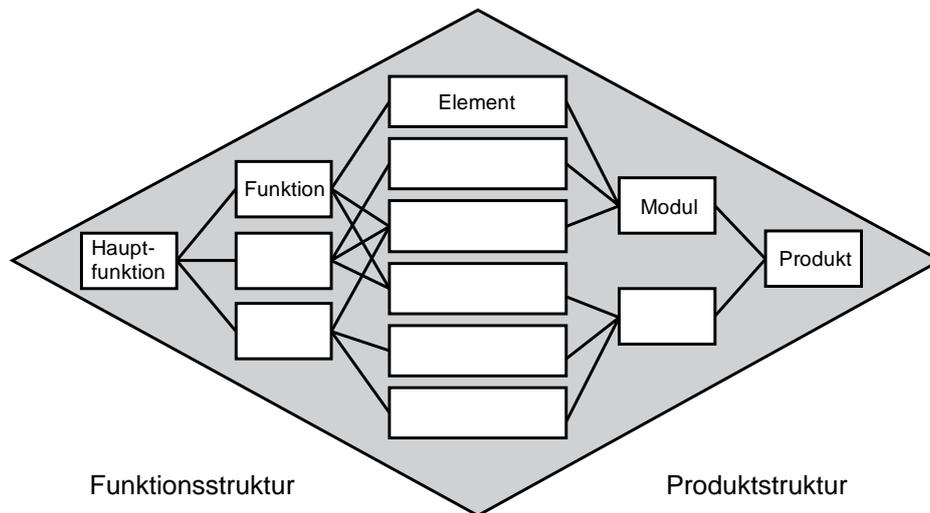


Bild 3-1: Darstellung der Funktionsstruktur und der Produktstruktur mit METUS (nach [TG10])

Bewertung: Die Sprachkonstrukte in METUS erlauben die Darstellung der Elemente und der Verknüpfung dieser zu Modulen. Die Wirkungsweise der Elemente wird nicht dargestellt, sondern implizit durch die Verbindung der Elemente mit den Funktionen ersichtlich. Die Integration des Systems in sein Umfeld, und damit die Interaktion des Systems mit dem Umfeld, wird nicht adressiert.

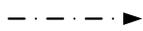
3.1.2 CONSENS - Sprache

Die Spezifikationstechnik CONSENS (CONceptual design Specification technique for the Engineering of complex Systems) wurde am Heinz Nixdorf Institut im Sonderforschungsbereich 614 entwickelt [ADG+09]. Sie basiert auf den Arbeiten von KALLMEYER, FRANK und GAUSEMEIER et al. [Kal98], [Fra06], [GEK01] und umfasst eine Modellierungssprache sowie eine Vorgehensweise zur Erstellung des Systemmodells. Ziel von CONSENS ist die ganzheitliche und disziplinübergreifende Beschreibung des Systemmodells im Rahmen der Konzipierung. Das System wird dabei durch sieben Aspekte beschrieben: Umfeld, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur, Gestalt und Verhalten. Im Rahmen des Forschungsprojekts VireS wurde die Sprache formal in einem Metamodell festgelegt [GLL12].

Das Umfeldmodell und die Wirkstruktur werden als einzelne Partialmodelle gesehen, die jedoch eng miteinander in Beziehung stehen. Das Partialmodell **Umfeld** definiert die Systemgrenze und beschreibt die Wechselwirkung des Systems mit seiner Umgebung. Umfeldelemente interagieren mit dem System. Zur Darstellung der Wechselwirkung werden Flussbeziehungen genutzt. Dabei wird zwischen den drei Flussarten Stoff, Energie und Information unterschieden. Zusätzlich steht eine weitere Beziehungsart zur Verfügung – die logische Verbindung. Diese wird genutzt, wenn keine der drei Flussarten zur Beschreibung des Zusammenhangs möglich ist (z.B. „ist umgeben von“, „enthält“). Das Umfeldmodell stellt eine Black-Box Betrachtung dar.

Die White-Box Betrachtung erfolgt im Partialmodell **Wirkstruktur**. Dieses beschreibt die Systemelemente, aus denen das System besteht, sowie deren Wechselwirkung. Auch in diesem Partialmodell werden die Flussarten Stoff, Energie und Information genutzt. Flussbeziehungen aus dem Umfeldmodell werden im System weiter spezifiziert, indem sie mit Systemelementen verbunden werden. Die Wirkstruktur bildet das zentrale Partialmodell der Spezifikationstechnik CONSENS. Tabelle 3-1 zeigt die Modellkonstrukte in CONSENS zur Systemstruktur.

Tabelle 3-1: Modellkonstrukte mit deren graphischer Notation und Semantik zur Beschreibung der Systemstruktur mit CONSENS

Modellkonstrukte	Graphische Notation	Semantik/Verfeinerung nach GEHRKE
Umfeldelement		Elemente außerhalb der Systemgrenze/ <i>keine</i>
System/ Systemelement		System entspricht dem zu entwickelnden Produkt und besteht aus Systemelementen/ Hardware-, Software- und logische Elemente
Energiefluss		Energiebeziehung zwischen zwei Elementen/ Energieart
Stofffluss		Stoffbeziehung zwischen zwei Elementen/ Stoffart
Informationsfluss		Informationsbeziehung zwischen zwei Elementen/ Informationsart
Logische Beziehung		Semantik ergibt sich durch die Bezeichnung der Beziehung

GEHRKE führt eine Formalisierung der Elemente und Beziehungen durch, mit dem Ziel, ausgehend von Funktionen eine automatische Suche von Systemelementen zu ermöglichen [Geh05]. Die Elemente werden in Hardware-, Software- oder logische Elemente

unterteilt. Zudem werden den Systemelementen Attribute annotiert. Diese können vom Modellierer je Element spezifisch definiert werden. Die Beziehungen Energie, Stoff und Information werden ebenfalls durch die Möglichkeit verfeinert, die Art der Beziehung zu benennen. Um welche Art es sich dabei handelt, wird nicht vordefiniert. Beispiele zeigen dem Anwender, welche er dafür wählen kann.

GÖHNER et al. zeigt in [Göh13], wie eine Konsistenzprüfung im frühen mechatronischen Entwurf durchgeführt werden kann. Hierzu wird das System mit den Partialmodellen von CONSENS erstellt. Zur Durchführung der Prüfung wird ein Metamodell beschrieben. Regeln für die Konsistenzprüfung richten sich an den formalen Aufbau (Syntax). Ein ähnliches Metamodell wurde im Rahmen der Entwicklung des dedizierten Softwarewerkzeugs (Mechatronic Modeller, Kapitel 3.3.2) erstellt und implementiert (vgl. [GLL12]). Ferner betont GÖHNER et al. die Notwendigkeit von Regeln zur inhaltlichen Vollständigkeit und Richtigkeit. Lösungen werden hierzu nicht vorgestellt.

Bewertung: CONSENS enthält Modellkonstrukte zur Beschreibung der Systemstruktur, die in zwei Partialmodellen erfolgt (Umfeld und Wirkstruktur). Die wenigen Modellkonstrukte zur Beschreibung der Systemstruktur sind einfach zu erlernen und durch die gewählten Begrifflichkeiten intuitiv anwendbar. Die Unterscheidung der Beziehungen ist auf mechatronische Systeme ausgerichtet, so dass Interaktionen dargestellt werden können. GEHRKE, GAUSEMEIER et al. sowie GÖHNER et al. stellen Lösungen vor, die eine Überprüfung der Syntax ermöglichen [Geh05], [GLL12], [Göh13]. Hierzu gehört z.B. auch die Bedingung, dass nur Elemente miteinander verbunden werden dürfen, die über gleiche Schnittstellen verfügen. Dennoch reicht die Formalisierung der Wirkstruktur nicht aus, um eine Vergleichbarkeit, Vollständigkeit oder Richtigkeit zu erreichen.

3.1.3 OPM

OPM steht für Object-Process Methodology und wurde von DORI zur Konzipierung komplexer Systeme entwickelt. Die Sprachkonstrukte bestehen zum Teil aus graphischen Konstrukten, die in entsprechenden Diagrammen dargestellt werden (Object-Process Diagrams, OPD) und einem Set aus Textbausteinen (Object-Process Language, OPL) [Dor02].

Dabei können aus den graphischen Modellen mit den Textbausteinen die Modellinhalte in Text beschrieben abgeleitet werden. Die Motivation der Sprachentwicklung war, aus wenigen graphischen Konstrukten, die wesentlichen Aspekte des Systems – Struktur und Verhalten – darstellen zu können. Die beiden Aspekte werden in einem Diagramm beschrieben und nicht in Partialmodellen getrennt. DORI ist der Ansicht, dass eine Trennung der Informationen zu Komplexität in der Modellierung führt.

Für die graphische Beschreibung des Systems stehen die Konstrukte Objekt, Zustand und Prozess zur Verfügung (Bild 3-2). Ein Objekt ist dabei ein Element das über gewisse Zeit existiert. Dabei kann ein Objekt Zustände haben, in denen es sich befinden kann.

Über einen Prozess wird ein Objekt transformiert. Die Beziehungen zwischen den Konstrukten entsprechen Links, unterteilt in strukturelle und prozesstechnische Verbindungen. Die strukturelle Beziehung ist eine permanente Verbindung zwischen zwei Objekten. Hingegen besteht zwischen einem Objekt, bzw. dem Zustand des Objekts und einem Prozess eine prozesstechnische Beziehung.

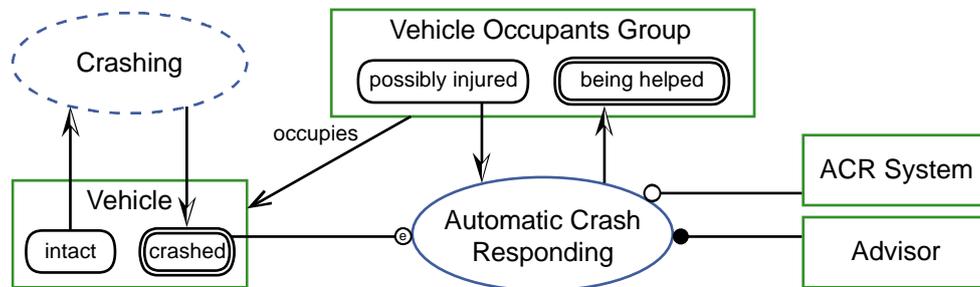


Bild 3-2: Beispielsystem dargestellt mit OPM (nach [Dor02])

Die Komplexität des Modells wird mit dem Konzept der Abstraktion beherrscht. Das beinhaltet die Möglichkeit, jedes Konstrukt (Objekt, Zustand und Prozess) detailliert aber auch abstrahiert zu beschreiben. So entstehen mehrere Ebenen für die Konstrukte, die entfaltet oder komprimiert dargestellt werden können [Dor02].

Bewertung: OPM ist eine einfache und intuitiv anwendbare Modellierungssprache zur Beschreibung komplexer Systeme. Mit der Möglichkeit, unterschiedliche Ebenen der Konstrukte zu beschreiben, wird ein Beitrag zur Klarheit geschaffen. Die Modelle können übersichtlich dargestellt werden. Mechatronische Systeme, als Teil von komplexen Systemen, können ebenfalls mit der Sprache abgebildet werden. OPM enthält jedoch zu wenige Konstrukte, um die Wirkbeziehungen zwischen den Elementen zu beschreiben. Die gegebenen Vorgaben reichen nicht aus, um Vergleichbarkeit der Modelle sicherzustellen.

3.1.4 SysML

Die SysML unterscheidet sich von den eben genannten Sprachen dahingehend, dass die Sprache unabhängig von jeglicher Methode entwickelt wurde. SysML steht für Systems Modeling Language und wurde für die Konzepte des Systems Engineering ausgerichtet. Das *International Council on Systems Engineering* (INCOSE) hat sich im Jahr 2001 zum Ziel gesetzt, eine standardisierte Sprache des Systems Engineering zu erstellen. Die Basis bildete die UML in der Version 2.1. Nach Anpassungen und Überarbeitungen wurde im April 2006 die SysML als Standard anerkannt und 2007 SysML 1.0 von der OMG (Object Management Group) offiziell veröffentlicht. An dem Standard wird stetig weitergearbeitet und neue Versionen veröffentlicht. Aktueller Stand 2013 ist die Version 1.3 [OMG12].

Mit Hilfe von SysML lassen sich die Aspekte **Anforderungen**, **Struktur**, **Parameter** und **Verhalten** eines Systems beschreiben. Hierzu stehen dem Anwender neun Dia-

grammarten zur Verfügung (Bild 3-3). Teilweise wurden Diagrammarten aus der UML direkt übernommen (Paket-, Zustands-, Use Case-, Sequenz-Diagramm), andere für SysML modifiziert (Blockdefinitions-, Internes Block-Diagramm) und zwei sind neu hinzugekommen (Zusicherungs-, Anforderungs-Diagramm). Bei den modifizierten Diagrammen wurde, neben den enthaltenen Konstrukten, auch die Bezeichnung des Diagramms angepasst. So entspricht das Blockdefinitionsdiagramm dem Klassendiagramm und das Interne Blockdiagramm dem Kompositionsstrukturdiagramm der UML. Die SysML [OMG12] hat darüber hinaus auch den Profilmechanismus der UML übernommen. Diese ermöglicht dem Benutzer, eine Anpassung der SysML-Modellelemente in Form von Stereotypen oder Tagged Values vorzunehmen [Alt12b]. Dabei erfolgt keine Änderung des Metamodells. In diesem Fall handelt es sich um eine leichtgewichtige Erweiterung – einen Dialekt der SysML.

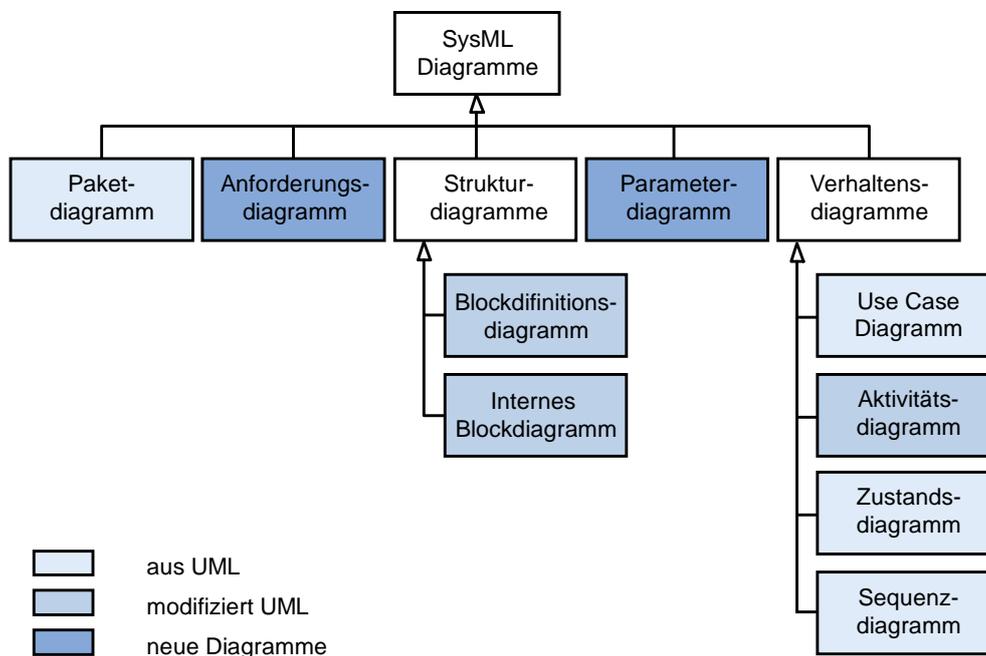


Bild 3-3: Diagramme der SysML nach [FMS12]

Die einzelnen Diagramme bilden eine Sicht auf das Modell. Das Modell selbst ist in dem sogenannten Repository enthalten. Somit kann ein und dasselbe Modellelement mehrfach in verschiedenen Diagrammen dargestellt werden [Alt12b].

Die Systemstruktur wird im **Blockdefinitionsdiagramm (bdd)** und **internes Blockdiagramm (ibd)** beschrieben. Blöcke sind Bestandteile des Systems und werden im bdd definiert. Die Zusammenhänge der Blöcke werden hingegen im ibd dargestellt. Diese Trennung ist eine Folge der Objektorientierung, die Grundlage der SysML ist. Hierzu wird ein Element (Block) als Typ definiert und kann erst dann in einer Rolle im Kontext des Systems beschrieben werden [Wei06]. Die Elemente verfügen über Schnittstellen in Form von Ports, die eine Interaktion mit anderen Elementen ermöglichen. Die Beziehung zwischen Elementen wird im ibd durch Konnektoren modelliert. Die graphische Notation orientiert sich an der UML. Ein Block wird in der SysML in Form eines wei-

ßen Rechtecks dargestellt und ein Port in Form eines Quadrats. Konnektoren sind gestrichelte Linien, die im Fall einer Flussbeziehung durchgezogen dargestellt wird.

Bewertung: SysML sieht eine strukturelle Beschreibung des Systems in den Diagrammarten ibd und bdd vor. Die Modellkonstrukte sind dabei nicht speziell auf mechanische Systeme ausgelegt, da die Modellierungssprache den Anspruch hat, allgemeingültig zu sein. Durch die Standardisierung können Modelle ineinander überführt werden. Dies ist jedoch nur bedingt möglich. Durch die allgemeingültige Definition der Konstrukte sind keine Regeln vorhanden, die zu einer Vergleichbarkeit, Vollständigkeit oder Richtigkeit der Modelle beitragen. Dieses kann jedoch durch Profile abgefangen werden. Daher eignet sich die SysML sehr gut für die formalisierte Beschreibung. Das Erlernen der Sprache fällt den Ingenieuren meistens schwer, da hierzu die Konzepte der Objektorientierung bekannt sein müssen. Hingegen können die graphischen Darstellungen schnell nachvollzogen werden, so dass das Lesen der Modelle keine Schwierigkeiten bereitet [BC10], [FMS12], [SMM+12].

3.2 Modellierungsmethoden

In diesem Abschnitt werden die Methoden der modellbasierten Entwicklung vorgestellt. Die Methoden METUS und CONSENS sind als Spezifikationstechniken entstanden. Das heißt, dass zu einer Vorgehensweise eine eigene Sprache (Kap. 3.1.1 und 3.1.2) definiert wurde. Anders ist dies der Fall bei den Methoden nach WEILKIENS, FRIEDENTHAL et al. und ALT, die die SysML (Kap. 3.1.4) nutzen – jedoch stets mit einer eigenen Anpassung (SysML-Profil).

3.2.1 METUS - Methode

Die Spezifikationstechnik METUS der ID-Consult GmbH beinhaltet eine Methode, mit der in sieben Schritten das Produktkonzept erstellt wird [TGH08]. Bild 3-4 zeigt die METUS-typische Rautenstruktur mit den Schritten der Methode.

Im ersten Schritt wird das **Anforderungskonzept** definiert. Hierzu wird eine Anforderungsanalyse durchgeführt, indem systematisch Anforderungen aus Kundensicht und aus Unternehmenssicht erfasst werden. Die Produktarchitektur vergleichbarer Produkte wird mit dem Ziel untersucht, Optimierungspotentiale aufzudecken und neue Anforderungen zu generieren.

Aus der Hauptfunktion des Systems werden Teilfunktionen gebildet und in einer **Funktionsstruktur** dargestellt. In diesem Schritt wird, losgelöst von der technischen Realisierung, die funktionale Betrachtung des Systems vorgenommen. Funktionale Anforderungen können definiert und präzisiert werden.

Aus der Funktionsstruktur wird eine **Produktstruktur** definiert. Dabei findet eine Art „Übersetzung“ der Funktionen in Elemente statt. Diese werden daraufhin zu Modulen zusammengefasst und ergeben schließlich das Gesamtprodukt.

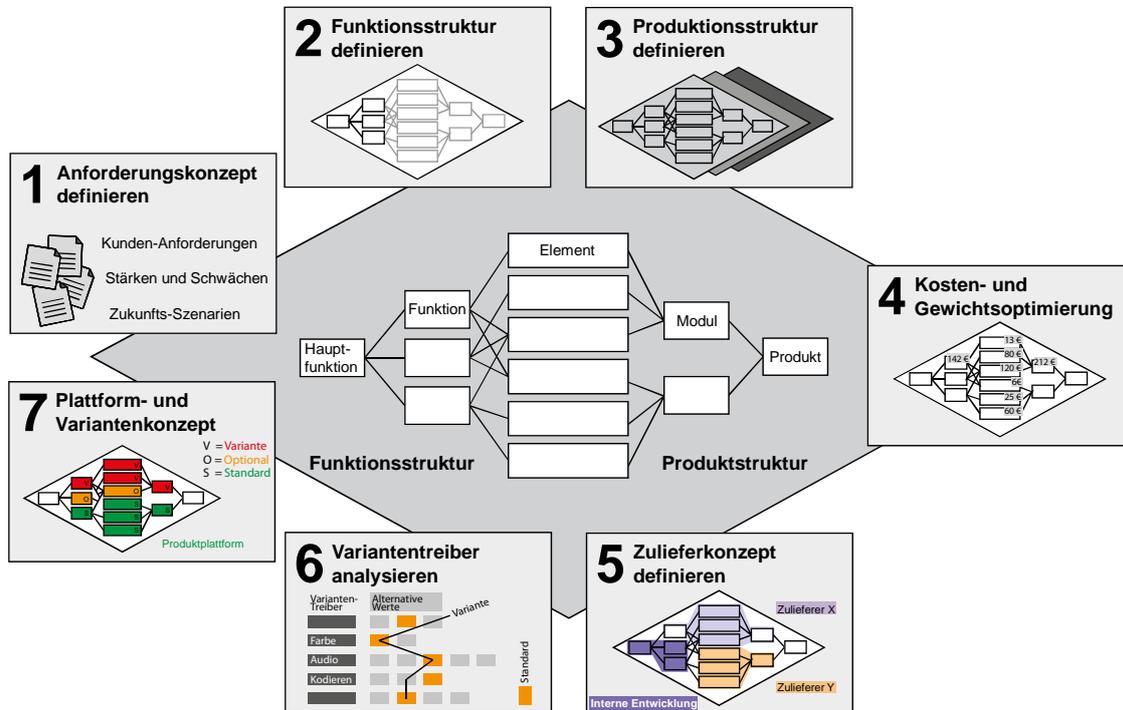


Bild 3-4: 7 Schritte der METUS-Methode (nach [TGH08], [TG10])

Die folgenden Schritte beinhalten Optimierungsverfahren, die abhängig von der Anwendung zum Einsatz kommen. **Kosten und Gewicht** sind wesentliche Optimierungsaspekte. Hierzu wird die Verknüpfung der Funktionen mit den Elementen und Modulen genutzt, um Zielkosten und -gewicht zuzuordnen und zu optimieren.

Anschließend erfolgt die Definition von **Standards, Optionen und Varianten**. Jedem Element wird ein entsprechender Marker gesetzt. Die Varianten werden für jedes Element in Form einer Ausprägung beschrieben. Daraufhin erfolgt eine **Produktkonfiguration**, indem die Ausprägungen der Varianten geeignet kombiniert werden. Die Zuordnung der Elemente und Module zu internen Entwicklung, Fertigung und Montage sowie Lieferanten endet in einem **Zuliefererkonzept**.

Bewertung: Die METUS-Methode ist vor dem Hintergrund der Modularisierung entstanden. Die funktionale Betrachtung des Systems erfolgt hierarchisch. Daraus werden physische Elemente abgeleitet, die wiederum in einer Hierarchie zusammengefasst werden. Die Interaktion der Elemente wird nicht betrachtet. Ebenfalls fehlt eine Ausrichtung auf mechatronische Systeme, wobei dieses nicht ausgeschlossen wird.

3.2.2 CONSENS - Methode

Die Vorgehensweise zur Erstellung des Systemmodells ist Teil der Spezifikationstechnik. Die Aspekte – und damit die entsprechenden Partialmodelle – sind im Wechselspiel zu erstellen, wenngleich eine gewisse Reihenfolge vorgesehen ist [GFD+09].

Zu Beginn wird im **Umfeldmodell** die Systemgrenze festgelegt. Dabei wird das System als Black-Box betrachtet. Elemente aus dem Umfeld, die mit dem System in Beziehung stehen, werden identifiziert und die Wechselwirkung modelliert. **Anwendungsszenarien** beschreiben verschiedene Situationen und das gewünschte Verhalten des Systems in dieser Situation. Auf Basis des Umfeldmodells und der Anwendungsszenarien werden Anforderungen abgeleitet. Diese werden in funktional und nichtfunktional unterschieden. Im nächsten Schritt werden alle funktionalen Anforderungen in **Funktionen** in der Form „Substantiv Verb“ übersetzt und in der Funktionshierarchie abgebildet. Eine Unterteilung der Gesamtfunktion in Teilfunktionen erfolgt fortlaufend, bis für die untersten Funktionen Lösungsmuster oder Wirkprinzipien ausgewählt werden. Aus der Summe aller Wirkprinzipien und Lösungsmuster wird eine geeignete Kombination zur Wirkstruktur synthetisiert. Dieser Schritt erfolgt meistens über einen Morphologischen Kasten, der jedoch nicht Bestandteil der Spezifikationstechnik CONSENS ist.

Die Definition der Systemarchitektur erfolgt in dem Partialmodell **Wirkstruktur**. Diese stellt den Blick in das System dar. In der Wirkstruktur werden somit der grundsätzliche Aufbau und die prinzipielle Wirkungsweise eines mechatronischen Systems definiert. Alle gestaltbehafteten Systemelemente werden im Partialmodell **Gestalt** durch eine Form abgebildet. Ziel ist die Abstimmung der Bauräume, Anordnung, Lage, Art der Wirkflächen und Wirkorte. Werden weitere Systemelemente benötigt, sind sowohl das Partialmodell Wirkstruktur als auch Gestalt anzupassen.

Für das **Verhalten** werden die Anwendungsszenarien als Ausgangsbasis genommen. Im Partialmodell Verhalten–Zustände wird nun festgelegt, welche Ereignisse einen Zustandsübergang bewirken. Ist ein Regelkreis vorhanden, kann die Regelung als eine Abfolge von Aktivitäten im Partialmodell Verhalten-Aktivitäten beschrieben werden.

Bewertung: Das Vorgehen der Spezifikationstechnik CONSENS enthält zwei Aspekte zur Strukturbeschreibung. Das Umfeldmodell betrachtet dabei die Einbettung des Systems in seine Umgebung, während die Wirkstruktur den Blick in das System darstellt. Beide Aspekte beschreiben die Wechselwirkung der Elemente untereinander. Die Methode geht auf die Bestandteile eines mechatronischen Systems nicht explizit ein, so dass keine Unterteilung der Elemente durch die Methode vorgegeben wird. Damit verbunden werden keine Vorgaben zur Vergleichbarkeit, Vollständigkeit oder Richtigkeit der Systemstruktur gemacht.

3.2.3 FAS

Die FAS-Methode (FAS, engl.: functional architecture for systems) beschreibt das Vorgehen von den Anforderungen bis zur Erstellung der funktionalen Systemarchitektur [LW10]. WEILKIENS und LAMM stellen die Methode mit einem angepassten SysML-Profil vor – die Methode selbst ist sprachenunabhängig.

In Bild 3-5 sind die einzelnen Schritte der Methode in einem SysML-Aktivitätsdiagramm dargestellt. Die Ausgangsbasis bilden die **funktionalen Anforderungen** des Systems, die durch **Anwendungsfälle** verfeinert werden.

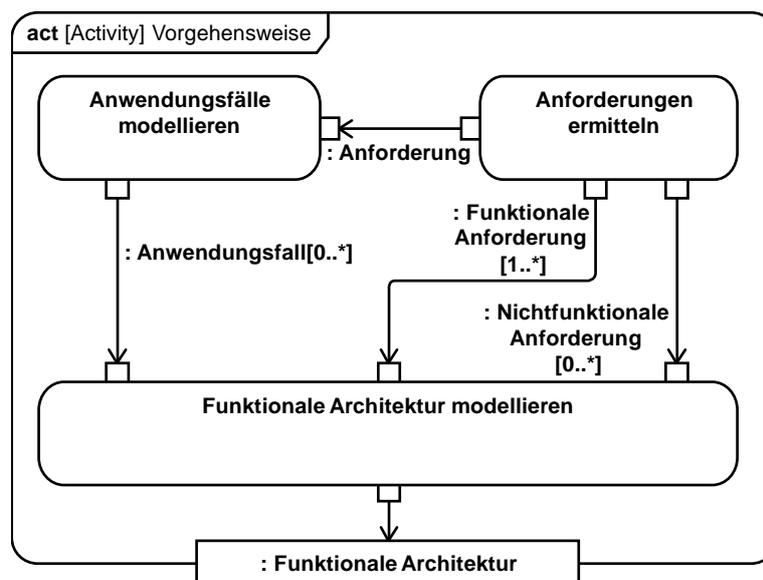


Bild 3-5: Vorgehensweise der FAS-Methode dargestellt mit einem SysML-Aktivitätsdiagramm (nach [LW10])

Ein Anwendungsfall beschreibt die Wechselwirkung der Funktionen mit allen externen Elementen (Umfeldelemente). Einen wesentlichen Anteil der externen Elemente bilden die menschlichen Akteure (Anwender).

Der nächste Detaillierungsschritt erfolgt mit den **Aktivitätsdiagrammen** in Form einer Baumstruktur. Dabei basiert die Hierarchie auf der Aufrufsemantik (eine Aktivität wird im Kontext der Oberaktivität aufgerufen). Die Wurzeln des Baums sind die Anwendungsfälle, so dass für alle Anwendungsfälle damit ein Wald von Aktivitätsbäumen entsteht [KLW11].

Die Methode unterscheidet darüber hinaus zwischen Funktionen und funktionalen Elementen, die als funktionale Blöcke im Modell repräsentiert werden. Ein **funktionales Element** stellt eine abstrakte Systemkomponente dar, die mindestens ein Input und Output besitzt. Diese stehen über eine **Funktion** in Beziehung. Die funktionalen Blöcke werden im internen Blockdiagramm miteinander verbunden. Standardports werden dabei für die Flüsse von Signalen oder Informationen verwendet und Flowports für Mate-

rie-, Kraft- oder Energie-Flüsse. Es entsteht eine Funktionsstruktur, deren Hierarchie eine „Enthält“-Beziehung bedeutet.

Für die Implementierung wird eine physische Lösung benötigt, die die Funktionen umsetzt. Zu einer funktionalen Architektur können verschiedene **physische Architekturen** entwickelt werden, so dass eine funktionale Architektur über Technologiegenerationen gültig bleiben kann.

In [KLW11] stellt WEILKIENS et al. eine softwaretechnische Unterstützung der Methode vor, mit der automatisch die funktionale Architektur generiert werden kann. Dabei wurden Vorgehensweisen erarbeitet und im Softwaretool Artisan Studio® implementiert. Wenig anspruchsvolle Modellierungsschritte wurden automatisiert, so dass der Modellierer neben einer Zeitersparnis, Flüchtigkeitsfehler vermeidet.

Bewertung: Die FAS-Methode beinhaltet zwei Arten von Architekturen: die funktionale und physische¹⁰ Architektur. Die funktionale Architektur entspricht der Funktionsstruktur, wie sie in den Konstruktionsmethoden gelehrt wird [PBF+07], [KK94], [Rot00]. Durch den Einsatz der Modellierungssprache SysML wird aufgezeigt, wie die Funktionsstruktur mit der physischen Struktur zusammenhängt und modelliert wird (z.B. durch die «allocation»-Beziehung). Die Umsetzung in ein Softwarewerkzeug bietet die Möglichkeit, bei aufwendigen und sich wiederholenden Schritten unterstützt zu werden.

3.2.4 SYSMOD

SYSMOD beschreibt ein Vorgehen zur Modellierung komplexer Systeme [Wei06]. Als Modellierungssprache für die Beschreibung des Systemmodells wird ein SYSMOD-Profil der SysML genutzt.

Das Vorgehen unterteilt sich in sieben übergeordnete Schritte (Makro-Modelle): Anforderungen ermitteln, Systemkontext modellieren, Anwendungsfälle modellieren, Fachwissen modellieren, Glossar erstellen, Anwendungsfälle realisieren. Zu jedem Schritt existiert ein verfeinertes Vorgehen, das in einem Aktivitätsdiagramm dargestellt wird. Bild 3-6 zeigt das Vorgehen für den übergeordneten Schritt Anwendungsfälle realisieren.

Die bereits modellierten Anwendungsfälle bilden die Ausgangsbasis. Für jeden Anwendungsfall wird ein Interaktionsmodell erstellt. Dieses beschreibt in Form von Sequenzdiagrammen die Wechselwirkung des Systems mit dem Benutzer/den Akteuren. Daraus werden die Schnittstellen des Systems abgeleitet. Im nächsten Schritt wird die Sys-

¹⁰ In den Veröffentlichungen wird hierzu auch der Begriff physikalisch verwendet. Gemeint ist eine technische Realisierung des Systems durch physische Elemente.

temstruktur dargestellt. Diese beinhaltet die Systembausteine, die Informationsflüsse zwischen den Bausteinen und damit die Schnittstellen der Bausteine.

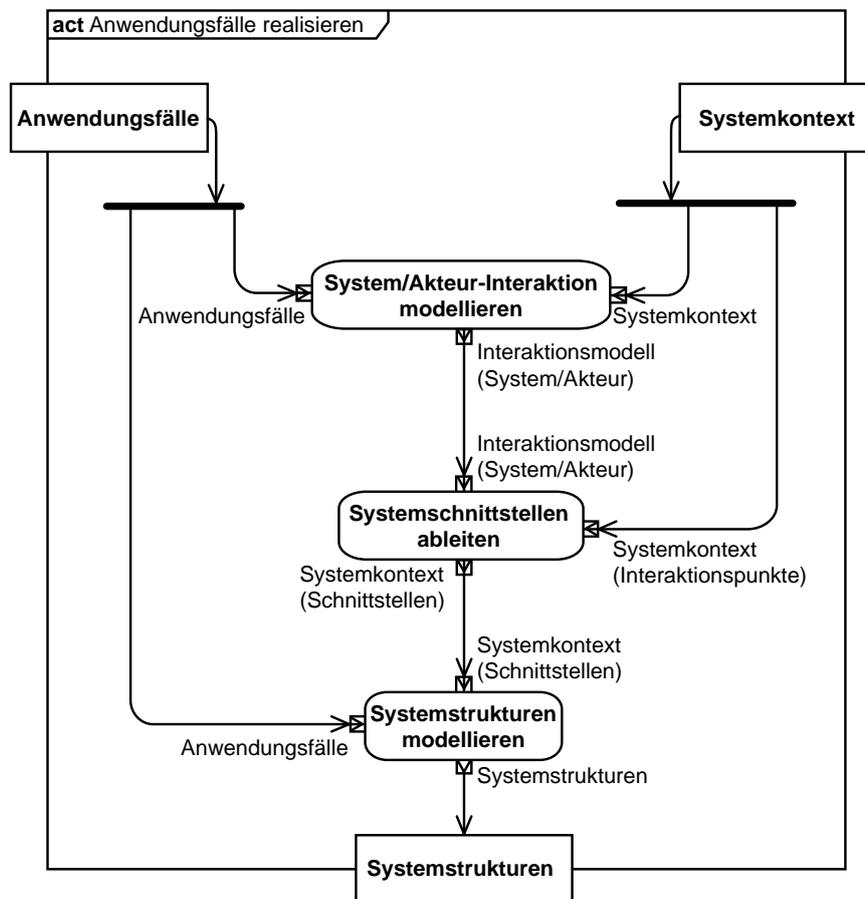


Bild 3-6: Vorgehen für den übergeordneten Schritt „Anwendungsfälle realisieren“ (nach [Wei06])

Das mit SYSMOD erstellte Systemmodell stellt eine abstrakte Lösung dar. Dazu zählen die Struktur und das Verhalten des Systems, nicht jedoch die Implementierung. Diese ergibt sich im Entwicklungsprozess erst in der Konkretisierung, wenn Technologieentscheidungen getroffen werden.

Die Beschreibung der einzelnen Schritte erfolgt in Form eines Steckbriefs (Bild 3-7). Neben einer kurzen Erläuterung werden die ein- und ausgehenden Daten aufgeführt und Leitfragen formuliert. Aus den einzelnen Steckbriefen ergibt sich ein Werkzeugkasten. Dieser kann für beliebige Projekte und Produkte angepasst werden [Wei06].

Bewertung: Die Methode SYSMOD ist aus der Softwareentwicklung entstanden und für die Entwicklung technischer Systeme angepasst worden. Jedoch sind die Vorgehensschritte am Verhalten des Systems orientiert, so dass der Fokus weiterhin auf der Software liegt. Die Systemstruktur wird als ein Konstrukt aus Systembausteinen beschrieben, wobei ein Baustein eine Software, Hardware, Person oder eine beliebige andere Einheit sein kann. An der Definition der Schnittstelle wird der softwareintensive

Fokus deutlich [Wei06]: „Die Schnittstelle definiert ein Verhalten mit einer Liste von Operationen.“

Steckbrief Systemstrukturen modellieren	
	<p>Ein- und ausgehende Daten</p> <p>Systemkontext [Schnittstellen] System mit Schnittstellenbeschreibung der Interaktionspunkte</p> <p>Anwendungsfälle Dienstleistungen des Systems</p> <p>Systemstrukturen Beschreibung der statischen Strukturen des Systems</p>
<p>Beschreibung</p> <p>Modellieren Sie die Systembausteine und ihre Zusammensetzung, die für das Gesamtsystem notwendig sind, um die Anforderungen zu erfüllen</p>	
<p>Leitfragen</p> <ul style="list-style-type: none"> <input type="checkbox"/> Welche Bausteine werden zur Umsetzung der Anwendungsfälle/ Anforderungen benötigt? <input type="checkbox"/> Wie ist ein Baustein aufgebaut? <input type="checkbox"/> Wie sind die Bausteine miteinander verbunden? <input type="checkbox"/> Welche Interaktionspunkte und Schnittstellen haben die Bausteine? 	

Bild 3-7: Beispiel eines Steckbriefs der SYSMOD Methode (nach [Wei06])

3.2.5 OOSEM

Die objektorientierte Systems Engineering Methode OOSEM (object oriented systems engineering method) wurde 1998 als Ergebnis einer Kooperation zwischen Lockheed Martin Corporation und dem Systems und Software Konsortium vorgestellt [FMS12]. Eine Weiterentwicklung und Anpassung erfolgt seit 2002 in der INCOSE OOSEM Arbeitsgruppe. Die szenariogetriebene Methode verläuft Top-Down und nutzt SysML als Modellierungssprache zur Beschreibung des Systemmodells.

OOSEM ist ein Bestandteil eines übergeordneten Entwicklungsprozesses bestehend aus den Phasen Management des Entwicklungsprozesses, Spezifikation und Entwurf des Systems sowie Integration und Verifikation. Bild 3-8 zeigt die wesentlichen Schritte der OOSEM, die den Fokus auf die Spezifikation und den Entwurf des Systems legt.

Die Struktur wird auf zwei Arten modelliert: logische und physische Architektur. Die logische Architektur enthält Systemelemente, die eine Funktion erfüllen, jedoch noch keine Aussage über die Realisierung liefern. Die konkrete Umsetzung wird durch physische Elemente beschrieben. Der Zusammenhang zwischen logischen und physischen Elementen wird durch die «allocation»-Beziehung modelliert.

Bewertung: Die Methode OOSEM beschreibt, wie die FAS-Methode, die Struktur auf zwei Abstraktionsebenen. Während bei der FAS-Methode die Funktionsstruktur die oberste Ebene darstellt, wird in dieser Methode eine logische Ebene bevorzugt. Diese

enthält damit nicht abstrakte Funktionen sondern Elemente, die später konkretisiert und in physische Elemente überführt werden. Eine Ausrichtung auf eine Klasse von Systemen ist nicht vorgegeben. Damit wird auch nicht auf die Bestandteile eines mechatronischen Systems eingegangen.

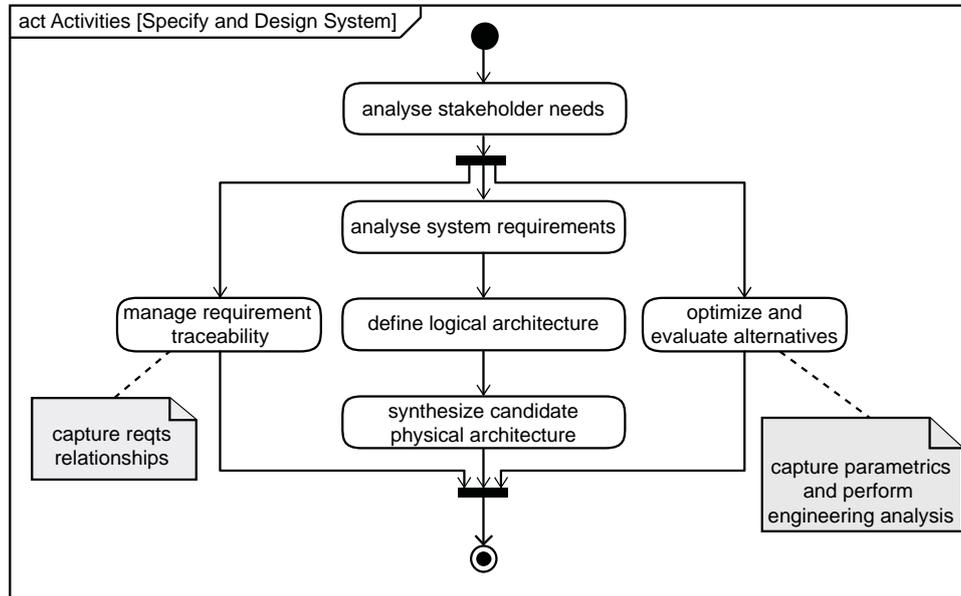


Bild 3-8: Vorgehensweise OOSEM (nach [FMS12])

3.2.6 Funktionale und technische Entwicklung nach ALT

ALT beschreibt keine eigene Vorgehensmethode, sondern orientiert sich an bestehenden Vorgehensweisen, wie dem V-Modell der Softwareentwicklung. Im Fokus steht die Unterstützung der Entwicklungsmethoden mit der Modellierungssprache SysML. Die Entwicklung der Struktur unterteilt er in die funktionale und technische Sicht [Alt12b].

Funktionale Entwicklung

Mit dem Fokus auf die Funktionalität soll in diesem Schritt geklärt werden, was das System können, bzw. leisten soll. Hierzu fallen Ergebnisse an, wie bspw. die Anwendungsfälle, funktionale Anforderungen, funktionale Architektur und weitere Verhaltensbeschreibungen. Die funktionale Architektur enthält Systemelemente, die gewissen Annahmen anhand des gegebenen Systemkontextes treffen, jedoch weitestgehend abstrakt bleiben. Übergeordnet wird nach dem Prinzip Eingabe-Verarbeitung-Ausgabe eine Architektur aufgebaut.

Technische Entwicklung

Die technische Sicht zeigt die Realisierung der abstrakt beschriebenen funktionalen Elemente. Die Modellierung erfolgt wiederum in zwei Architekturen: technisch-physi-

kalisch¹¹ und der technischen Wirkkettenarchitektur. Die technisch-physikalische Sicht enthält die physischen Systemelemente. Damit ist die Software in dieser Sicht nicht enthalten. Hierzu wird eine Wirkkettendarstellung erzeugt. Diese baut ebenfalls auf dem EVA-Prinzip auf und gibt die Eingangssignal- und die Ausgangskette vor.

Bewertung: ALT zeigt zwei Strukturbeschreibungen auf. Dabei trennt er zwischen funktionaler und technischer Sicht. In beiden Fällen wird jedoch auf Ebene der Elemente gearbeitet. Dabei stellt die funktionale Sicht eine abstrahierte Form der technischen Sichten dar. Mit den Grundarchitekturen werden wiederkehrende Strukturen definiert, die im Fokus die Eingabe und Ausgabe von Signalen betreffen. Diese macht die Überprüfung der Architektur möglich. Die Betrachtung über die Sensor-Aktorkette hinaus erfolgt nicht. Die Beziehungen innerhalb der Sichten werden nicht weiter charakterisiert.

3.3 Modellierungswerkzeuge

Im Folgenden werden Modellierungswerkzeuge aufgezeigt, die die Modellierungssprachen aus Kapitel 3.1 und/oder Methoden aus Kapitel 3.2 softwaretechnisch unterstützen. Für die SysML existieren mehrere Softwarewerkzeuge, die ursprünglich den Fokus auf der UML-Implementierung hatten. Diese wurden soweit angepasst, dass damit die SysML in Teilen abgedeckt wird. Alle Werkzeuge ähneln sich in der Struktur – sie unterscheiden sich im Wesentlichen in der Lizenzierung, Benutzungsfreundlichkeit, Umfang der Add-Ins sowie der Möglichkeiten, eigene Add-Ins zu implementieren. In Kapitel 3.3.3 wird repräsentativ für die SysML-Werkzeuge Enterprise Architect® (mit SysML Plug-In) vorgestellt. Zu den weiteren Softwarewerkzeugen zählen beispielsweise Cameo System Modeler (ehemals: MagicDraw mit SysML Plug-In), Rhapsody, Papyrus 4 SysML und Artisan Studio, auf die jedoch nicht weiter eingegangen wird.

3.3.1 METUS - Software

Das Softwarewerkzeug METUS wurde zur Unterstützung der gleichnamigen Methode von der ID-Consult GmbH entwickelt (vgl. Kapitel 3.2.1). Die Software wurde als eine Erweiterung der PLM-Lösung Teamcenter von Siemens PLM Software integriert und ermöglicht einen bidirektionalen Datenaustausch.

Die Benutzungsoberfläche besteht aus mehreren Fenstern, die beliebig angeordnet werden können und die verschiedenen Informationen über das System enthalten. Dabei beinhaltet ein graphischer Editor die Funktionsstruktur und die Produktstruktur, wie in Bild 3-1 angedeutet. In diesem können Elemente hinzugefügt oder überarbeitet werden. Eine Toolpalette ist nicht vorhanden, da das Anlegen über die Auswahlmeneü erfolgt.

¹¹ Physikalisch und physisch werden im Buch synonym verwendet.

Eine Drag-and-Drop Funktion ist implementiert, so dass Inhalte, die bereits in Tabellen vorliegen, einfach in den Editor gezogen werden können.

Bewertung: Das Werkzeug ist an die gleichnamige Methode angepasst. Die Handhabung und Modellierung der Informationen ist intuitiv gestaltet. Durch Zusatzfunktionen wird der Benutzer bei der Modellierung unterstützt. Die Daten sind formalisiert im Repository (Datenmodell) enthalten und werden z.B. durch die Anbindung an Teamcenter mit Entwicklungsdokumenten verbunden. Eine Auswertung der Modellinhalte ist möglich. Dadurch, dass die Methode die Wirkungsweise der Elemente nicht fokussiert, können diese im Werkzeug nicht modelliert werden.

3.3.2 Mechatronic Modeller

Der Mechatronic Modeller ist ein dediziertes Softwarewerkzeug für die Spezifikationstechnik CONSENS (vgl. Kapitel 3.1.2 und 3.2.2). Die prototypische Umsetzung des Werkzeugs erfolgte im Rahmen des BMBF Verbundprojekts VireS – Virtuelle Synchronisation von Produktentwicklung und Produktionssystementwicklung [GLL12]. Mit dem Mechatronic Modeller können die Partialmodelle Umfeld, Anwendungsszenario, Anforderung, Funktionen, Wirkstruktur und Verhalten spezifiziert werden. Das Partialmodell Gestalt wurde bei der Softwareentwicklung ausgeklammert, da hierzu bestehende 3D CAD Programme genutzt werden können.

Der grundsätzliche Aufbau der Benutzungsoberfläche ist in Bild 3-9 dargestellt. Wesentliche Bereiche sind der Projekt-Browser und der Editorbereich. Die modellierten Daten sind im Projekt-Browser in einer Baumstruktur aufgeführt. Sie sind dem jeweiligen Partialmodell zugeordnet. Die Partialmodelle lassen sich über den Modell-Browser öffnen und im Editorbereich bearbeiten. Abhängig vom Partialmodell stehen formularbasierte und graphische Editoren zur Verfügung.

Die Partialmodelle Anforderungen und Anwendungsszenario sind durch formularbasierte Editoren umgesetzt worden. Diese bestehen aus vordefinierten Textfeldern, die beschrieben werden können. Drei weitere Editoren wurden im Tool umgesetzt, die das Arbeiten und Auswerten der Modellinformationen ermöglichen. Dazu gehören Bibliotheken, Klassifikationsmerkmale und Filter (Queries). Die Bibliotheken enthalten physikalische Größen (z.B. Strom, Spannung) sowie vordefinierte Einheiten und Dimensionen (z.B. mV, V, kV). In den Bibliotheken können diese Größen verwaltet und weitere definiert werden (z.B. logische Größen). Klassifikationsmerkmale können vom Benutzer festgelegt und in jedem Partialmodell den Modellelementen zugeordnet werden. Über die Filterfunktion lassen sich Modellelemente hervorheben, die bestimmte Eigenschaften haben (z.B. Systemelemente, die ein bestimmtes Klassifikationsmerkmal hinterlegt haben).

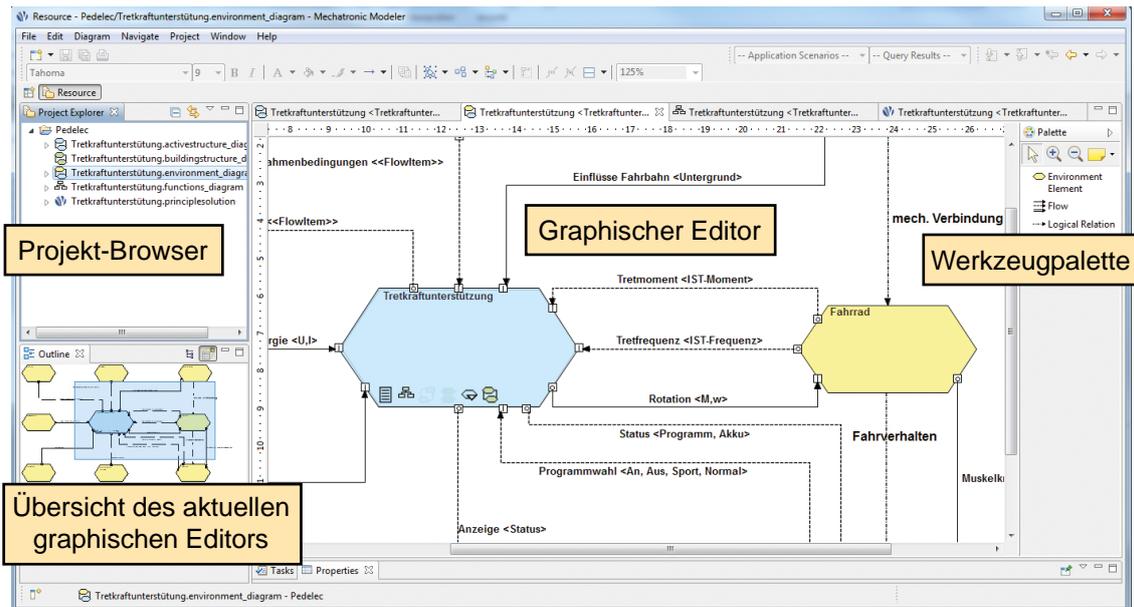


Bild 3-9: Graphische Benutzungsoberfläche des Mechatronic Modellers

Die graphischen Editoren verfügen seitlich vom Editorbereich über eine Werkzeugleiste, die für das Partialmodell notwendigen Konstrukte enthält. Der Benutzer wird damit unterstützt, die richtigen Konstrukte im jeweiligen Partialmodell zu verwenden. Weiterhin wird der Benutzer bei der Modellierung unterstützt, indem einige Operationen im Hintergrund automatisiert erfolgen. Dabei handelt es sich um Operationen, wie das Anlegen von Schnittstellen (Ports) beim Verbinden von zwei Systemelementen mit einem Fluss. Die Querbeziehungen zwischen zwei Modellelementen (z.B. Funktion und Systemelement) wird über Verlinkung realisiert. Die enge Verbindung zwischen Umfeld und der Wirkstruktur wurde ebenfalls umgesetzt. Hierzu kann der Benutzer im Umfeldmodell in das System „eintauchen“ und gelangt in die nächste Hierarchieebene – in die Wirkstruktur. Alle Flussbeziehungen am System werden auf der nächsten Hierarchieebene angezeigt und können damit weiter verbunden werden.

Bewertung: Der Mechatronic Modeller wurde dediziert für die Spezifikationstechnik CONSENS umgesetzt. Die Methode gibt dabei vor, welche Modellinhalte erstellt werden. Bei der Modellierung wird der Benutzer teilweise durch Zusatzfunktionen unterstützt (Layout der Funktionshierarchie oder automatische Portgenerierung). Darüber hinaus sind Funktionen implementiert, die den Benutzer durch die verknüpfte Modellinhalte leitet (Aktiver Link von z.B. einer Funktion zum Systemelement). Eine Auswertung der Wirkstruktur auf Vergleichbarkeit, Vollständigkeit oder Richtigkeit ist nicht vorhanden.

3.3.3 Enterprise Architect – Systems Engineering Edition

Enterprise Architect (EA) ist ein kommerzielles Modellierungswerkzeug des australischen Softwareherstellers Sparx Systems. Es handelt sich dabei um ein UML-

Modellierungswerkzeug mit einem SysML Plug-in (MDG Technology für SysML in der Systems Engineering Edition). Durch die günstige Lizenz ist das Werkzeug weit verbreitet und findet im Bereich des Model-Based Systems Engineering erste Einsätze [Alt12b]. Die Implementierung der SysML 1.3 ist in der Version EA 10 erfolgt.

Bild 3-10 zeigt den wesentlichen Aufbau der Benutzungsoberfläche. Durch die klare Trennung des Modells (entspricht hier dem Repository) und Diagramm (vgl. Kap. 3.1.4) existieren zwei wesentliche Bereiche: Projekt-Browser und das Diagramm mit der Werkzeugpalette. Im Projekt-Browser wird das Modell hierarchisch in einem Baumdiagramm angezeigt. Der Benutzer kann die Hierarchie durch Erstellen von Packages beeinflussen und damit eine eigene Ordnung vorgeben. Die Darstellung der Elemente erfolgt in den Diagrammen. Zu jedem Diagramm sind die entsprechenden Werkzeugpaletten vorhanden. Innerhalb des Diagramms können Modellelemente gelöscht, ohne dass diese aus dem Modell entfernt werden. Wird hingegen ein Element aus dem Modell gelöscht (im Projekt-Browser) ist dieses Element vollständig entfernt und wird auch aus allen Diagrammen gelöscht, in denen es dargestellt ist.

Die Verknüpfung von Modellelementen, die im Modell enthalten sind, aber nicht notwendigerweise im Diagramm abgebildet wurden, können im EA über eine Beziehungsmatrix dargestellt und bearbeitet werden.

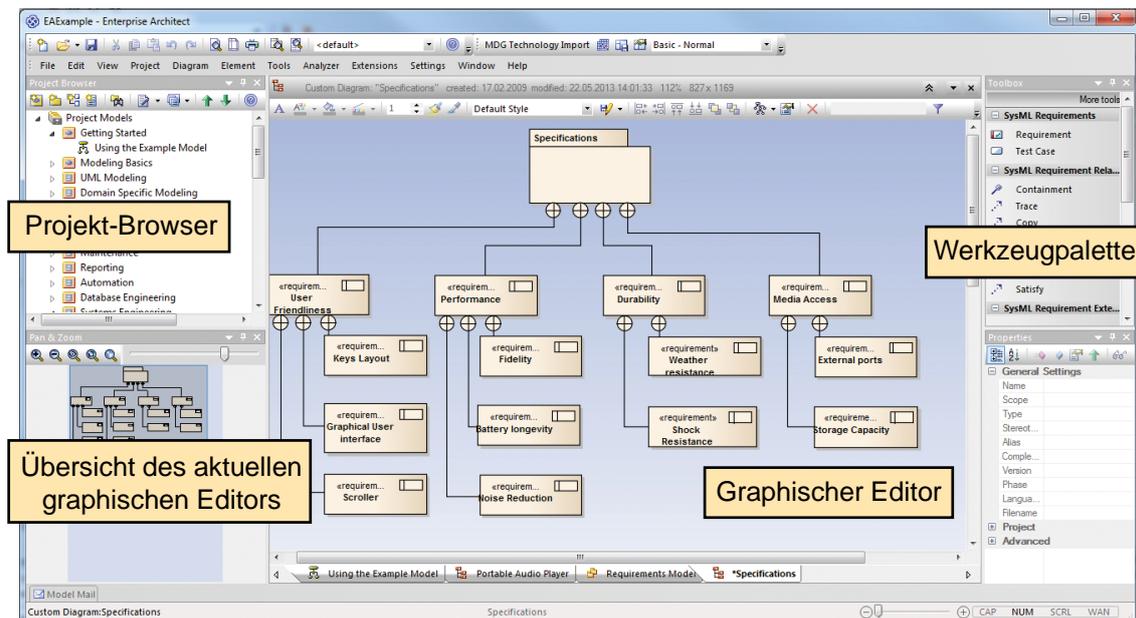


Bild 3-10: Graphische Benutzungsoberfläche des Enterprise Architect

Bewertung: SysML-Werkzeuge sind sehr umfangreich und für viele Anwendungen von UML zur Softwareentwicklung über Prozessmodellierung bis Systemmodellierung anwendbar. Der Benutzer muss sich mit dem Werkzeug sehr intensiv auseinandersetzen, um abschätzen zu können, welche Bestandteile des Werkzeugs für den Einsatz der Systemmodellierung notwendig sind [AHJ+09]. Die Modellierung ist nicht intuitiv und eine Benutzerführung wird nicht unterstützt. Die graphische Darstellung der Modelle dient

zur Erstellung des Repository (Datenmodells). Dies wird vor allem dann deutlich, wenn Modelle von einem SysML-Werkzeug in ein anderes übertragen werden. Das Repository wird dabei vollständig übergeben, das Layout der Diagramme jedoch nicht. Analysen sind grundsätzlich möglich, da die Daten formalisiert erstellt werden. Jedoch können eigene Prüfbedingungen im EA nicht implementiert werden.

3.3.4 CATIA V6 Systems

CATIA V6 Systems ist ein Softwareprodukt von Dassault Systèmes. Durch die Integration der PLM-Lösung ENOVIA V6 bietet das Werkzeug eine Umsetzung des Systems Engineering Ansatzes. Basis für das Vorgehen mit CATIA V6 ist das RFLP-Modell mit den Ursprüngen in der VDI-Richtlinie 2206 (vgl. Kapitel 2.3). Damit besteht das Systemmodell aus dem Anforderungsmodell (**R**equirements), dem Funktionsmodell (**F**unctional), dem Logischen Modell (**L**ogical) und dem 3D- Modell (**P**hysical) [Kle13].

In CATIA V6 Systems findet die Navigation durch die Ebenen des RFLP-Modells im sogenannten Silverlayer statt (Bild 3-11). Die Modellierung innerhalb der Ebenen wird über Editoren gestartet. Die Anforderungen werden dabei in enger Synchronisation mit ENOVIA V6 erstellt. Ein Im- und Export der Anforderungen in ein anderes Format, wie beispielsweise Microsoft Word, ist vorhanden.

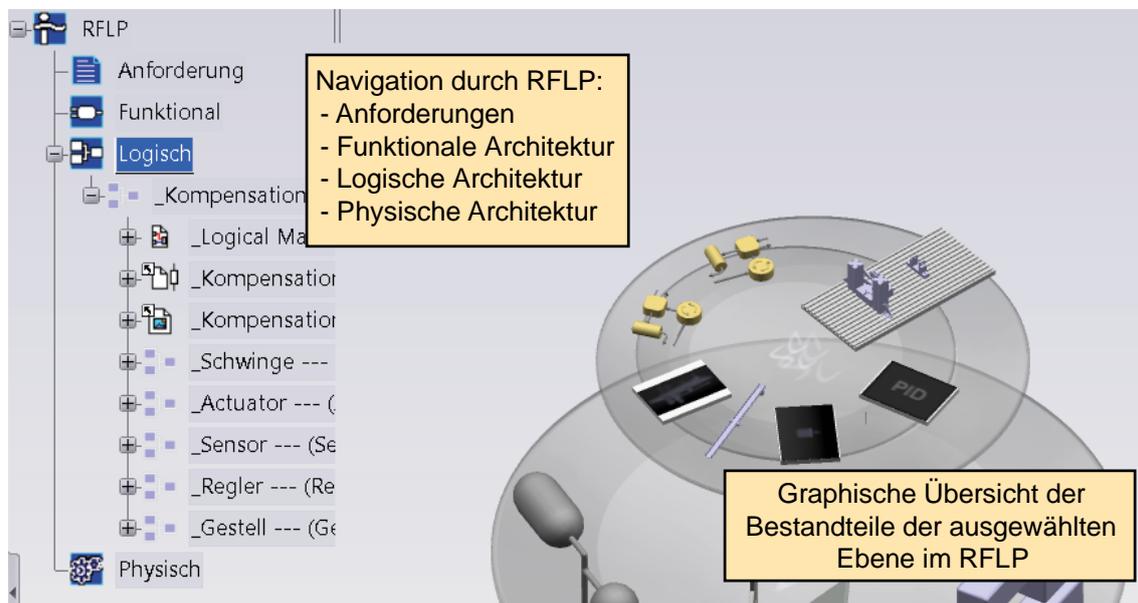


Bild 3-11: Silverlayer in CATIA zur Navigation durch die Ebenen des RFLP-Modells

In einem eigenen Editor (Workbench Funcional Logical Editor) können die beiden Modelle Functional und Logical erstellt werden. Die funktionalen Sicht beinhaltet eine Funktionsstruktur. Hierzu besitzen die funktionalen Elemente Ports, die über Konnektoren miteinander verbunden werden können. Die logische Struktur entspricht einer Systemarchitektur. Durch die Integration von Dymola in CATIA V6 Systems wurde eine Möglichkeit geschaffen, die logischen Modelle zu simulieren.

Die Ursprünge von CATIA liegen in der 3D-Modellierung. Dieser wesentliche Aspekt ist weiterhin Bestandteil und wird im letzten Schritt des RFLP-Ansatzes dargestellt. Durch die Integration der einzelnen Modelle in einem Werkzeug können die Modellelemente miteinander verknüpft werden – Anforderungen mit Funktionen der Funktionsstruktur sowie den logischen oder physischen Elementen. Eine Analyse der Anforderungen sowie eine Rückverfolgbarkeit werden damit gewährleistet.

Bewertung: CATIA V6 Systems stellt eine Plattform bereit, in der die Aspekte Anforderungen, Funktionen, logische Struktur und 3D Modell verknüpft werden können. Die Struktur kann sowohl im Editor Functional als auch Logical dargestellt werden. Die Modellinhalte werden formalisiert erstellt und können ausgewertet werden. Das Werkzeug gibt keine Bedingungen für die Darstellung der Strukturen vor. Damit ist auch keine Analyse implementiert, die diese Bedingungen prüft.

3.3.5 Mechatronic Concept Designer

Der Mechatronic Concept Designer (MCD) stellt die Softwarelösung für Systems Engineering von Siemens PLM Software dar. Der Fokus liegt auf dem Maschinen- und Anlagenbau. Das Systemmodell besteht aus den Anforderungen, Funktionen, dem 3D-Gestaltmodell sowie der Möglichkeit, Bewegungsabläufe zu simulieren. Hierzu wurde eine Physik-Engine von Nvidia implementiert, die bei Computerspielen eingesetzt wird.

Im Graphikbereich des MCD ist das 3D Modell des Systems zu sehen (Bild 3-12). Dieses kann in der Sicht bearbeitet oder mit einem Bewegungsverhalten hinterlegt werden. Der Ablauf kann dann simuliert werden, wobei ein Editor die einzelnen Bewegungssequenzen visualisiert. Alle weiteren Angaben, wie Anforderungen, Funktionen oder Eigenschaften der Elemente werden im Baumdiagramm bearbeitet. Hierzu sind unterschiedliche Navigatoren vorhanden: z.B. Funktions-, Physik-, Baugruppen-Navigator.

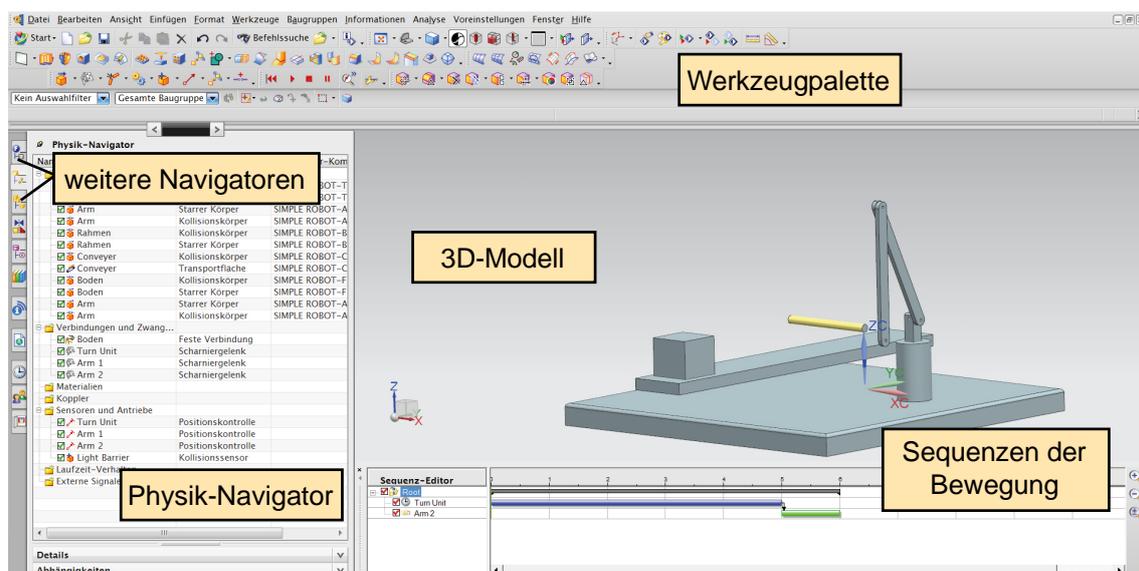


Bild 3-12: Übersicht der Benutzungsoberfläche Mechatronic Concept Designer

Bewertung: Der Mechatronic Concept Designer orientiert sich stark am 3D-Modell, das die einzige Strukturbeschreibung im Werkzeug darstellt. Das Verhalten beruht auf der implementierten Physik-Engine. Damit kann das Verhalten im ersten Entwurf simuliert werden. Die Elemente sind in einer Objektbibliothek enthalten und können einfach wiederverwendet werden. Durch die offenen Schnittstellen können Daten aus anderen Werkzeugen importiert werden. Der Fokus der Software liegt dabei eindeutig auf dem Anlagen- und Maschinenbau. Damit wird die Bandbreite mechatronischer Systeme nicht genügend berücksichtigt.

3.4 Modellierungsrichtlinien

Die SysML ist in der Anwendung und in der Forschung weit verbreitet. Da es sich rein um eine Modellierungssprache handelt und die Möglichkeit besteht, eigene Profile zu definieren, sind viele Einsatzgebiete vorhanden. Funktionale Modellierung [EGG+12], Modellierung zur Kopplung von Modelica und dem Systemmodell [RPC+12], Erstellung von Bibliotheken [LGS+11], embedded Systems [PHA+12] oder Anpassungen an Methoden (z.B. Contact & Channal Methode in [AZ11], SYSMOD in [Wei06] oder CONSENS in [KDH+13]).

Dieser breite Anwendungsbereich unterstreicht die Flexibilität der Sprache, aber auch wie unterschiedlich die Strukturen dargestellt werden können. Über leichtgewichtige Anpassungen (Profil) kann jeder Fall individuell angepasst dargestellt werden.

Daher werden vereinzelt Richtlinien aufgestellt, die bei der Modellierung mit SysML unterstützen [Alt12b], [KWH+11-ol]. Diese beruhen teils auf der Erfahrung der Autoren oder werden aus bestehenden Methoden generiert.

3.4.1 Kochbuch für MBSE mit SysML

Das sogenannte Kochbuch für MBSE mit SysML ist aus einer MBSE-Initiative der INCOSE entstanden [KWH+11-ol]. Es enthält Richtlinien, Hinweise, Modellbeispiele und Best Practices. An einem realen komplexen System, einem Teleskop (Active Phasing Experiment – APE), wurde innerhalb einer Gruppe die Modellierungssprache SysML angewandt. Ziel waren Lösungen für die Herausforderungen des MBSE-Einsatzes. SysML wird dabei als reine Sprache eingesetzt, die zum Anwendungszweck von dem Team angepasst wurde (SysML-Profil).

Das Kochbuch zeigt im Bereich der Strukturmodellierung Richtlinien und Hinweise auf. Diese geben keine konkreten Vorgaben, sondern führen auf, welche Unterschiede möglich sind und in welchem Fall eine Darstellungsform besser geeignet ist. Beispielsweise wird darauf hingewiesen, dass es verschiedene Möglichkeiten der Hierarchisierung gibt. Diese kann funktional in gruppierte Subsysteme erfolgen oder konkrete Elemente mit realen Schnittstellen besitzen [KWH+11-ol]. Am Beispiel eines Kabels wird aufgezeigt,

dass mit SysML ein und derselbe Sachverhalt auf mehrere Weisen dargestellt werden kann [KWH+11-ol].

Zusätzlich werden Views definiert, die unterschiedliche Sichten auf die Struktur vorgeben. Diese sind: *mechanical*, *optical*, *electrical*, *information*. Da zur Modellierung SysML genutzt wird und diese zwischen Diagramm und Repository trennt, kann jeweils in einem Diagramm (ibd) eine Sicht (view) graphisch erstellt werden [KWH+11-ol].

Bewertung: Das Kochbuch beschreibt viele Richtlinien, die aus Erfahrung bei der Modellierung mit SysML entstanden sind. Die Richtlinien geben dabei Hinweise, wie ein Sachverhalt dargestellt werden kann. Dabei wird deutlich, dass derselbe Sachverhalt auf unterschiedliche Weise beschrieben werden kann. Die SysML und auch die Richtlinien geben hierzu keine konkreten Vorgaben. Das gibt dem Modellierer die Freiheit, alles abbilden zu können. Der Nachteil ist jedoch, dass keine Prüfbedingungen abgeleitet werden können. Das Kochbuch gibt einen Vorschlag zum Umgang mit verschiedenen Sichten auf die Struktur. Dabei wird jede Sicht in einem Diagramm erstellt. Jedoch wird nicht aufgezeigt, wie eine graphische Sicht entstehen kann, die die einzeln aufgebauten Diagramme vereint oder nur Teile daraus visualisiert.

3.4.2 Richtlinien nach ALT

ALT stellt Modellierungsregeln auf, die die formale Korrektheit adressieren. Die aufgestellten Regeln können mit Hilfe von Werkzeugen auf ihre Einhaltung überprüft werden. Die aufgestellten Regeln beziehen sich auf die Namenskonventionen für Modellelemente, Architekturkomponenten, Architekturschnittstellen, Verknüpfungen und Modellstruktur. Dabei wird bspw. die Annahme getroffen, dass jedes Architekturelement sowie jede Schnittstelle mindestens eine Anforderung erfüllen muss. Daraus abgeleitet muss das Element/Schnittstelle mit mindestens einer Anforderung über die Beziehung «satisfy» verknüpft sein [Alt12b].

Bewertung: Die Richtlinien nach ALT geben Vorgaben zu formal korrekten Modellen. Diese sollten in einem guten Modellierungswerkzeug implementiert sein oder die Möglichkeit vorhanden sein, dies nachträglich zu implementieren. Zum Beispiel sollte ein Werkzeug Richtlinien, wie das Verbinden von nur typkompatiblen Ports [Alt12b] beinhalten. So kann der Anwender bei der Modellierung unterstützt werden, indem nur formal richtige Modelle erstellt werden können.

3.5 Funktionsmodellierung

Die Funktionsmodellierung ist in der Systemtechnik ein wichtiger Schritt des Problemlösungsprozesses. Die Forderung nach dem Denken in Funktionen soll eine lösungsneutrale Sicht auf das Problem ermöglichen und damit ein voreiliges Festlegen von Lösungsvarianten vermeiden [Pat82]. Hieraus wurden in der Produktentwicklung unterschiedliche Konzepte der funktionsorientierten Systembeschreibung entwickelt,

die sich hauptsächlich auf die Konstruktion beziehen. Dabei lassen sich zwei wesentliche Unterschiede in den Konzepten feststellen: Festlegen auf wenige Funktionen, die die Grundfunktionen darstellen und Erstellen von Funktionskatalogen.

Grundfunktionen

Grundfunktionen werden auf Basis der drei Wirkbeziehungen zwischen Funktionen definiert: Information, Energie und Materie/Stoff. Es werden im Wesentlichen fünf Funktionsverben genannt: wandeln, ändern, verknüpfen, leiten und speichern [PBF+07]. KOLLER/KASTRUP stellt hierzu die Annahme auf, dass die Vorgänge in komplexen technischen Systemen sich aus einer Anzahl an elementaren Tätigkeiten zusammensetzen lassen. Diese sogenannten Grundoperationen werden für die Energieumsätze, Stoffumsätze und Daten-, bzw. Informationsumsätze definiert. Zu den Grundfunktionen werden Prinziplösungen in einem Katalog bereitgestellt, die auf physikalische/chemische Effekten beruhen [KK94].

Funktionskataloge

BIRKHOFER erstellte ein Katalog mit 222 Funktionsverben, die technische Funktionen beschreiben. Die Basis bildete ein technisches Wörterbuch [Bir80]. Eine Erweiterung dieser Funktionsverben ist in ROTH und LANGLOTZ zu finden. ROTH nutzt diese Verben zusätzlich zur Unterstützung der Lösungssuche mit Hilfe eines Konstruktionskatalogs [Rot00]. Dieser enthält zu den Funktionen bewährte Lösungsprinzipien. LANGLOTZ ergänzt diesen Ansatz, indem durch eine Formalisierung der Funktionsbeschreibung rechnerinterne Wissensspeicher genutzt werden können [Lan99]. Eine ähnliche Vorgehensweise nutzt SHEA et al., indem sie Funktionen formal mit der Modellierungssprache SysML abbildet und in einer Bibliothek speichert [LGS+11]. Dabei wurde die NIST¹²-Funktionsdatenbank implementiert. Diese beinhaltet eine einheitliche Beschreibung von Basisfunktionen [HMW02]. DUMITRESCU erweitert die bestehenden Funktionskataloge um Funktionen der Informationsverarbeitung. Er definiert eine einheitliche Spezifikation von wiederverwendbaren Lösungsmustern, in denen die Funktionen einen Bestandteil bilden. Dies ermöglicht die Suche nach Lösungsmustern über Funktionen [Dum11].

Bewertung: Die funktionale Beschreibung eines Systems wird in allen Ansätzen zur Lösungssuche genutzt. Dabei werden Funktionskataloge verwendet und mit Lösungsprinzipien oder -mustern verknüpft. Die Modellierung von Funktionen hat den Vorteil, lösungsneutral zu arbeiten. Dies ist sinnvoll, wenn neue Konzepte erarbeitet werden müssen. In vielen Unternehmen wird jedoch auf bestehenden Produkten aufgesetzt. So zeigt sich auch, dass Entwickler nur schwer in Funktionen denken können. Die Ansätze zur Formalisierung von Funktionen zeigen auf, dass dies ein wichtiger Schritt zur Analyse und einheitlichen Beschreibung von Modellinhalten darstellt (in diesem Fall den Funktionsstrukturen).

¹² National Institute of Standards and Technology (NIST), eine Bundesbehörde der USA

3.6 Zusammenfassende Betrachtung der Strukturmodellierung

Die Analyse des Stands der Technik zeigt auf, dass eine Menge an Ansätzen des Model-Based Systems Engineering vorhanden ist. Alle beinhalten auch eine Systemstruktur, die einen wesentlichen Bestandteil der Ansätze ausmacht. Hier zeigen sich aber auch die Unterschiede. Während bei Begrifflichkeiten wie Anforderungen und Verhalten Einigkeit herrscht, wird der Struktur-, bzw. Architekturbegriff vielfältig genutzt. Es lassen sich vier Strukturbegriffe identifizieren: funktional, logisch, physisch und wirkungsorientiert (Wirkstruktur). Deren wesentliche Merkmale und damit auch deren Unterschiede werden hier zusammengefasst beschrieben.

Funktionale Architektur

Die funktionale Architektur wird auch Funktionsstruktur, Funktionshierarchie oder Nutzungsebene genannt. Diese wird in der Funktionsmodellierung hauptsächlich adressiert. Ziel ist eine funktionale Sicht auf das System. Hierzu werden die Funktionen lösungsneutral [PBF+07] – in einigen Fällen bevorzugt aus der Sicht des Anwenders [PHA+12] – betrachtet. Eine rein hierarchische Unterteilung von Bestandteilen stellt nach [Pat82] ebenfalls eine Struktur dar. So finden sich Funktionshierarchien, die unter dem Begriff Funktionsstruktur geführt werden (z.B. bei METUS). Die funktionale Architektur ist Bestandteil in METUS, CONSENS, FAS, CATIA V6 Systems und Mechatronic Concept Designer.

Logische Architektur

Während in der funktionalen Architektur die Funktionen beschrieben und miteinander in Beziehung gesetzt werden, findet hier die Betrachtung auf Ebene der Systemelemente statt. Die Verknüpfung von logischen Elementen führt zu einer logischen Architektur. Logische Elemente erfüllen eine oder mehrere Funktionen, lassen jedoch offen, wie die Realisierung aussehen wird. Diese Elemente und die Verbindung der Elemente finden sich in den Ansätzen wieder, die softwareintensive Systeme betrachten. Dazu gehören SYSMOD und OOSEM. Auch die funktionale Architektur nach ALT wird in diese Kategorie eingeordnet. In CATIA V6 Systems ist eine logische Sicht vorgesehen. Diese ist mit Simulationsmodellen in Dymola gekoppelt. Teilweise wird jedoch auch die funktionale Sicht in CATIA für logische Elemente genutzt. Hier werden vom Werkzeug keine Vorgaben gemacht. An diesen Beispielen zeigt sich, wie gleiche Begrifflichkeiten (funktionale Sicht) für unterschiedliche Sachverhalte (Beschreibung von Funktionen oder Elementen) verwendet werden.

Physische Architektur

Die Realisierung der Elemente wird in der physischen Architektur spezifiziert. Diese enthält reale, physische Elemente und Schnittstellen. Nach BROY et al. wird diese Ebene auch als technische Architektur bezeichnet. Sie

„dient als „Zielmodell“ für die modellbasierte Entwicklung von Software für eingebettete Systeme, d.h. sie stellt die Ebene mit dem niedrigsten Abstraktionsgrad im Architekturmodell dar.“ [BFG+08]

Diese Art der Strukturbeschreibung findet sich ebenfalls in den Entwicklungsansätzen der softwareintensiven Systeme wieder: SYSMOD und OOSEM. ALT unterscheidet hierbei noch einmal zwischen der technisch-physikalischen und technischen Wirkketten-Architektur. In CATIA V6 Systems und dem Mechatronic Concept Designer wird unter der physischen Architektur die Gestalt der Systeme verstanden. Die Darstellung erfolgt in einem 3D-Modell.

Wirkstruktur

Die Wirkstruktur aus CONSENS lässt sich nicht einfach in die oberen Architekturen einordnen und wird gesondert betrachtet. Es scheint ein Mittelweg zwischen der logischen und der physischen Architektur zu sein. Die Elemente können sowohl abstrakt beschrieben werden und entsprechen damit den logischen Elementen, als auch sehr konkret in Form von Lösungselementen¹³. Damit beinhalten sie auch die Realisierung des Elements. Bei den Wechselwirkungen bleibt es, bedingt durch die Unterscheidung in die Arten Energie, Stoff, Information und logische Beziehung, auf der Ebene der logischen Architektur. Eine Analyse bestehender Wirkstrukturen zeigt, dass hier sowohl funktionale Wirkbeziehungen als auch realisierte Schnittstellen modelliert werden. Es findet eine Vermischung der Konzepte statt, so dass Modelle auf vielfältige Weise entstehen und damit nicht vergleichbar sind. Ein klassisches Beispiel hierfür ist die Darstellung von Druckluft. Diese ist ein Medium zur Übertragung von Energie, so dass es einer Energiebeziehung entspricht. Dennoch ist häufig in der Wirkstruktur ein Stofffluss vorzufinden. Hingegen wird die Übertragung von Signalen als ein Informationsfluss beschrieben. Analog zu dem Beispiel Druckluft müsste hier die elektrische Verbindung dargestellt werden, die zur Übertragung der Informationen benötigt wird. Es fehlt an klaren Vorgaben zur Beschreibung der Elementen und Beziehungen.

¹³ Lösungselemente stehen für konkrete Umsetzungen eines Systemelements. Sie können beispielsweise Norm-, Zukaufteilen oder wiederverwendeten Elementen entsprechen. Die Spezifikation der Elemente liegt vollständig vor.

3.7 Handlungsbedarf

Im Stand der Technik (Kapitel 3.1 bis 3.5) wurde jeder vorgestellte Ansatz bewertet. Die Übersicht der Bewertungen ist in Bild 3-13 aufgezeigt. Zusammenfassend lassen sich die Anforderungen wie folgt bewerten.

A1) Sprachenunabhängig: Die Methoden sind weitestgehend sprachenunabhängig definiert. Die Anwendung der Methoden mit anderen Sprachen, als zur Methodenbeschreibung verwendet (z.B. FAS mit SysML), kann dazu führen, dass nicht alle Informationen abgebildet werden können. Daher wird eine Trennung der Methode von der verwendeten Sprache nicht empfohlen. Die Werkzeuge sind nicht sprachenunabhängig. Lediglich die SysML-Werkzeuge können als sprachenunabhängig angesehen werden, da es sich um UML-Werkzeuge handelt und durch Profile weitere Sprachen implementiert werden können.

A2) Systemstruktur als Basis für ein einheitliches Systemverständnis: Die beschriebenen Ansätze können eine Systemstruktur abbilden. Dabei reichen die Elemente und Beziehungen bei METUS und OPM nicht aus, um die Wirkungsweise des Systems in der Systemstruktur zu beschreiben. Dies wird bei METUS nur indirekt durch die Verbindung von Komponenten und Funktionen erreicht. Nicht alle Ansätze sind für die Strukturbeschreibung von mechatronischen Systemen geeignet. Eine explizite Ausrichtung auf mechatronische Systeme durch z.B. Einschränkung der abzubildenden Elemente, findet sich in keinem der Ansätze.

A3) Vergleichbarkeit: Die Definitionen der Systemstrukturen werden von den Methoden vorgegeben. Dies geschieht jedoch auf einem abstrakten Niveau ohne Bedingungen an die darzustellenden Elemente und Beziehungen. Damit entstehen Modelle die abhängig vom Modellierer sind und seinen Erfahrungen in der Systemmodellierung. Eine Vergleichbarkeit zwischen den Modellen ist nicht gegeben. Erste Ansätze finden sich bei GEHRKE für die Modellierung mit CONSENS. Auch ALT gibt eine grobe Struktur durch das EVA-Prinzip vor.

A4) Vollständigkeit: Durch die Orientierung des Ansatzes nach ALT auf das EVA-Prinzip können Rückschlüsse auf Vollständigkeit gemacht werden. Die Richtlinien nach ALT hingegen beziehen sich nur auf die Vollständigkeit hinsichtlich der Benennung von Elementen. Da keines der Ansätze sich explizit auf mechatronische Systeme fokussiert, finden sich darüber hinaus keine Richtlinien, die sich auf die Vollständigkeit einer Systemstruktur beziehen.

A5) Richtigkeit: Die formale Richtigkeit wird in den Ansätzen teilweise adressiert. Dies erfolgt über die Definition von Richtlinien (z.B. nach ALT) oder durch Bedingungen implementiert im Werkzeug (z.B. Mechatronic Modeller). Konkrete Bedingungen an die Beschreibung der Elemente und Beziehungen werden dabei nicht getroffen. Die semantische Richtigkeit wird von keinem der Ansätze verfolgt.

A6) Klarheit: Die Handhabung der graphischen Modelle wird durch vereinzelte Konzepte (z.B. Filter oder *views*) verfolgt. Der Mechatronic Modeller bietet die Möglichkeit, Filter zu definieren. Die gefilterten Elemente werden in den Diagrammen dann hervorgehoben dargestellt. Eine Vorgabe über mögliche Filter wird nicht mitgeliefert. Das Kochbuch für MBSE zeigt durch den Einsatz von *views* auf, wie mit SysML die Diagramme spezifische Sichten enthalten (z.B. optische, mechanische). Dies lässt sich mit SysML gut realisieren, da hier zwischen Sicht (Diagramm) und dem Repository getrennt wird.

A7) Vereinbarkeit von intuitiver Anwendung und formalisierter Beschreibung: Die untersuchten Ansätze fokussieren entweder die formalisierte oder die wenig formale dafür aber intuitive Anwendung. Keines der Ansätze schafft eine Vereinbarkeit der beiden Gegensätze.

A8) Rechnerunterstützte Modellierung: Für jede Methode gibt es eine softwaretechnische Unterstützung. Diese sind im Fall von CONSENS und METUS auf die Methode und Sprache dediziert ausgerichtet. In METUS sind zudem Funktionen implementiert, die das Modellieren erleichtern (Drag-and-Drop, Layoutgestaltung, etc). Die untersuchten Werkzeuge enthalten wenig bis keine Funktionen, die den Anwender bei der Erstellung und Pflege des Modells unterstützen. Es fehlt bei den Werkzeugen darüber hinaus an automatischen Überprüfungen der erstellten Strukturen hinsichtlich Vergleichbarkeit oder Vollständigkeit. Syntaktische Richtigkeit wird teilweise durch das implementierte Metamodell abgefangen.

Bewertung: Der Ansatz hat die Anforderung...  voll erfüllt  teilweise erfüllt  nicht erfüllt	Anforderungen							
	A1	A2	A3	A4	A5	A6	A7	A8
Modellierungssprache								
METUS - Sprache								
CONSENS - Sprache								
OPM								
SysML								
Modellierungsmethode								
METUS - Methode								
CONSENS - Methode								
FAS								
SYSMOD								
OOSEM								
Funktionale und technische Entwicklung nach ALT								
Modellierungswerkzeuge								
METUS - Software								
Mechatronic Modeller								
Enterprise Architect								
CATIA V6 Systems								
Mechatronic Concept Designer								
Modellierungsrichtlinien								
Kochbuch für MBSE mit SysML								
Richtlinien nach ALT								
Funktionsmodellierung								
Grundfunktionen und Funktionskataloge								

Bild 3-13: Bewertung des untersuchten Stands der Technik anhand der Anforderungen

Keine der untersuchten Ansätze und auch keine triviale Kombination der Ansätze erfüllen die Anforderungen ganzheitlich. Wie in Kapitel 3.6 aufgezeigt, liegt dies unter anderem am unterschiedlichen Verständnis einer Systemstruktur/Architektur. Dies äußert sich auch in den Softwarewerkzeugen. Es besteht ein Handlungsbedarf für ein *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*.

4 Rahmenwerk zur Modellierung einer plausiblen Systemstruktur

Kapitel 4 bildet den Kern der vorliegenden Arbeit und beschreibt die Bestandteile des *Rahmenwerks zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*. In Kapitel 3 wurden die gängigen Systemstrukturen untersucht. Es zeigte sich, dass diese in vier Strukturklassen unterteilt werden können: funktionale, logische und physische Architektur sowie die Wirkstruktur. Die in dieser Arbeit adressierte Systemstruktur orientiert sich an der Wirkstruktur aus CONSENS. Dabei werden die Beziehungen zwischen den Elementen, anders als bisher in der Wirkstruktur, auf die rein funktionalen Zusammenhänge beschränkt. Damit wird eine Vermischung der Bedeutung einer Beziehung vermieden (vgl. Kapitel 3.6). Ein Element enthält funktionale Schnittstellen, die vorgeben, durch welche Wirkbeziehung das Element mit anderen Elementen interagieren kann. Das Element selbst entspricht dabei einem Funktionsträger. Bei den meisten zu beschreibenden Elementen ist der Funktionsträger gleichzeitig das zugehörige physische Bauteil/Modul (z.B. Elektromotor). In diesem Fall kann die Realisierung des Elements bekannt sein, so dass es in Form eines Lösungselements spezifiziert wird. Andernfalls erfolgt die abstrakte Beschreibung des Elements.

Ein weiterer Unterschied wird zu der CONSENS-Wirkstruktur vorgenommen: Die **Systemstruktur** trennt die Aspekte **Wirkstruktur und Umfeld** nicht. Es handelt sich hierbei lediglich um unterschiedliche Hierarchieebenen, so dass die Beschreibung der Interaktion des Systems mit dem Umfeld als Teil der Systemstruktur verstanden wird.

Bild 4-1 zeigt die sechs Bestandteile des Rahmenwerks. Die Systemstruktur besteht aus den Modellkonstrukten, die Elemente und Beziehungen darstellen. Hierzu findet zu Beginn eine **Definition und Klassifikation der Modellkonstrukte** statt (Kapitel 4.1). Zur Modellierung vergleichbarer, vollständiger und richtiger Systemstrukturen werden in Kapitel 4.2 **Richtlinien und Bedingungen** formuliert. Im Rahmen der **Plausibilitätsprüfung** werden diese auf ihre Einhaltung geprüft (Kapitel 4.3). Zur Prüfung der Richtigkeit werden darüber hinaus Hilfsmittel in Form von Matrizen (Energieumsetzende Elemente und Sensoren) vorgestellt. Auf Basis der Element- und Beziehungsklassen können **Sichten auf die Systemstruktur** generiert werden (Kapitel 4.4). Damit wird die visuell wahrgenommene Komplexität reduziert.

Die Bestandteile aus Kapitel 4.1 bis 4.5 bilden die Grundlage für das Konzept einer **Werkzeugunterstützung** (Kapitel 4.5) sowie einem **Vorgehensmodell** (Kapitel 4.6). Im Softwarewerkzeug sind die Element- und Beziehungsklassen sowie die Bedingungen im Metamodell der Sprache implementiert. Damit werden die zugehörigen Konzepte genutzt, den Anwender bei der Modellerstellung zu unterstützen und eine automatische Sichtenbildung und Prüfung zu ermöglichen. Das Vorgehensmodell beschreibt die interdisziplinäre Vorgehensweise zur Erstellung einer plausiblen Systemstruktur.

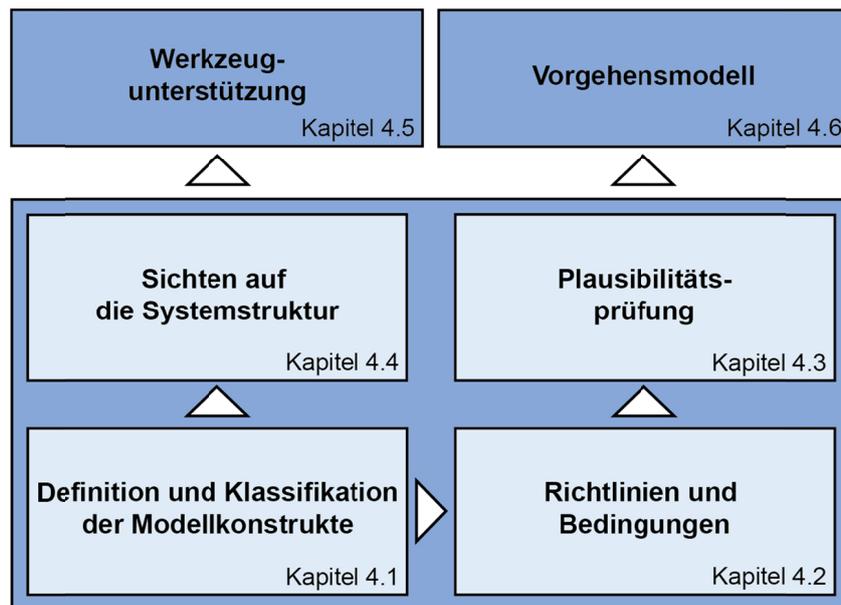


Bild 4-1: Bestandteile des Rahmenwerks

4.1 Definition und Klassifikation der Modellkonstrukte

Die modellhafte Abbildung der Systemstruktur erfolgt mit vordefinierten Modellkonstrukten. Dabei besteht eine Systemstruktur aus Elementen, die miteinander in Interaktion stehen. Diese wird durch Beziehungen zwischen den Elementen dargestellt. Bild 4-2 zeigt eine Übersicht der Modellkonstrukte. Zu Beginn werden die verschiedenen Beziehungen beschrieben (Kapitel 4.1.1). Anschließend erfolgt die Betrachtung der Elemente, unterteilt in technische Elemente (Kapitel 4.1.2) und nicht-technische Elemente (Kapitel 4.1.3). Dabei können technische Elemente innerhalb oder außerhalb der Systemgrenze liegen. Je nach Zugehörigkeit (Umfeld oder System) werden diese Elemente als Umfeldelemente oder Systemelemente spezifiziert. Dagegen entsprechen nicht-technischen Elemente den Umfeldelementen, da sie in der Regel außerhalb der Systemgrenze liegen. Die Begriffe Klasse und Typ werden im Zusammenhang mit der Unterteilung von Elementen und Beziehungen synonym verwendet.

Für die Elementklassen wurden Steckbriefe erstellt, die als Hilfsmittel zum Erlernen und Anwenden der Klassen dienen (Anhang A1). Im Steckbrief wird das Element beschrieben, Grundfunktionen der Elemente aufgeführt, eine beispielhafte Anwendung sowie häufig auftretende Fehler bei der Modellierung gezeigt.

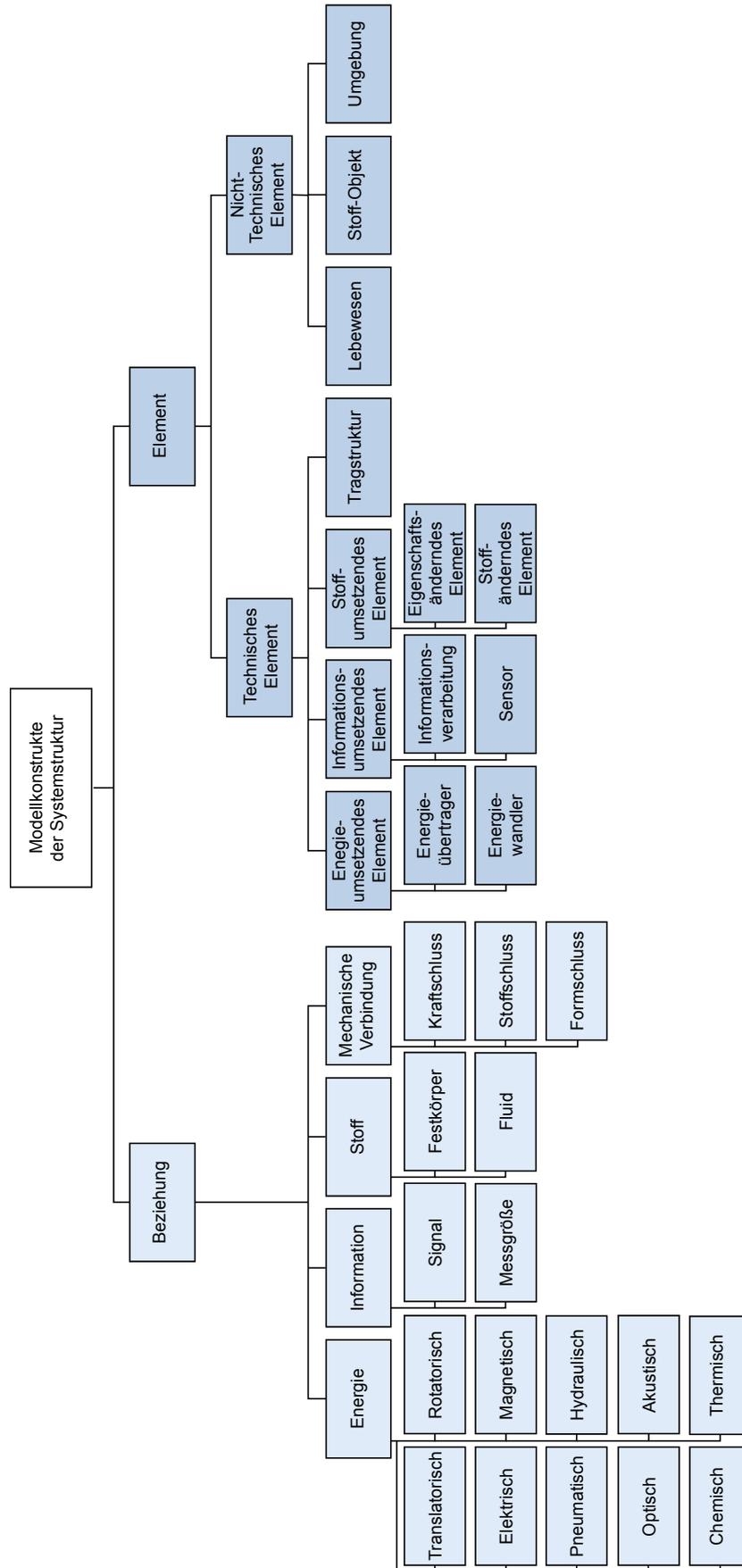


Bild 4-2: Übersicht der Modellkonstrukte

In Anlehnung an CONSENS wurde für die Darstellung der Konzepte folgende graphische Notation gewählt: das Sechseck für ein Element (blau = Element innerhalb der Systemgrenze, gelb = außerhalb der Systemgrenze), durchgezogene Linie für Energiebeziehung, gestrichelte Linie für Informationsbeziehung und doppelte Linie für Stoffbeziehung. Für die mechanische Verbindung ist eine ungerichtete durchgezogene Linie eingeführt worden. Darüber hinaus werden Ports verwendet, die die Schnittstellen der Elemente darstellen. Die Klasse wird in doppelten spitzen Klammern (Guillemets) am Element oder an der Beziehung dargestellt. Für die spätere Auswertung der Elemente, werden weitere Informationen über die Elemente benötigt. Hierzu erhalten die Elemente die Attribute (*Energie-*)*Quelle*, *aktiv* oder *gestaltbehaftet* (vgl. Kapitel 4.2.2), die in Form von Piktogrammen visualisiert werden. Bild 4-3 zeigt die verwendete graphische Notation.

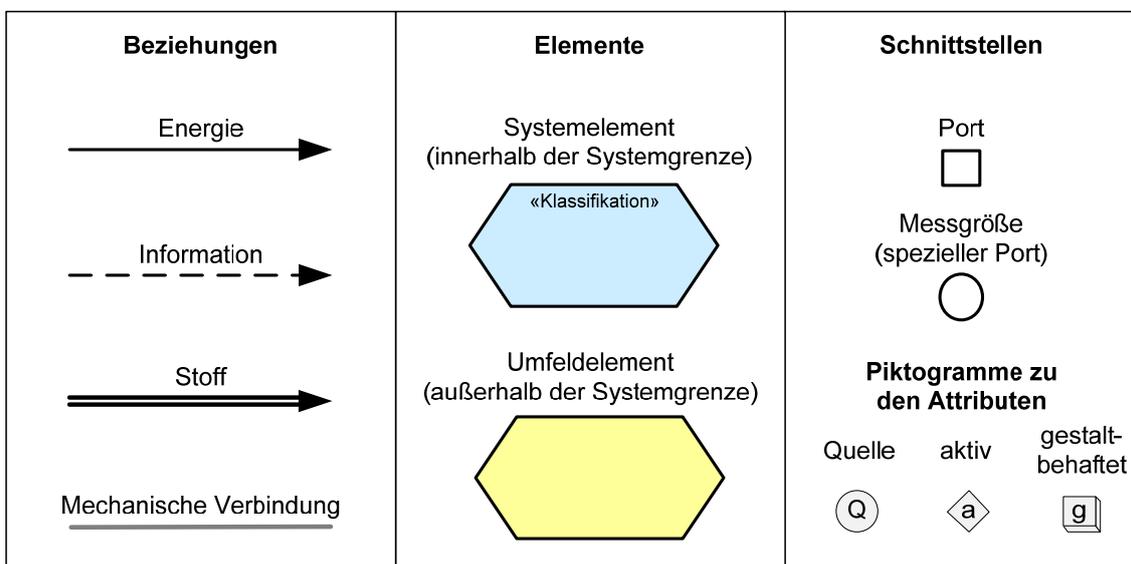


Bild 4-3: Graphische Notation der Beziehungen, Elemente und Schnittstellen

4.1.1 Beziehungen

Zur Darstellung der Beziehungen in der Systemstruktur werden die aus der Systemtheorie stammenden Beziehungsarten Energie, Stoff und Information verwendet. Diese zeigen die Wirkungsweise und sind damit gerichtete Beziehungen. Zu den drei Arten kommt eine weitere Beziehung hinzu, die in der Systemstruktur zum Einsatz kommt: mechanische Verbindung (Bild 4-4). Diese Beziehung unterscheidet sich von den drei vorher genannten dahingehend, dass es sich dabei um keine gerichtete Wirkbeziehung, sondern um eine konstruktive Beziehung handelt – eine indirekte Wirkbeziehung. Erst durch die Verbindung zweier Elemente werden Teilfunktionen realisiert, wie z.B. *leiten* oder *isolieren*. Die Wahl der mechanischen Verbindung zwischen zwei Bauteilen entscheidet über die Güte der zu erfüllenden Funktion. Zweck der Verbindung ist das Übertragen/Leiten von Kräften, Momenten oder die Aufnahme von Relativbewegungen

[PBF+07]. Die einzelnen Beziehungen werden im Folgenden weiter verfeinert. Die Konzepte nach GEHRKE werden dabei aufgegriffen und angepasst [Geh05].

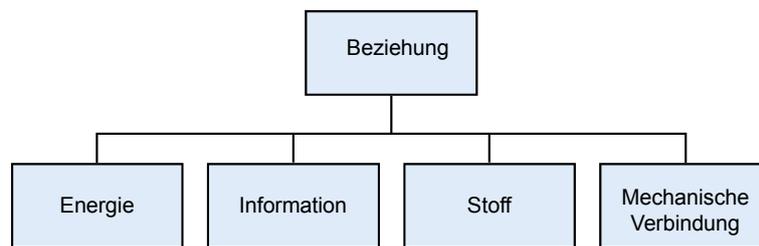


Bild 4-4: Klassifikation der Beziehung

4.1.1.1 Energie

Die Energiebeziehung wird als gerichtete Beziehung dargestellt mit der Pfeilrichtung in Wirkungsrichtung. Mit der Beziehung wird ausgedrückt, dass Energie übertragen wird. Dabei erfolgt die Energieübertragung stets über ein Medium: z.B. hydraulische Energie über Öl oder akustische Energie über Luft. Dieses wird in der Systemstruktur jedoch nicht angegeben. Es wird die Wirkbeziehung (z.B. hydraulische Energie) zwischen zwei Elementen dargestellt und nicht das Medium zur Übertragung der Energie (z.B. Öl als Stofffluss).

Energie ist eine physikalische Größe, die in unterschiedlichen Formen auftreten kann. Dabei gibt es eine begrenzte Anzahl an möglichen Energieformen, die in Bild 4-5 aufgeführt sind. Diese ergeben die Klassifikation der Beziehung *Energie*.

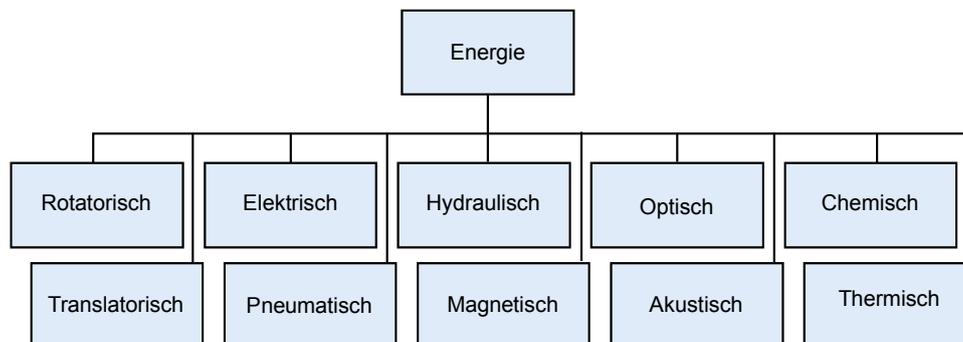


Bild 4-5: Klassifikation der Energiebeziehung

Neben den hier aufgeführten Energieformen sind noch weitere Formen aus der Physik bekannt: Kernenergie, potentielle Energie oder kinetische Energie. Diese Energieformen sind jedoch Charakteristika eines Elements und beschreiben das Element – nicht jedoch die Beziehung. Kinetische Energie ist beispielsweise eine Bewegungsenergie, die über Translation oder Rotation als Wirkung übertragen wird.

Eine Wirkbeziehung zwischen zwei Elementen wird über die aufgeführten Energieformen charakterisiert. Mit der Energieform gehen physikalische Parameter einher, die

diese charakterisieren. So kann Rotation z.B. über ein Drehmoment, Winkelgeschwindigkeit oder Drehimpuls beschrieben werden.

4.1.1.2 Information

Die Informationsbeziehung ist ebenfalls gerichtet und wird in der Systemstruktur auf zwei unterschiedliche Weisen genutzt (Bild 4-6). Mit der Beziehung wird die Kommunikation zwischen Elementen dargestellt. Dabei werden Signale, Daten oder Informationen übertragen. Wie die Übertragung umgesetzt wird (bspw. Änderung von physikalischen Größen, wie Spannung, Widerstand, Induktivität [KK94]), ist nicht Teil der Systemstruktur. In diesem Fall gilt analog zur Energiebeziehung, dass die Realisierung der Beziehung nicht dargestellt wird. Diese Beziehungsart wird zusammengefasst unter der Klasse **Signal** geführt.

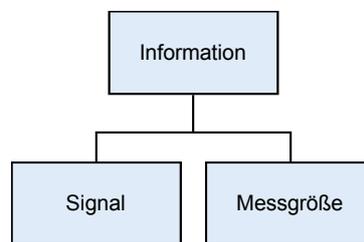


Bild 4-6: Klassifikation der Informationsbeziehung

Die Informationsbeziehung wird darüber hinaus zur Veranschaulichung einer Messung genutzt. Diese Beziehungsart wird mit **Messgröße** typisiert. Eine Messgröße ist nach der DIN 1319 eine physikalische Größe, der eine Messung gilt [DIN1319]. Die Größe kann direkt bestimmt werden oder sich aus anderen zu messenden Größen errechnen lassen. Die Beziehung vom Typ «Messgröße» endet stets an einem Sensor (vgl. Kap. 4.1.2.2). Eine weitere Besonderheit ist mit der Beziehung verbunden: Während alle anderen Beziehungen nur zwischen zwei Elementen existieren und damit an einem Element beginnen und am zweiten Element enden, kann die Messgröße zusätzlich an einer Beziehung starten. Durch die besondere Semantik sollte für die graphische Darstellung der Beziehung ein Unterschied zur Informationsbeziehung «Signal» vorliegen. Dies lässt sich jedoch nur im Fall einer eigenen Sprachdefinition umsetzen. In den meisten Fällen ist ein Einfluss auf die graphische Notation nicht gegeben.

Der Startpunkt der Beziehung sagt aus, wo bzw. an welchem Objekt die Messgröße erfasst wird. Beginnt die Beziehung am Element, wird eine Eigenschaft des Elements gemessen (z.B. Temperatur eines elektrischen Bauteils). Im Fall eines Stoffumsetzenden Elements (Kap. 4.1.2.3) kann zusätzlich die Eigenschaft des Stoffs erfasst werden, der im Element verarbeitet wird. Dies trifft auch zu, wenn der Startpunkt der Beziehung an einer anderen Beziehung liegt. Dann werden Eigenschaften der Beziehung, bzw. des Stoffs erfasst (z.B. Position, Farbe, Druck, Volumenstrom). In Bild 4-7 sind die zwei Fälle am Beispiel der Temperaturmessung aufgezeigt. Im ersten Fall wird die *Tempera-*

tur des Bauteils *E-Motor* erfasst (Bild 4-7 a)) – die Beziehung beginnt am Element. In Bild 4-7 b) wird hingegen die *Wassertemperatur* bestimmt. Das *Wasser* wird durch die Stoffbeziehung dargestellt. Die Beziehung «Messgröße» beginnt somit an der Stoffbeziehung *Wasser*.

Die Beziehung hat die Semantik, dass eine Messgröße aufgenommen werden kann bzw. wird. Nach welchem physikalischen/chemischen Prinzip dies geschieht, ist in einem zusätzlichen Modellelement «Sensor» hinterlegt.

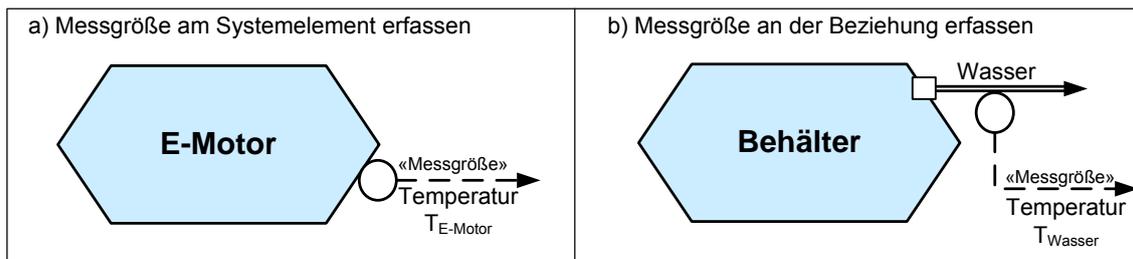


Bild 4-7: Beispielhafte Darstellung der Erfassung einer Messgröße a) am Element, b) an der Beziehung

4.1.1.3 Stoff

Die gerichtete Stoffbeziehung zeigt die Richtung des Stoffflusses an und beschreibt damit welchen Weg ein Stoff im System durchläuft. Die Beziehung ist direkt mit der Beschreibung des Stoff-Objekts verbunden (vgl. Kap. 4.1.3.2). Die Beziehung lässt sich weiter unterteilen in die Klassen **Fluid** und **Festkörper** (Bild 4-8). Gase und Flüssigkeiten werden unter **Fluide** zusammengefasst, da sie ähnliche Eigenschaften haben und mit ähnlichen Parametern beschrieben werden (z.B. Druck, Volumenstrom).

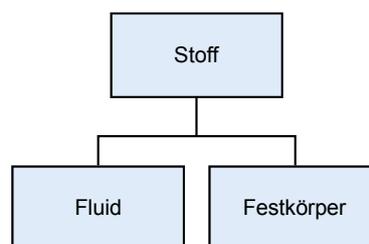


Bild 4-8: Klassifikation der Stoffbeziehung

Im System werden Stoffe vielseitig eingesetzt. Stoffe können z.B. transportiert, gemischt, getrennt, gefaltet, getrocknet, gefärbt oder galvanisiert werden. Innerhalb des Prozesses findet eine Zustandsänderung des Stoffs statt [PBF+07].

Stoffe werden im Modell als Objekte angelegt und mit einem Namen (Bezeichnung), Merkmal und Merkmalsausprägung versehen. Die Merkmalsausprägung kann eine Eigenschaft des Stoffs sein (z.B. Farbe) oder einen Zustand des Stoffs beschreiben, den es einnehmen kann (z.B. bestimmte Temperatur, Form). In Tabelle 4-1 sind Stoffe, deren Merkmale sowie Merkmalsausprägungen beispielhaft aufgeführt.

Tabelle 4-1: Beispiele für Stoffe mit Name, Merkmal und Merkmalsausprägung

	Merkmal	Merkmalsausprägung
Fluid	Druck	2 bar .. 10 bar
	Temperatur	5°C .. 80°C
	Aggregatzustand	gasförmig, flüssig
	Zusammensetzung	Frischwasser, Lauge
Festkörper	Menge	1..10
	Farbe	rot, schwarz, gelb
	Form	rund, eckig, gebogen
	Fertigungszustand	Rohstoff, Zustand 1..n, Erzeugnis

4.1.1.4 Mechanische Verbindung

Die mechanische Verbindung stellt die konstruktive Beziehung zwischen zwei Elementen dar. Erst durch die Verbindung zweier Elemente können Funktionen, wie *abdichten* oder *Drehmoment übertragen*, erfüllt werden. Die verbundenen Elemente werden dabei durch Kräfte zusammengehalten, so dass keine Wirkrichtung vorhanden ist und die Beziehung ungerichtet in der Systemstruktur dargestellt wird. In der Konstruktion werden drei Schlussarten unterschieden, die eine Verbindung charakterisieren: **Stoffschluss**, **Formschluss** und **Kraftschluss** (Bild 4-9).

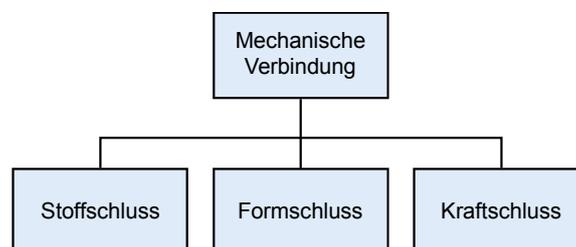


Bild 4-9: Klassifikation der mechanischen Verbindung

Beim **Stoffschluss** erfolgt eine stoffliche Vereinigung von Werkstoffen über Molekular- und Adhäsionskräfte an der Wirkfläche der Fügestelle. Dies kann auch durch das Hinzufügen eines Zusatzwerkstoffes erfolgen (z.B. Klebstoff). Allgemeine Beispiele für einen Stoffschluss sind Schweiß-, Löt- und Klebverbindungen. Eine wichtige Eigenschaft ist die Unlösbarkeit einer Stoffschlussverbindung [PBF+07].

Der **Formschluss** erfolgt durch Normalkräfte an ineinandergreifenden Wirkflächen der Verbindungselemente. An der Wirkfläche erfolgt eine Flächenpressung, wodurch Zusatzfunktionen wie Dichten, Isolieren oder Leiten erfüllt werden. Im Gegensatz zum Stoffschluss handelt es sich in der Regel um eine lösbare Verbindung [PBF+07].

Aufgrund der Wirkung von Kräften zwischen den Wirkflächen der zu verbindenden Teile entsteht eine **kraftschlüssige** Verbindung. Diese kann weiter unterteilt werden, je nachdem welche Art der Kraft für die Schlussart verwendet wird. Beim Reibkraft-

schluss wird an den Wirkflächen zwischen den zu verbindenden Elementen eine Reibkraft erzeugt. Werden hingegen Feldkräfte, wie z.B. Magnetkraft, für die Verbindung genutzt, handelt es sich bei der Schlussart um einen Feldkraftschluss. Bei dem elastischen Kraftschluss werden Verformungen von elastischen Elementen als Kraftspeicher genutzt. Federelemente sind typische Vertreter dieser Schlussart [PBF+07].

4.1.2 Technische Elemente

Die Systemstruktur wird durch technische Elemente beschrieben, die sich in vier Klassen unterteilen lassen: **Energieumsetzendes Element**, **Stoffumsetzendes Element**, **Informationsumsetzendes Element** sowie **Tragstruktur** (Bild 4-10). Je nachdem wie die Systemgrenze gesetzt ist, sind technische Elemente Teil des Systems (Systemelemente) oder Teil des Umfelds (Umfeldelemente). Ein Element kann in Subelemente zerlegt werden.

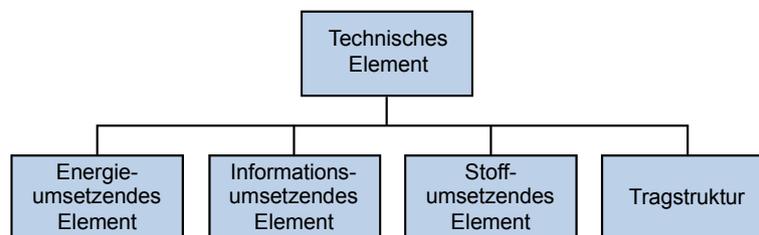


Bild 4-10: Klassifikation der technischen Elemente

Die Unterteilung in die Umsatzarten (Energie, Stoff und Information) wird durch den Hauptfluss des Elements bestimmt, der sich wiederum aus der Hauptfunktion ergibt [Ise02], [PBF+07], [Czi08]. Elemente mit den Hauptfunktionen *Energie wandeln*, *Stoff leiten* oder *Informationen verarbeiten* können eindeutig einer Klasse zugeordnet werden. Dabei ist entscheidend, dass der Hauptfluss (Energie, Stoff, Information oder mechanische Verbindung), spezifiziert durch eine ein- und ausgehende Beziehung am Element, vorhanden ist. Nebenflüsse können hingegen durch eine eingehende oder durch eine ausgehende Beziehung spezifiziert sein.

Dies wird an einem Beispiel veranschaulicht. Die Hauptfunktion einer Pumpe ist das Fördern von Fluiden. Damit entspricht der Hauptfluss der Stoffbeziehung vom Typ «Fluid». Zur Förderung des Stoffs wird eine Energie benötigt. Der Nebenfluss vom Typ «elektrische Energie» ist ein eingehender Nebenfluss. Des Weiteren kann die Pumpe gesteuert werden. Die Informationsbeziehung vom Typ «Signal» mit einer Steuergröße ist ebenfalls ein eingehender Nebenfluss (Bild 4-11).

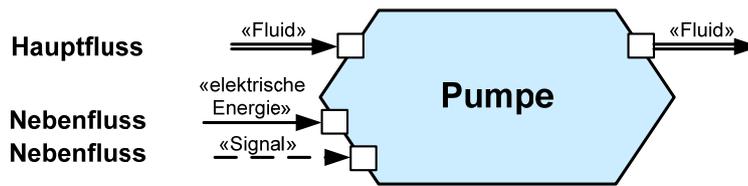


Bild 4-11: Darstellung eines Hauptflusses (Stoff) und der Nebenflüsse (Energie und Information) am Beispiel einer Pumpe

Neben den umsetzenden Elementen gibt es im System konstruktive Elemente, die meistens keine aktive Interaktion (in Form von Energie, Information oder Stoff) besitzen, jedoch einen wesentlichen Bestandteil des Systems bilden. Diese werden in der Elementklasse Tragstruktur geführt. Tragstrukturen und die mechanischen Verbindungen der Bauteile mit der Tragstruktur erfüllen Nebenfunktionen: *gegen Fluide und Umwelteinflüsse abdichten, elektrische oder thermische Energie leiten* sowie Designaspekte (Haptik, Optik, etc).

Im Folgenden wird eine weitere Unterteilung der Elemente durchgeführt, die auf Basis von Grundfunktionen nach KOLLER, PAHL/BEITZ und ROTH beruht [KK94], [PBF+07], [Rot00]. Hierzu wurde untersucht, wie die Elemente, die diese Grundfunktionen erfüllen, in der Systemstruktur modelliert werden.

Die umsetzenden Elemente weisen dabei jeweils zwei Unterklassen auf, die sich aus den Bedingungen der eingehenden und ausgehenden Beziehung ergeben. Unter Berücksichtigung der Klassifikation der Beziehungen (Kap. 4.1.1) ist folgender Zusammenhang vorhanden (Bild 4-12):

- a) Klasse der eingehenden Beziehung ist gleich der Klasse der ausgehenden Beziehung
- b) Klasse der eingehenden Beziehung unterscheidet sich von der Klasse der ausgehenden Beziehung.

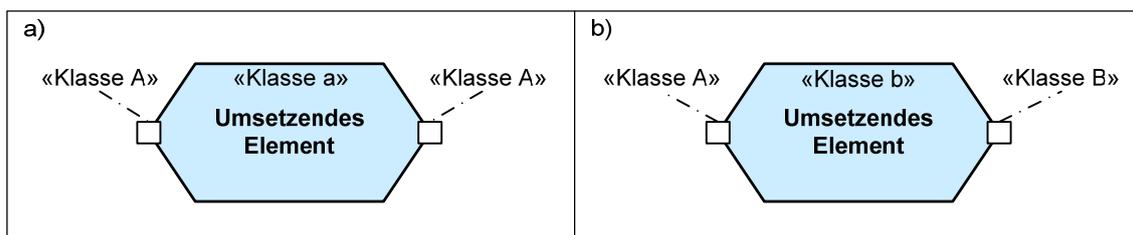


Bild 4-12: Zusammenhang der eingehenden und ausgehenden Beziehungen bei den Unterklassen der umsetzenden Elemente

4.1.2.1 Energieumsetzendes Element

Energieumsetzende Elemente haben eine eingehende und ausgehende Energiebeziehung (Hauptfluss). Aus der eben beschriebenen Unterteilung ergeben sich für Energieumsetzende Elemente die Klassen **Energieübertrager** und **Energiewandler** (Bild 4-13).

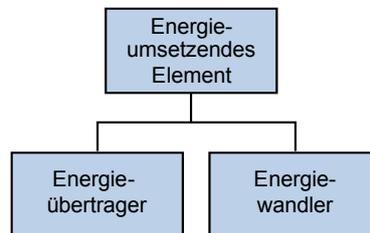


Bild 4-13: Klassifikation der Energieumsetzenden Elemente

Energieübertrager

Die Energieart am Ein- und Ausgang des Energieübertragers ist identisch (Bild 4-14). Dieser Elementtyp ändert zum Beispiel einen Wert der Energiekomponente auf einen anderen Wert (z.B. Spannung U_1 auf einen Wert U_2 ändern) [KK94], [PBF+07]. Energieübertrager erfüllen damit die Grundfunktion *vergrößern/verkleinern*.

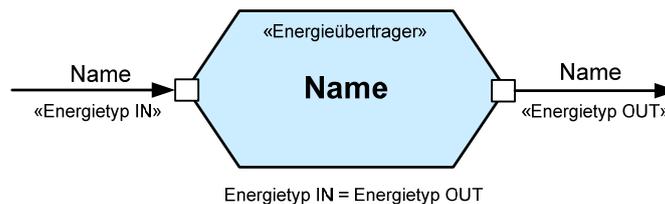


Bild 4-14: Schema eines Energieübertragers

Die Grundfunktion *Richtung ändern* wird ebenfalls vom Energieübertrager realisiert. Die Energie ist eine vektorielle physikalische Größe, deren Richtung vom Element geändert werden kann (z.B. ändert der Spiegel die Richtung der optischen Energie). In der Systemstruktur kann dies über die Parameter beschrieben werden. Hinsichtlich der Energieart ändert sich jedoch zwischen der eingehenden und ausgehenden Beziehung nichts.

Nicht immer ist am Eingang bzw. Ausgang des Elements nur eine Energiebeziehung vorhanden. Diese setzt sich aus mehreren zusammen, wenn die Grundfunktion *sammeln/teilen* bzw. *mischen/trennen* entspricht. Beim *Sammeln* ist die Energiemenge unterschiedlich, wohingegen beim *Mischen* unterschiedliche Energiekomponenten vorhanden sein können (z.B. Wechselströme unterschiedlicher Frequenzen) [KK94].

Beispiel für einen Energieübertrager: Ein *Getriebe* überträgt und ändert ein Drehmoment. Damit ist die eingehende Energie vom Typ *«rotatorische Energie»* mit den Parametern (M_1, ω_1) und ausgehende Energie vom gleichen Typ (*«rotatorische Ener-*

gie»), jedoch mit anderen Parameterwerten (M_2, ω_2). In Bild 4-15 ist das Beispiel mit den Angaben der Parameter dargestellt.



Bild 4-15: Getriebe als Beispiel eines Energieübertragers

Energiewandler

Ein Wandler ändert eine physikalische Größe in eine andere physikalische Größe [KK94]. In diesem Fall handelt es sich bei der physikalischen Größe um Energie. Die Energie kann in verschiedene Energieformen gewandelt werden – z.B. elektrische in mechanische Energie oder mechanische Energie in Wärme. Elemente, die diese Funktion realisieren, werden als Energiewandler bezeichnet. Sie sind in der Systemstruktur dadurch charakterisiert, dass der Hauptfluss vom Typ Energie ist. Zusätzliche Bedingung ist, dass sich die Energieform am Eingang und Ausgang unterscheidet (Bild 4-16).

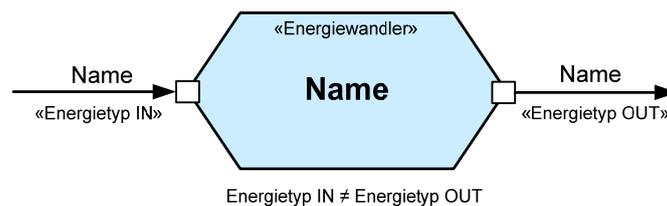


Bild 4-16: Schema eines Energiewandlers

Beispiel für einen Energiewandler: Ein Hydraulikzylinder wandelt hydraulische Energie in mechanische Energie. Je nach Bauart ist am Ausgang eine translatorische oder rotatorische Bewegung vorzufinden. In der Systemstruktur wird ein Hydraulikzylinder nach Bild 4-17 modelliert. Am Eingang ist die «*hydraulische Energie*» und am Ausgang die «*translatorische Energie*» als eine Form der mechanischen Energie spezifiziert.



Bild 4-17: Zylinder als Beispiel für einen Energiewandler

4.1.2.2 Informationsumsetzendes Element

Zu dem Informationsumsetzenden Element zählen zwei Arten – der **Sensor** (Informationsaufnehmer) und die **Informationsverarbeitung** (Bild 4-18). Bei der Informationsverarbeitung ist der Typ der Beziehung am Ein- und Ausgang identisch («*Signal*»). Die-

se Elemente erfüllen die Grundfunktionen *Informationen verknüpfen*, *vervielfältigen*, *umcodieren*, *leiten/isolieren* und *speichern* [KK94]. Sensoren haben wiederum eine eingehende Beziehung vom Typ «Messgröße» und eine ausgehende Beziehung vom Typ «Signal». Anders als bei den Energieumsetzenden Elementen gibt es nicht beliebig viele Kombinationen aus ein- und ausgehender Beziehung. Dies liegt an der Definition der Beziehung Messgröße (vgl. Kap. 4.1.1.2). So ist nicht erlaubt, dass die Informationsverarbeitung am Ein- und Ausgang die Beziehung vom Typ «Messgröße» hat. Ebenfalls kann die Kombination «Signal» IN und «Messgröße» OUT nicht vorkommen.

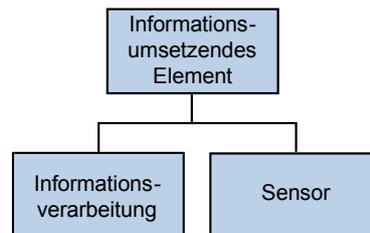


Bild 4-18: Klassifikation der Informationsumsetzenden Elemente

Informationsverarbeitung (IV)

Die Informationsverarbeitung, kurz IV, hat eine ein- und ausgehende Beziehung vom Typ «Signal» (Bild 4-19). Die Herausforderung bei der Darstellung der Software besteht darin, dass diese keine physische Komponente ist, wie z.B. ein Energiewandler in Form eines Elektromotors. Eine Software benötigt jedoch eine physische Komponente – die Hardware, auf der die Software läuft. In der Systemstruktur ist diese nicht aufgeführt – sie gehört zur Realisierung. Die Informationsverarbeitung zeigt auf, durch welche Funktionsträger Signale verarbeitet und an weitere Elemente gesendet werden.



Bild 4-19: Schema einer IV

Beispiel für eine IV: Innerhalb einer Kamera ist eine Softwarekomponente *Bildverarbeitung* vorhanden. Diese wandelt die *Bilddaten*, von einem Bildsensor kommend, zu einem *Bild* – der Hauptfluss ist Information vom Typ «Signal» (Bild 4-20). Wie die Daten entstehen, bzw. wie sie kodiert werden, ist Teil der Realisierung und damit nicht in der Systemstruktur zu erkennen.



Bild 4-20: Bildverarbeitung als Beispiel für eine IV

Sensor

Der Sensor wandelt eine physikalische/chemische Größe in ein elektrisches Signal, so dass daraus eine Information über den Zustand eines Elements (z.B. Temperatur) oder des Flusses (z.B. Durchflussmenge) gewonnen werden kann. Nach welchem physikalischen/chemischen Effekt der Sensor arbeitet, kann in dem Sensor als Eigenschaft hinterlegt werden. Der Sensor hat einen eingehenden Fluss vom Typ «Messgröße» und einen ausgehenden Fluss vom Typ «Signal» (Bild 4-21). Sensoren können auch als Informationsaufnehmer aufgefasst werden. Sie nehmen die Information über die Messgröße auf und wandeln sie in ein Signal.

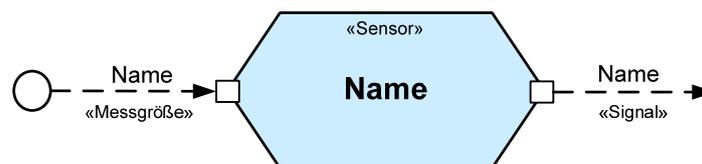


Bild 4-21: Schema eines Sensors

Beispiel für einen Sensor: Zur Messung der Temperatur eignen sich mehrere physikalische Effekte. Die Darstellung erfolgt jedoch stets in dergleichen Art. Die «Messgröße» ist die *Temperatur*. Sie wird vom «Sensor» erfasst und in ein «Signal» gewandelt (Bild 4-22). Dieses Signal kann eine Spannungsänderung sein, die wiederum anschließend in einen Temperaturwert umgerechnet wird. Da die Realisierung der Messung nicht beschrieben wird, wird die Funktionsweise des Sensors aus der Systemstruktur nicht deutlich: Der *PTC* ist ein Kaltleiter, dessen elektrischer Widerstand bei zunehmender Temperatur steigt.

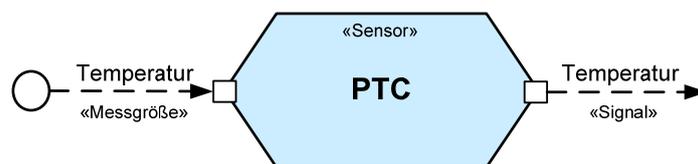


Bild 4-22: PTC-Temperatursensor als Beispiel eines Sensors

4.1.2.3 Stoffumsetzendes Element

Bei Stoffumsetzenden Elementen gibt es ebenfalls eine Unterteilung in Unterklassen, bedingt durch die Ein- und Ausgänge der Elemente. Diese sind jedoch, anders als bei den Energie- oder Informationsumsetzenden Elementen, nicht vom Typ der Beziehung abhängig. Ausgehend von den Grundoperationen nach KOLLER (*wandeln*, *vergrößern/verkleinern*, *leiten/isolieren*, *fügen/lösen*, *mischen/trennen* und *sammeln/teilen* [KK94]) sind folgende Unterklassen gebildet worden: **Eigenschaftsänderndes Element** und **Stoffänderndes Element** (Bild 4-23).

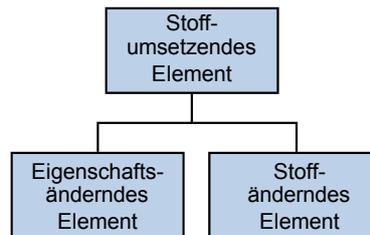


Bild 4-23: Klassifikation der Stoffumsetzenden Elemente

Eigenschaftsänderndes Element

Am Ein- und Ausgang eines Eigenschaftsändernden Elements ist derselbe Stoff vorhanden (Bild 4-24). Das Element ändert dabei die Eigenschaften des Stoffs, nicht den Stoff selbst. Damit erfüllt das Element die Grundfunktionen *vergrößern/verkleinern*, *wandeln* und *leiten/isolieren*. Transportiert oder speichert ein Element die eingehenden Stoffe, so zählt dies ebenfalls zu den Eigenschaftsändernden Elementen, da der Stoff zwischen Ein- und Ausgang nicht geändert wird. Die Struktur ist eine statische Betrachtung des Systems, so dass das Speichern eines Stoffes ebenfalls durch ein- und ausgehende Beziehung desselben Stoffes ist.



Bild 4-24: Schema des Eigenschaftsändernden Elements

Der Typ der Beziehung spielt dabei keine Rolle. Dieser kann sich zwischen dem Ein- und Ausgang ändern, wenn ein Stoff die Eigenschaftsänderung Schmelzen, bzw. Erstarren (Gefrieren) erfährt. Dabei ändert sich der Aggregatzustand eines Stoffes von fest in flüssig, so dass der Typ der Beziehung sich von «Festkörper» in «Fluid» ändert, der Stoff aber gleich bleibt (z.B. Eisen).

Beispiel für ein Eigenschaftsänderndes Element: Der *Kessel* erhitzt *Wasser* von einer Temperatur auf eine andere. Das *Wasser* befindet sich im Zustand T_1 wenn es in den *Kessel* gegeben wird. Sobald das *Wasser* aus dem *Kessel* in das nächste Element gebracht wird, hat es die Temperatur T_2 . Der Stoff ändert sich dabei nicht (Bild 4-25).



Bild 4-25: Kessel als Beispiel eines Eigenschaftsändernden Elements

Bei einer Zustandsänderung des Stoffs muss Energie aufgebracht werden, um die Zustandsänderung zu erhalten. Dies wird in Form eines Nebenflusses modelliert. Das Erhitzen des Wassers benötigt einen Energiefluss vom Typ «thermische Energie».

Stoffänderndes Element

Die zweite Gruppe der Stoffumsetzenden Elemente hat mehrere eingehende und/oder mehrere ausgehende Stoffbeziehungen (Bild 4-26). Dabei unterscheiden sich die eingehenden und ausgehenden Beziehungen in der Art, dass neue Stoffe oder Mischformen (z.B. durch Fügen) entstehen.

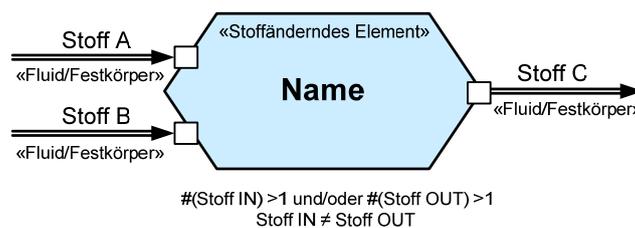


Bild 4-26: Schema des Stoffändernden Elements

Beispiel für ein Stoffänderndes Element: Die *Brühgruppe* eines Kaffeevollautomaten ist ein Stoffänderndes Element (Bild 4-27). In das Element gelangt *Kaffeepulver*. Nachdem dieses verdichtet wurde, wird *Wasser* hindurchgepresst. Der fertige *Kaffee* fließt aus der *Brühgruppe* heraus. Damit wurde aus zwei verschiedenen Stoffen (*Kaffeepulver* und *Wasser*) ein Stoff (*Kaffee*). Da das *Kaffeepulver* sich nicht im *Wasser* auflöst, entsteht der *Kaffeesatz*, der in der *Brühgruppe* zurückbleibt, jedoch wieder entfernt werden muss (ausgehende Stoffbeziehung). Wie dieser Prozess im Detail abläuft, ist im Verhalten des Systems hinterlegt.



Bild 4-27: Brühgruppe eines Kaffeevollautomaten als Beispiel eines Stoffändernden Elements

4.1.2.4 Tragstruktur

Elemente, die keinen Hauptfluss in Form von Energie, Stoff oder Information aufweisen, sind vom Typ Tragstruktur. Eingehende und ausgehende Beziehungen sind mechanische Verbindungen (Bild 4-28). Die Tragstrukturen erfüllen Nebenfunktionen, wie *Kräfte leiten*, *Elemente tragen* oder *vor Umwelteinflüssen schützen*. Darüber hinaus werden nichtfunktionale Anforderungen (z.B. Design und Haptik) an die Elemente der Klasse Tragstrukturen gestellt.



Bild 4-28: Schema einer Tragstruktur

Tragstrukturen können mehrfach in einem Produkt vorhanden sein. Diese haben nicht nur eine Beziehung, sondern sind meistens mit mehreren Bauteilen verbunden. Unterklassen sind nicht vorhanden.

Beispiel für eine Tragstruktur: Das *Gehäuse* eines Produkts umschließt die innenliegenden Bauteile und schützt dadurch vor Umwelteinflüssen (Bild 4-29). Über mechanische Verbindungen ist das *Gehäuse* mit anderen Bauteilen verbunden, die wiederum vom unterschiedlichen Typ sein können (z.B. Energiewandler).



Bild 4-29: Gehäuse als Beispiel einer Tragstruktur

4.1.3 Nicht-technische Elemente

Nicht-technische Elemente, die mit dem System in Interaktion stehen, befinden sich außerhalb der Systemgrenze und werden als Umfeldelemente modelliert. Die Darstellung der Beziehungen zwischen Umfeldelementen und System erfolgt auf die gleiche Weise, wie die Darstellung der Beziehungen zwischen Elementen innerhalb des Systems.

In diesem Abschnitt werden nicht-technische Elemente aufgeführt, die wiederkehrend im Umfeldmodell auftreten. Diese sind: **Lebewesen**, **Stoff-Objekt**, **Umgebung** (Bild 4-30). In den einzelnen Fällen wird auf die Besonderheit der Darstellung eingegangen.

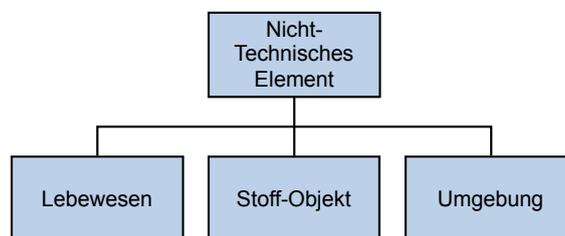


Bild 4-30: Klassifikation der nicht-technischen Elemente

4.1.3.1 Lebewesen

Die Klasse «Lebewesen» schließt sowohl Tiere als auch Menschen ein, die mit dem System interagieren. Tiere interagieren mit dem System, wenn es beispielsweise speziell für sie (z.B. Futterstation) oder für den landwirtschaftlichen Betrieb entwickelt wird (z.B. Melkmaschine). Im Weiteren wird der Fokus auf den Menschen gelegt.

Der Mensch kann mehrere Rollen im Betrieb einnehmen und so unterschiedliche Wechselwirkung mit dem System eingehen. In der Systemstruktur wird die Interaktion des Menschen mit dem System im Betrieb (inkl. Wartung und Service) dargestellt – Wechselwirkungen innerhalb der Entwicklung oder der Fertigung/Montage sind nicht Teil der Systemstruktur. Hierfür eignen sich andere Modelle besser – wie z.B. das Use Case Diagramm der SysML [Wei06].

Im Betrieb agiert der Mensch in der Rolle des Benutzers. Dieser übergibt Informationen an das System in Form des Benutzerwunsches (z.B. An/Aus) und löst dadurch ein Verhalten aus. Zusätzlich kann er wiederum Informationen (z.B. über Systemzustand) erhalten. Dabei kann der Benutzer auf unterschiedliche Weise den Benutzerwunsch übermitteln – z.B. mechanisch (durch das Betätigen einer Taste) oder akustisch (durch Sprachbefehle). Der Benutzerwunsch muss vom System aufgenommen werden. Dies geschieht über Elemente vom Typ «Sensor», da der Benutzerwunsch eine Messgröße darstellt.

Die Informationen, die der Benutzer im Gegenzug erhält, sind vom Typ «Signal». Die Übermittlung der Information kann ebenfalls auf unterschiedliche Weise realisiert werden – z. B. akustisch (durch Töne), optisch (durch eine Anzeige im Display) oder mechanisch (durch die Stellung eines Schalters). Diese Angaben können in der Beziehung hinterlegt werden.

In Bild 4-31 ist am Beispiel des Systems *Funkschlüssel* die Wechselwirkung mit dem *Benutzer* dargestellt. Bei dem *Funkschlüssel* handelt es sich um ein Informationsumsetzendes Element vom Typ «*Sensor*». Die eingehende Information wird als Messgröße erfasst. Dies geschieht zum Beispiel durch Tastendruck. Der Benutzerwunsch *Auto aufschließen* wird vom Funkschlüssel verarbeitet und als Signal an das Fahrzeug gesandt. Der Benutzer bekommt daraufhin zwei Rückmeldungen. Der Funkschlüssel meldet, dass der Benutzerwunsch erfasst wurde. Dies geschieht durch den Tastenwiderstand (haptisch) und durch das Aufleuchten einer LED (optisch). Auch das Fahrzeug sendet eine Information an den Benutzer, dass der Wunsch erhalten und die Tür entsprechend auf- oder abgeschlossen wurde. Dies geschieht optisch durch die Blinker und durch ein akustisches Signal.

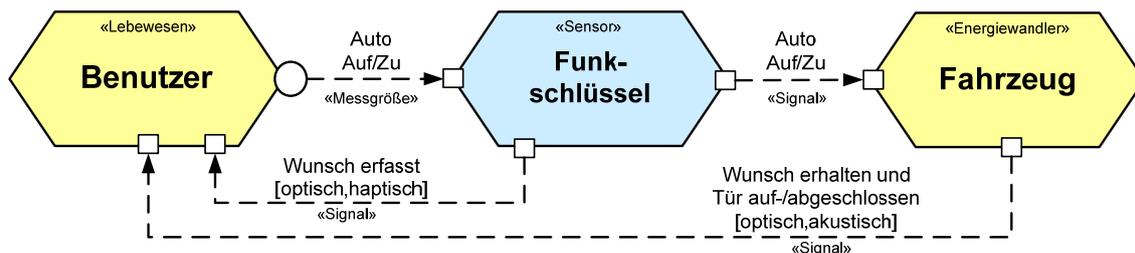


Bild 4-31: Wechselwirkung zwischen Benutzer und System am Beispiel Funk Schlüssel

4.1.3.2 Stoff-Objekt

Handelt es sich bei dem System um eine Anlage, die einen bestimmten Prozess enthält (z.B. sortieren, reinigen, falten), so existiert in der Regel ein **Stoff-Objekt**, das den Prozess durchläuft. Dieses ist nicht Teil des Systems (liegt außerhalb der Systemgrenze) und wird als Umfeldelement dargestellt. Das Besondere an diesem Element ist, dass es zeitgleich ein Umfeldelement und ein Objekt der Stoffbeziehung darstellt. Als Umfeldelement hat dieses eine Wechselwirkung mit dem System, das durch den Prozess charakterisiert wird. Zum Beispiel muss die Wäsche im Prozess Kräfte (z.B. beim Schleudern) aushalten. Das Element als Objekt der Stoffbeziehung erfährt im System Zustandsänderungen. Welche Zustände das Element einnehmen kann, ist in dem Element zu hinterlegen (die *Wäsche* hat z.B. die Zustände: verschmutzt, sauber, nass, feucht). Die Wäsche wird in die Waschmaschine durch den Benutzer gebracht. Dies ist durch die Stoffbeziehung *Wäsche* vom *Benutzer* in die *Waschmaschine* spezifiziert (Bild 4-32).

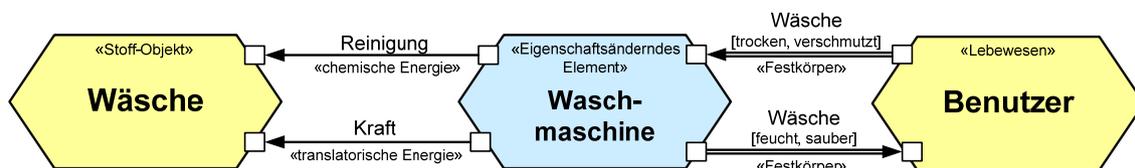


Bild 4-32: Wäsche als Stoff-Objekt sowie als Objekt der Stoffbeziehung

4.1.3.3 Umgebung

Im Betrieb des Systems wirken Umwelteinflüsse, wie z.B. Regen, elektromagnetische Strahlen oder Temperatur. Sie tragen in den meisten Fällen nicht zur direkten Wirkung des Systems bei. Innerhalb bestimmter Wertebereiche sind diese Größen sogar störend bzw. können zum Ausfall des Systems führen. In der Umfeldanalyse werden wesentliche Störgrößen identifiziert, bzw. Störgrößen aufgeführt, die bei bereits ähnlichen Produkten zu Problemen führten.

Alle Umwelteinflüsse können zusammengefasst in einem Umfeldelement «*Umgebung*» aufgeführt werden. Diese steht dann in Wechselwirkung mit dem System (Bild 4-33).

Störende Wirkung ist mit der Eigenschaft „störend“ zu versehen. In der graphischen Präsentation kann dies farblich kenntlich gemacht werden.

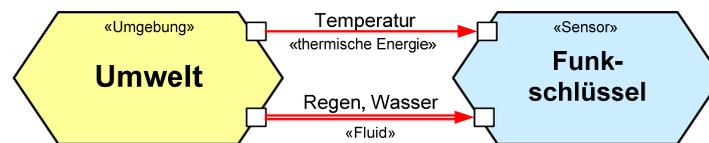


Bild 4-33: Umfeldelement Umgebung in Wechselwirkung mit dem System

4.2 Richtlinien und Bedingungen der Modellierung

Die Systemstruktur beschreibt das System durch seine Elemente und Beziehungen. Die Bestandteile der Systemstruktur wurden im vorherigen Kapitel eingeführt und klassifiziert. Daraus lassen sich Richtlinien und Bedingungen ableiten. Dabei haben Richtlinien einen empfehlenden Charakter, während Bedingungen verpflichtend sind. In der rechnerinternen Modellierung können Bedingungen ausgewertet und auf ihre Einhaltung geprüft werden. Das Ziel ist eine plausible Systemstruktur, so dass die Richtlinien und Bedingungen nach den drei Kriterien *Vergleichbarkeit*, *Vollständigkeit* und *Richtigkeit* sortiert sind. Damit wird nicht ausgeschlossen, dass eine Richtlinie/Bedingung zu mehr als einem Kriterium beitragen kann. Daher erfolgt die Einordnung nach dem Kriterium, das die Richtlinie oder Bedingung hauptsächlich adressiert.

Über die Definition der Bestandteile einer Systemstruktur hinaus, werden Annahmen getroffen, die einen Beitrag zur Plausibilität leisten. Nach einer Erläuterung wird die Richtlinie oder Bedingung formuliert. Anhang A2 enthält eine gesammelte Auflistung.

4.2.1 Vergleichbarkeit

Wird eine Systemstruktur von einem Team erstellt und das gleiche System von einem zweiten Team in einer anderen Systemstruktur beschrieben, so entstehen viele Unterschiede in den Modellen. Dies tritt nicht nur dann auf, wenn unterschiedliche Modellierungssprachen verwendet werden. Auch beim Einsatz derselben Modellierungssprache und demselben zu beschreibenden Produkt sind immer wieder Abweichungen vorhanden. Dies kommt alleine dadurch zustande, dass die Elemente und Beziehungen beliebig benannt werden. Mit der Klassifikation der Elemente und der Beziehungen wird ein Beitrag zur Vergleichbarkeit von Modellen geleistet. Damit ist es möglich, unabhängig von der Bezeichnung der Beziehung oder des Elements eine einheitliche Modellierung zu erreichen.

Bedingung: *Jedes Element und jede Beziehung ist einer Klasse zuzuordnen.*

Aus der Definition der Systemstruktur stellen die Beziehungen eine gewollte Wechselwirkung zwischen zwei Elementen dar. Dabei entscheidet der Zweck der Verbindung (z.B. Übertragung von Informationen) über den Typ der Beziehung. Auf oberster Ebene

wird zwischen Energie, Information, Stoff und mechanischer Verbindung unterschieden.

Richtlinie: *Die Klasse (Energie, Information, Stoff, mechanische Verbindung) einer Beziehung wird durch den Zweck der Verbindung bestimmt.*

Für die Elemente gilt ähnliches Vorgehen. Die Hauptfunktion, respektive Hauptaufgabe des Elements, entscheidet über die Elementklasse. Die Hauptfunktion definiert damit auch den Hauptfluss durch das Element. Auf oberster Ebene wird zwischen Energie-, Information-, Stoffumsetzenden Elementen und Tragstruktur unterschieden.

Richtlinie: *Die Elementklasse der technischen Elemente wird auf Basis der Hauptaufgabe des Elements festgelegt (Energie umsetzen, Information umsetzen, Stoff umsetzen oder Element halten/schützen).*

Bei nicht-technischen Elementen wird zwischen den Klassen «Lebewesen», «Stoff-Objekt» und «Umgebung» unterschieden. Diese befinden sich in der Regel außerhalb der Systemgrenze.

Richtlinie: *Menschen oder Tiere, die mit dem System interagieren, werden durch Elemente mit der Elementklasse «Lebewesen» präsentiert.*

Richtlinie: *Stoffe, die durch das System umgesetzt werden, werden als Elemente vom Typ «Stoff-Objekt» präsentiert.*

Richtlinie: *Einflüsse aus dem Systemumfeld (z.B. Umwelteinflüsse, Raumbedingungen) werden als Beziehung dargestellt. Diese starten am Element vom Typ «Umgebung».*

Elemente können in Subelemente dekomponiert werden (Bild 4-34). Die damit entstehende Hierarchie bringt weitere Bedingungen mit sich, die bei der Modellierung zu berücksichtigen sind. Die Darstellung des Systems in seinem Umfeld stellt die oberste Ebene der Systemstruktur dar. In dieser Ebene ist das System eine Black-Box. In der nächsten Ebene beginnt die Unterteilung des Systems in seine Subsysteme – die White-Box-Betrachtung. Wie viele Hierarchieebenen innerhalb eines Systems vorhanden sind, hängt vom betrachteten System und dem Detaillierungsgrad ab.

Die Dekomposition von Elementen kann nach verschiedenen Gesichtspunkten erfolgen, wie z.B. gestaltorientiert oder funktional. Mit einer rein gestaltorientierten Dekomposition können nicht alle Bestandteile eines mechatronischen Systems abgedeckt werden. Die Software würde dabei nicht berücksichtigt werden. In der Systemstruktur sollte daher die Dekomposition der Elemente nach funktionalen Gesichtspunkten erfolgen.

Richtlinie: *Dekomposition von Elementen nach funktionalen Gesichtspunkten.*

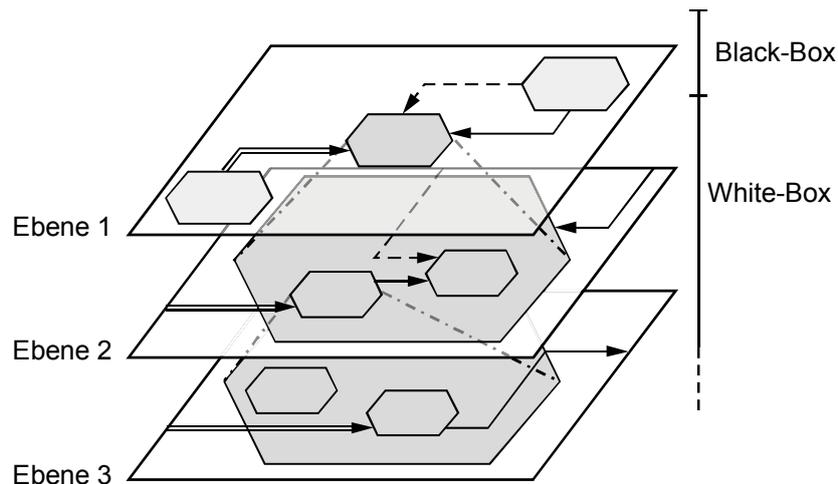


Bild 4-34: Ebenen der Systemstruktur

Dekomponierte Elemente können Baugruppen oder Module darstellen, die gemeinsam eine Funktion erfüllen. Bauteile hingegen können nur auf der untersten Ebene vorhanden sein. Daraus folgt, dass ein dekomponiertes Element kein eigenständiges Bauteil sein kann, wie z.B. Gehäuse eines Systems.

Richtlinie: *Ist ein Element in weitere Elemente dekomponiert, darf das übergeordnete Element keinem Bauteil entsprechen.*

4.2.2 Vollständigkeit

Der Definition nach kann ein Modell nicht vollständig sein, da es die Abstraktion eines Sachverhalts ist (vgl. Kap. 2.1). Für die Systemstruktur können jedoch Annahmen getroffen werden, die zur Vollständigkeit beitragen. Diese ergeben sich aus der Definition der Elemente und Beziehungen. Zum anderen lassen sich Annahmen aus der Tatsache treffen, dass mechatronische Systeme beschrieben werden.

Benennung von Elementen und Beziehungen

Zur Vollständigkeit des Modells gehören auch die Benennung von Elementen und Beziehungen. Diese müssen vorliegen und gut gewählt sein. Dabei gilt für die Elemente, dass die Benennungen für das gesamte Projekt gelten und in den fachdisziplinspezifischen Modellen zu übernehmen sind. Abkürzungen für Elemente, wie bspw. „TKU“ für Tretkraftunterstützung, sind zusätzlich in einem Glossar zu vermerken.

Bei den Beziehungen wird in vielen Fällen aus der Diskussion heraus eine Beziehung zwischen Elementen gezogen, ohne sie zu benennen. Innerhalb der Gruppe herrscht Einigkeit über die Beziehung, jedoch wird beim nächsten Treffen nicht mehr ersichtlich, welche Beziehung damit gemeint war. Daher muss jede Beziehung benannt werden. Dies kann beschreibend (z.B. Versorgungsenergie) sein oder Parameter ($I_{\text{Versorgung}}$, $U_{\text{Versorgung}}$) enthalten.

Bedingung: *Elemente und Beziehungen müssen benannt werden.*

Richtlinie: *Ein projektweites Glossar enthält mindestens die Abkürzungen und die entsprechenden ausgeschriebenen Elementbezeichnungen.*

Hauptfluss der technischen Elemente

Der Hauptfluss muss als Ein- und Ausgang am Element spezifiziert sein. Diese Annahme gilt nur für Elemente innerhalb der Systemgrenze. Für die in Kapitel 4.1.2 definierten Elementklassen sind Bedingungen an den Typ des Hauptflusses bestimmt worden.

Bedingung: *Energieumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Energie».*

Bedingung: *Informationsumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Information».*

Bedingung: *Stoffumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Stoff».*

Bedingung: *Tragstrukturen haben mindestens eine Beziehung vom Typ «mechanische Verbindung».*

Außerhalb der Grenze (für Umfeldelemente) kann die Bedingung verletzt werden. Dies ist zulässig, da bei Umfeldelementen nicht die Vollständigkeit der Elemente im Vordergrund steht, sondern deren Wechselwirkung mit dem System.

Detaillierung der Klassen von Elementen und Beziehungen

Die Klassen der Elemente und Beziehungen können auf zwei Detailstufen vergeben werden. Bei den Elementen entspricht dies auf der unteren Stufe der Unterteilung in Energie-, Stoff- oder Informationsumsetzende Elemente. Die Beziehungen können auf dieser Stufe die Klassen Energie, Stoff, Information oder mechanische Verbindung erhalten. Der Detailgrad steigt mit der nächsten Stufe und der Unterteilung in die Unterklassen (z.B. Energiewandler oder Energieübertrager bei Energieumsetzenden Elementen). Erst wenn alle Elemente und Beziehungen in der höchsten Detailstufe vorliegen, wird das Modell als vollständig betrachtet.

Bedingung: *Alle Elemente und Beziehungen sind einer Klasse der höchsten Detailstufe zuzuordnen.*

Dekomposition von Elementen

Jedes Element kann in weitere Elemente zerlegt werden. Dabei ist ein Element erst dann zu unterteilen, wenn die neue Ebene aus mehr als einem Element besteht. Es liegt ein 1:m-Verhältnis ($m > 1$) zwischen dem dekomponierten Element und seinen Subelementen vor.

Bedingung: Es besteht ein 1:m-Verhältnis, mit $m > 1$, zwischen dem Element der Ebene n und den Elementen der Ebene $n+1$.

Vererbung der Elementklasse

Die eingeführten Klassen für technische Elemente können auf allen Ebenen der Hierarchie angewandt werden. Auf oberer Ebene des Elements existieren viele ein- sowie ausgehende Beziehungen, die von unterschiedlichem Typ sind. In diesem Fall muss die Hauptfunktion und damit der Hauptfluss des Elements identifiziert werden. Ist dieser gefunden, wird in dem Element die Klasse hinterlegt. In der nächsten Hierarchieebene muss mindestens ein Element die Klasse des Vater-Elements erben. Das heißt, ein «Energiewandler» auf Ebene n muss auf Ebene $n+1$ mindestens ein Element vom Typ «Energiewandler» besitzen.

Eine Ausnahme bilden die Eigenschaftsändernden Elemente, die *Stoffe transportieren*. Für diese Funktionen müssen Kräfte auf das zu transportierende Objekt wirken. Erst durch diesen Effekt kann ein Objekt von A nach B bewegt werden. Damit kann das Eigenschaftsändernde Element in der nächsten Hierarchieebene aus «Energiewandler» und «Energieübertrager» bestehen. Am Beispiel von zwei gegenläufigen Rollen zum Transport von Papierblättern wird dies deutlich. Das Eigenschaftsändernde Element *Papierwalze* besteht aus zwei Energieübertrager, die erst in Kombination den Effekt des Transports aufbringen (Bild 4-35).

Bedingung: Mindestens ein Element der Ebene $n+1$ erbt die Elementklasse des Vater-Elements. Die Ausnahme bildet die Elementklasse «Eigenschaftsänderndes Element».

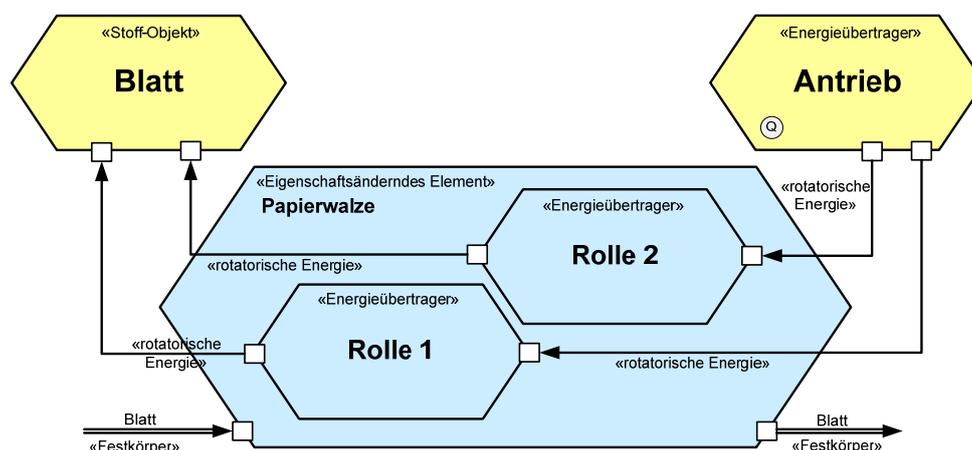


Bild 4-35: Beispiel des Ausnahmefalls für Eigenschaftsändernde Elemente

Beziehungen zwischen Elementen über Hierarchieebenen hinweg

In der obersten Ebene werden Beziehungen zwischen den Umfeldelementen und dem System modelliert. Diese enden damit nicht an der Systemgrenze, sondern müssen in der White-Box Betrachtung weitergeführt werden (Bild 4-36). Dies gilt vor allem für die gewollten Wirkbeziehungen. Alle ungewollten Beziehungen werden in der Sys-

temstruktur dargestellt, um graphisch die störenden Wirkzusammenhänge aufzuzeigen. Diese Beziehungen können, müssen aber nicht, weitergeführt werden. In diesem Fall gilt die Semantik: Endet eine störende Beziehung an der Elementgrenze, so wirkt sich die Beziehung auf alle innenliegenden Elemente aus. Beispielsweise wirkt die Umgebungstemperatur auf das gesamte System, sodass in der Entwicklung der Systemelemente diese Störwirkung berücksichtigt werden muss. In der Konstruktion müssen Materialien und Gestalt so ausgelegt werden, dass die Temperaturen keine ungewünschte Verformung bewirken – oder durch die Konstruktion ein Kühleffekt in die Gestalt integriert werden kann, der wiederum die Elektronik vor Überhitzung schützt.

Bedingung: *Gewollte Wirkbeziehungen vom Typ «Energie», «Information» oder «mechanische Verbindung» müssen bis zur untersten Hierarchieebene geführt werden.*

Richtlinie: *Ungewollte Wirkbeziehungen (störende Einflüsse) können auf beliebiger Hierarchieebene mit der Semantik enden, dass die Wirkung sich auf alle innenliegenden Elemente bezieht.*

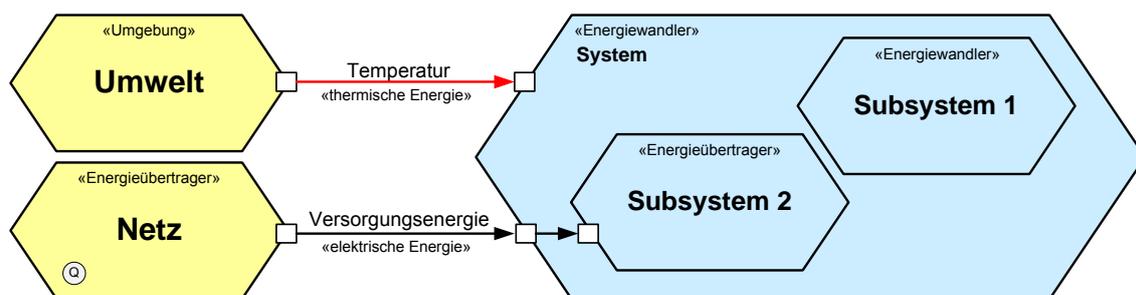


Bild 4-36: Endpunkte einer Beziehung am Beispiel einer gewollten Wirkung und einem störenden Einfluss

Die Eigenschaftsändernden Elemente zeigen durch die oben beschriebene Besonderheit in der Hierarchisierung auch in diesem Fall eine Ausnahme. Damit lautet die Bedingung für diese Elemente: Besteht ein Eigenschaftsänderndes Element in der untersten Hierarchieebene nur noch aus Energiewandler oder/und Energieübertrager, muss die Stoffbeziehung nicht weiter in die unterste Ebene gezogen werden. Am Beispiel des Papiertransports ist eine ein- und ausgehende Stoffbeziehung *Papierblatt* vorhanden. Sie wird nicht in die Ebene der zwei Rollen weitergezogen. Die Wirkung der zwei Rollen auf das Blatt wird durch eine Energiebeziehung auf das Umfedelement «*Stoff-Objekt*» dargestellt (Bild 4-35).

Richtlinie: *Gewollte Wirkbeziehungen vom Typ «Stoff» können am Stoffumsetzenden Element der Ebene n enden, wenn keine Stoffumsetzenden Elemente auf der Ebene $n+1$ enthalten sind.*

Energiequelle

Mechatronische Systeme benötigen ein Element, das die Systemelemente mit Energie versorgt – eine Energiequelle. Diese kann Teil des Systems sein oder sich außerhalb des Systems im Umfeld befinden. Wird in dem System die Quelle integriert, z.B. durch einen Akku, so muss eine Schnittstelle vorhanden sein, diesen wieder aufladen zu können.

Richtlinie: Die Energiequellen des Systems sind mit dem Attribut „Energiequelle“ zu versehen.

Bedingung: In der Systemstruktur ist mindestens eine Energiequelle vorhanden.

Die Energiequelle kann unterschiedliche Energie zur Verfügung stellen: z.B. elektrische, mechanische, hydraulische oder pneumatische Energie. Um welche Art es sich dabei handelt, hängt von der Realisierung des Systems ab. Wird eine Anlage elektrisch betrieben, muss ein entsprechender Anschluss vorhanden sein (bspw. Hochleistungsnetz oder Hausnetz).

Die Vorgaben der Anschlüsse kann Teil der Lösungsfindung sein oder aber vom Kunden vorgegeben werden, wenn dieser vorhandene Anschlüsse nutzen will. Wird bspw. eine Anlage entwickelt, die durch Zylinder betrieben wird, so kann der Kunde vorgeben, dass hierzu die vorhandene Druckluftanlage genutzt werden soll. Damit wirkt sich dieses auf die Lösungsfindung (pneumatischer Zylinder) aus und die Schnittstellen müssen im Vorfeld in das Modell mit einfließen.

In Bild 4-37 ist beispielhaft eine Produktionsanlage gezeigt, die von außen zwei Energiequellen besitzt. Diese sind eine elektrische Energieversorgung am Netz sowie eine Druckluftanlage, die die benötigte pneumatische Energie bereitstellt. Die Elemente *Netz* und *Druckluftquelle* sind sowohl als «Energieübertrager» klassifiziert als auch mit dem Attribut *Energiequelle* versehen worden. In der graphischen Darstellung ist dies durch das Piktogramm mit dem Symbol Q (für Quelle) ersichtlich.

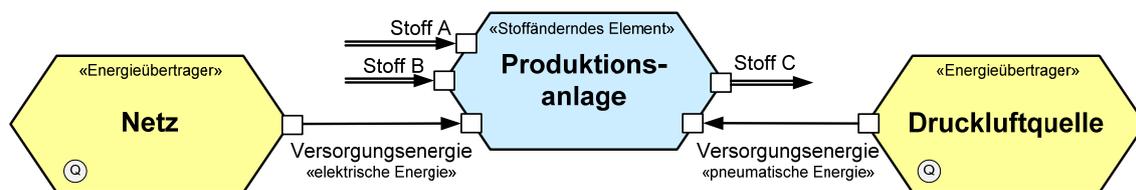


Bild 4-37: Energiequellen eines Systems

Energieverbraucher

Aktive Elemente entsprechen Energieverbrauchern. Das heißt, dass sie zur Erfüllung ihrer Funktion mit elektrischer Energie versorgt werden müssen. In den meisten Fällen sind das die Energieumsetzenden Elemente im System, die durch den Hauptfluss bereits eine eingehende Energiebeziehung besitzen. Diese ist jedoch nicht notwendigerweise vom Typ «elektrische Energie». Darüber hinaus benötigen auch Sensoren oder Eigenschaftsändernde Elemente eine Energieversorgung. Bei den Sensoren kann nicht von

vornherein die Annahme getroffen werden, dass sie stets eine eingehende Energiebeziehung besitzen. Passive Sensoren zum Beispiel benötigen keine elektrische Hilfsenergie, da sie durch das Messprinzip ein elektrisches Signal erzeugen können [Czi08].

Richtlinie: *Elemente, die elektrische Energie benötigen, sind mit dem Attribut „aktiv“ zu versehen.*

Bedingung: *Elemente mit dem Attribut „aktiv“ besitzen mindestens eine eingehende Beziehung vom Typ «elektrische Energie».*

Nebenfluss bei Eigenschaftsändernden Elementen

Findet bei Eigenschaftsändernden Elementen eine Zustandsänderung des Stoffs statt, wird hierzu eine Energie benötigt (vgl. Kap. 4.1.2.3). Die Stoffzustände werden an der Beziehung modelliert. Aus dieser wird ersichtlich, ob ein Zustand durch das Element geändert wurde. Die Energie wird als Nebenfluss spezifiziert. Die Energieart ergibt sich aus dem Wirkprinzip, das zur Zustandsänderung gewählt wird.

Bedingung: *Wird der Zustand eines Stoffes durch ein Eigenschaftsänderndes Element geändert, besitzt das Element mindestens eine eingehende Beziehung vom Typ «Energie».*

Mechanische Verbindung

Elemente können das Attribut *gestaltbehaftet* erhalten. Diese stellen damit Bauteile oder Module dar. Für die Gestalt spielen diese Elemente eine wesentliche Rolle, da sie bei der Bauraumbestimmung und der Festlegung von Wirkflächen berücksichtigt werden müssen. Auch für den Fertigungs- und Montageprozess ist es notwendig, diese Elemente zu identifizieren.

Gestaltbehaftete Elemente haben mindestens eine Beziehung vom Typ «mechanische Verbindung». Die Befestigungsart (kraft-, stoff- oder formschlüssig) muss nicht zwangsläufig angegeben sein. Die Elementklassen «Tragstruktur», «Energieumsetzendes Element» sowie «Stoffumsetzende Elemente» werden auf der untersten Hierarchieebenen automatisch mit dem Attribut versehen. Bei Informationsumsetzenden Elementen muss dies individuell geprüft werden. Dabei kann die Informationsverarbeitung in der Systemstruktur einer Software entsprechen und ist damit nicht gestaltbehaftet. Dies kann vom Benutzer dennoch mit dem Attribut versehen werden, wenn er die dafür notwendige Hardware mit dem Element mit berücksichtigen möchte, ohne sie in der Systemstruktur zu spezifizieren. Die Hardware wird im System ebenfalls einen Bauraum benötigen und auch eine mechanische Verbindung erhalten.

Richtlinie: *Elemente, die einem physischen Element entsprechen, erhalten das Attribut „gestaltbehaftet“.*

Bedingung: *Gestaltbehaftete Elemente haben mindestens eine Beziehung vom Typ «mechanische Verbindung».*

Sind gestaltbehaftete Elemente vorhanden, die miteinander verbunden sind, muss es eine mechanische Verbindung vom System geben, die nach außen an ein Umfeldelement geht. Dies trifft vor allem dann zu, wenn das zu betrachtende System ein Subsystem eines anderen Systems darstellt. Die Elemente können nicht frei im Raum „schweben“ und benötigen damit mindestens eine mechanische Verbindung.

Bedingung: *In der Systemstruktur ist mindestens eine Beziehung vom System zu einem Umfeldelement vom Typ «mechanische Verbindung» vorhanden.*

4.2.3 Richtigkeit

Mit der Definition der Elemente in Kapitel 4.1 sind Bedingungen an die Darstellung formuliert worden. Werden diese eingehalten, ist das Element richtig modelliert. Die Bedingungen beziehen sich auf die Spezifikation der ein- und ausgehenden Beziehungen der Elemente. Zum Beispiel ist ein Energiewandler dann richtig modelliert, wenn die eingehende Energieform sich von der ausgehenden Energieform unterscheidet. Diese Bedingungen werden im Folgenden aufgeführt.

Energieumsetzende Elemente

Bedingung: *Für Elemente der Klasse «Energieübertrager» gilt: Energie-Typ IN = Energie-Typ OUT.*

Bedingung: *Für Elemente der Klasse «Energiewandler» gilt: Energie-Typ IN \neq Energie-Typ OUT.*

Informationsumsetzende Elemente

Bedingung: *Für Elemente der Klasse «Sensor» gilt: «Messgröße» IN und «Signal» OUT.*

Bedingung: *Für Elemente der Klasse «Informationsverarbeitung» gilt: «Signal» IN und «Signal» OUT.*

Stoffumsetzende Elemente

Bedingung: *Für Elemente der Klasse «Eigenschaftsänderndes Element» gilt: Stoff IN = Stoff OUT.*

Bedingung: *Für Elemente der Klasse «Stoffänderndes Element» gilt: Stoff IN \neq Stoff OUT.*

Lebewesen

Bedingung: *Für Elemente der Klasse «Lebewesen» gilt: ausgehende Informationsbeziehung ist vom Typ «Messgröße».*

Stoff-Objekt

Bedingung: *Der Objektfluss durch das System wird durch die Stoffbeziehung spezifiziert. Hierzu wird ein Element vom Typ «Stoff-Objekt» angelegt und mit Merkmalen und Merkmalsausprägungen beschrieben.*

Bedingung: *Ein Element vom Typ «Stoff-Objekt» wird als Umfeldelement visualisiert, wenn Wechselwirkungen des Systems mit dem Objekt spezifiziert werden.*

Bedingung: *Die Stoffbeziehung, die den Objektfluss durch das System beschreibt, darf nicht am entsprechenden Element «Stoff-Objekt» beginnen.*

4.3 Plausibilitätsprüfung

Die Richtlinien und Bedingungen leiten den Modellierer an, die Systemstruktur plausibel zu erstellen. Ist eine Systemstruktur erstellt worden, kann diese auf Plausibilität geprüft werden. Hierzu werden die Richtlinien und Bedingungen auf ihre Einhaltung überprüft. Die systematische Überprüfung erfolgt anhand einer Checkliste (Tabelle 4-2). Diese ist in die drei Bereiche Vergleichbarkeit, Vollständigkeit und Richtigkeit unterteilt. Die einzelnen Fragen sind mit Ja oder Nein zu beantworten und geben damit eine Aussage über die Einhaltung der entsprechenden Richtlinie oder Bedingung. Im Einzelfall wurden aus einer Richtlinie mehrere Fragen erstellt, so dass die Bewertung spezifischer erfolgen kann. Zum Beispiel wurde für die Bedingung: „*Alle Elemente und Beziehungen sind einer Klasse der höchsten Detailstufe zuzuordnen.*“ je Element- und Beziehungs-Klasse der untersten Detailstufe eine Frage generiert (vgl. Fragen 2.7 bis 2.13).

Werden die Fragen mit Ja beantwortet, ist das Modell vergleichbar, vollständig sowie richtig spezifiziert. Wird hingegen eine Frage mit Nein beantwortet, muss die Systemstruktur angepasst werden, um den Modellierungsfehler zu beheben. In einigen Fällen ist dabei projektspezifisch zu überprüfen, ob ein Ausnahmefall vorliegt und damit keine Anpassungen notwendig sind. Zwei Systemstrukturen, die diese Bedingungen erfüllen und den gleichen Sachverhalt darstellen, sind damit vergleichbar, vollständig und richtig modelliert.

Tabelle 4-2: Checkliste zur Plausibilitätsprüfung

	Nr.	Frage	Ja/ Nein
Vergleichbarkeit	1.1	Ist jedes Element einer Klasse zugeordnet?	
	1.2	Ist jede Beziehung einer Klasse zugeordnet?	
	1.3	Entspricht die Beziehungsart (Information, Energie, Stoff, mechanische Verbindung) dem Zweck der Verbindung zwischen den Elementen?	
	1.4	Entspricht die Elementklasse der Hauptaufgabe des Elements (Energie umsetzen, Information umsetzen, Stoff umsetzen oder Element halten/schützen)?	
	1.5	Falls Menschen oder Tiere mit dem System interagieren, sind diese Elemente der Klasse «Lebewesen» zugeordnet?	
	1.6	Falls Stoffe vom System umgesetzt werden, wurde ein entsprechendes Element vom Typ «Stoff-Objekt» angelegt?	
	1.7	Wurden Einflüsse aus dem nicht-technischen Umfeld als Beziehung dargestellt, die am Element vom Typ «Umgebung» starten?	
	1.8	Wurde das System funktional dekomponiert?	
	1.9	Falls Elemente physische Bauteile repräsentieren, sind diese auf der untersten Hierarchieebene spezifiziert?	
Vollständigkeit	2.1	Ist jedes Element und jede Beziehung benannt?	
	2.2	Ist ein projektweites Glossar angelegt mit Abkürzungen und den entsprechenden ausgeschriebenen Elementbezeichnungen?	
	2.3	Haben Energieumsetzende Elemente eine ein- <u>und</u> ausgehende Beziehung vom Typ «Energie»?	
	2.4	Haben Informationsumsetzende Elemente eine ein- <u>und</u> ausgehende Beziehung vom Typ «Information»?	
	2.5	Haben Stoffumsetzende Elemente eine ein- <u>und</u> ausgehende Beziehung vom Typ «Stoff»?	
	2.6	Haben Tragstrukturen mindestens eine Beziehung vom Typ «mechanische Verbindung»?	
	2.7	Falls Beziehungen vom Typ «Energie» vorhanden sind, wurden sie mit der Klasse versehen, die der Energieform entspricht?	
	2.8	Falls Beziehungen vom Typ «Information» vorhanden sind, wurden sie mit der Klasse «Messgröße» oder «Signal» versehen?	
	2.9	Falls Beziehungen vom Typ «Stoff» vorhanden sind, wurden sie mit der Klasse «Fluid» oder «Festkörper» versehen?	
	2.10	Falls Beziehungen vom Typ «mechanische Verbindung» vorhanden sind, wurden sie mit der Klasse «Stoffschluss», «Formschluss» oder «Kraftschluss» versehen?	
	2.11	Falls Energieumsetzende Elemente vorhanden sind, wurden sie mit der Klasse «Energieübertrager» oder «Energiewandler» versehen?	
	2.12	Falls Informationsumsetzende Elemente vorhanden sind, wurden sie mit der Klasse «Sensor» oder «Informationsverarbeitung» versehen?	

	Nr.	Frage	Ja/ Nein
Vollständigkeit	2.13	Falls Stoffumsetzende Elemente vorhanden sind, wurden sie mit der Klasse «Eigenschaftsänderndes Element» oder «Stoffänderndes Element» versehen?	
	2.14	Besteht eine 1:m-Verhältnis, mit $m > 1$, zwischen dem Element der Ebene n und den Elementen der Ebene n+1?	
	2.15	Ist mindestens ein Element (Kind) der unteren Hierarchieebene n+1 vom gleichen Typ wie das übergeordnete Element (Vater) auf Ebene n?	
	2.16	Wurde jede gewollte Wirkbeziehungen vom Typ «Energie», «Information» oder «mechanische Verbindung» bis zur untersten Hierarchieebene geführt?	
	2.17	Hat jede ungewollte Wirkbeziehung, die auf beliebiger Hierarchieebene endet die Semantik, dass die Wirkung sich auf alle innenliegenden Elemente bezieht?	
	2.18	Ist mindestens eine Energiequelle vorhanden?	
	2.19	Sind Energieverbraucher vorhanden und wurden diese mit dem Attribut „aktiv“ versehen?	
	2.20	Falls aktive Elemente vorhanden sind, haben diese eine eingehende Beziehung vom Typ «elektrische Energie»?	
	2.21	Besitzen Eigenschaftsändernden Elemente, die eine Zustandsänderung am Stoff durchführen, mindestens eine eingehende Beziehung vom Typ «Energie»?	
	2.22	Sind gestaltbehaftete Elemente vorhanden und wurden diese mit dem Attribut „gestaltbehaftet“ versehen?	
	2.23	Falls gestaltbehaftete Elemente vorhanden sind, haben diese mindestens eine Beziehung vom Typ «mechanische Verbindung»?	
	2.24	Ist mindestens eine Beziehung vom System zu einem Umfeldelement vom Typ «mechanische Verbindung» vorhanden?	
Richtigkeit	3.1	Ist für jedes Element der Klasse «Energieübertrager» die Bedingung erfüllt Energie-Typ IN = Energie-Typ OUT?	
	3.2	Ist für jedes Element der Klasse «Energiewandler» die Bedingung erfüllt Energie-Typ IN \neq Energie-Typ OUT?	
	3.3	Ist für jedes Element der Klasse «Sensor» die Bedingung erfüllt «Messgröße» IN und «Signal» OUT?	
	3.4	Ist für jedes Element der Klasse «Informationsverarbeitung» die Bedingung erfüllt «Signal» IN und «Signal» OUT?	
	3.5	Ist für jedes Element der Klasse «Eigenschaftsänderndes Element» die Bedingung erfüllt Stoff IN = Stoff OUT?	
	3.6	Ist für jedes Element der Klasse «Stoffänderndes Element» die Bedingung erfüllt Stoff IN \neq Stoff OUT?	
	3.7	Ist für jedes Element der Klasse «Lebewesen» die ausgehende Information-Beziehung vom Typ «Messgröße»?	
	3.8	Wurde das Element vom Typ «Stoff-Objekt» mit Merkmalen und Merkmalsausprägungen beschrieben?	
	3.9	Wurde die Wechselwirkung des Systems mit dem Stoff durch Beziehungen vom System auf das entsprechende Element vom Typ «Stoff-Objekt» modelliert?	
	3.10	Trifft die Aussage zu: „Am Element vom Typ «Stoff-Objekt» ist keine Stoff-Beziehung vorhanden, die den Objektfluss durch das System beschreibt“?	

Die Richtigkeit der Systemstruktur richtet sich an die Syntax des Modells. Die semantische Richtigkeit kann nicht überprüft werden, da sie sich erst aus dem Kontext ergibt. Für einzelne Elemente können jedoch Hilfsmittel bereitgestellt werden, die den Anwender bei der Überprüfung unterstützen. Im Folgenden werden zwei Matrizen vorgestellt, die sich auf die Überprüfung von Energieumsetzenden Elementen sowie Sensoren beziehen.

4.3.1 Matrix der energieumsetzenden Elemente

Die Energieumsetzenden Elemente können in einer Matrix aufgeführt werden [Rop75], [Czi08]. In dieser werden die ein- und ausgehenden Energiebeziehungen gegenübergestellt. Die Energieformen werden in der vertikalen und in der horizontalen Achse aufgeführt, wobei die vertikale Achse die eingehenden Beziehungen und die horizontale Achse die ausgehenden Beziehungen enthält. Durch diesen Aufbau können Energieübertrager und -wandler eingeordnet werden.

Dabei sind Energieübertrager in der Diagonale aufgeführt, da diese die gleiche Energieform als ein- und ausgehende Beziehung aufweisen. Energiewandler finden sich in den Feldern über und unter der Diagonale. Bild 4-38 zeigt einen Ausschnitt mit z.B. *Akkumulator* als Energieübertrager im Kreuzungspunkt *elektrische Energie IN/OUT*. Energiewandler, wie z.B. Elektromotoren (*DC-Motor*) sind im Kreuzungspunkt *elektrische Energie IN* und *rotatorische Energie OUT*. *Linearmotoren*, die ebenfalls zu Elektromotoren gehören, haben *elektrische Energie IN* und *translatorische Energie OUT*.

		Energie				
		Translatorisch	Rotatorisch	Elektrisch	Pneumatisch	...
Energie	Translatorisch	Hebel, Feder, Dämpfer, Hydraulik-Pressse, Keil	Zahnradstange	Piezo	Kolbenpumpe, Luftpumpe	
	Rotatorisch	Gewindespindel, Nockengetriebe, Zahnradstange	Gebtriebe, Welle, Rad, Zahnrad	Generator	Verdichter	
	Elektrisch	Hubaktor, Linearmotor, Piezo, Zylinder	Rotationsmotor, DC-Motor, AC-Motor	Akkumulator, Leistungselektronik, Netzteil	Pneumatikpumpe	
	Pneumatisch	Zylinder	Turbine, Windrad	Pneumatikgenerator	Ventil, Druckminderer	
	...					

Energie-wandler

Energie-übertrager

Bild 4-38: Ausschnitt der Matrix mit Energieumsetzenden Elementen (nach [Czi08])

Wird diese Matrix mit gängigen Energieumsetzenden Elementen gefüllt, kann sie bei der Überprüfung der Richtigkeit unterstützend eingesetzt werden. Im Anhang (A3) ist die Matrix beispielhaft mit Elementen gefüllt. Sie dient als Ausgangsbasis und muss unternehmensspezifisch ausgeprägt werden. Dabei können die Elemente als Lösungselemente aufgeführt werden und damit sehr detailliert beschrieben sein, oder als Elementkategorie. Innerhalb einer Kategorie haben die Elemente gemeinsame Eigenschaften, die in einer Detaillierung unterschiedlich ausgeprägt sein können. Die Kategorie Akkumulator hat beispielsweise die Eigenschaften: Kapazität, Ladezyklen, Leistungsdichte, Energiedichte, Nennspannung. Je nach Art des Akkus variieren die Angaben zu den Eigenschaften.

Bei der Überprüfung der Richtigkeit werden die Informationen der Matrix wie folgt genutzt: Im ersten Schritt werden die Energieübertrager und Energiewandler in der Systemstruktur identifiziert. Je Element wird die ein- und ausgehende Beziehung bestimmt. Dieses definiert den Kreuzungspunkt in der Matrix. Im nächsten Schritt folgt der Abgleich des Elements in der Systemstruktur mit den Elementen im Kreuzungspunkt. Stimmt ein Element der Matrix mit dem beschriebenen Element überein oder entspricht es der richtigen Kategorie, so ist das Element in der Systemstruktur richtig spezifiziert. Das heißt, dass das gewählte Element in der Systemstruktur den benötigten In- und Output bereitstellen kann.

Beispiel: In der Systemstruktur wird ein Elektromotor mit eingehender Beziehung vom Typ elektrische Energie und ausgehender Beziehung vom Typ translatorische Energie spezifiziert. Im Kreuzungspunkt der Matrix ist der Linearmotor aufgeführt. Der Anwender kann nun entscheiden, ob das Element mit der Bezeichnung „Elektromotor“ in die Kategorie Linearmotor fällt. Trifft dies zu, kann er die Benennung präzisieren in „Linearmotor“ oder die Kategorie dem Element als Information hinterlegen. Trifft die Kategorie nicht zu, sind die ein- und ausgehenden Beziehungen zu hinterfragen und gegebenenfalls anzupassen.

Die Darstellungsform der Matrix für Energieumsetzende Elemente ist angelehnt an die Funktionsgrößenmatrix (FGM) der VDI Richtlinie 2222 [VDI2222]. Die quadratische Matrix beinhaltet physikalische Größen in Zeilen und Spalten (Bild 4-39). Besteht ein physikalischer Zusammenhang zwischen zwei Größen, ist dieses in dem Kreuzungspunkt markiert. Der zugehörige physikalische Effekt ist mit entsprechender Formel in einem Katalog hinterlegt. In der Konstruktion wird die Matrix als Hilfsmittel bei der systematischen Suche nach Wirkprinzipien genutzt [Rot00], [FHH+05], [PL11].

	F	s	P _I	v	M	φ	L	ω	u	π	V	p	I	Φ	Q	U
F	■															
s		■														
P _I			■													
v				■												
M					■											
φ						■										
L							■									
ω								■								
u									■							
π										■						
V											■					
p												■				
I													■			
Φ														■		
Q															■	
U																■

■ Direkter Effekt (quartifizierbare Beziehung) zwischen beiden Variablen existiert

Kraft durch Druck
 $F=p \cdot A$

Bild 4-39: Funktionsgrößenmatrix (nach [VDI2222])

Besteht kein direkter Zusammenhang zwischen zwei Größen, kann über eine Verkettung aus mehreren Effekten ein Zusammenhang gebildet werden [VDI2222], [FHH+05]. Jeder Output kann wiederum als Input genutzt und damit die Kombination erarbeitet werden. Die Funktionsgrößenmatrix hat eine andere Abstraktionstiefe als die Matrix mit Energieumsetzenden Elementen. In der Systemstruktur wird auf der Ebene der Elemente modelliert, die mehrere physikalische Effekte enthalten können. Daher werden die Konzepte der FGM auf die Ebene der Elemente übertragen und in Form der Matrix mit Energieumsetzenden Elementen genutzt.

Damit kann die Matrix mit Energieumsetzenden Elementen auch im konstruktiven Sinne genutzt werden. Mit der Verkettung von Elementen (Output eines Elements ist Input des nächsten Elements) kann eine Wirkkette im System beschrieben werden. Darauf wird in Kapitel 4.5 *Werkzeugunterstützung* näher eingegangen.

4.3.2 Matrix der Sensoren

Analog zu Energieumsetzenden Elementen existieren Matrizen für Sensoren (z.B. in [Czi08]). Die Matrix stellt die Messgröße gegenüber dem Output dar und beschreibt damit das Wirkprinzip (Bild 4-40). In der Systemstruktur wird jedoch nur die Messgröße (als eingehende Beziehung) spezifiziert. Die Matrix der Sensoren kann wie folgt zur Überprüfung genutzt werden. Innerhalb der Systemstruktur werden die Elemente vom Typ «Sensor» identifiziert. Die eingehende Beziehung vom Typ «Messgröße» bestimmt die Zeile der Matrix. Ausgehend von der Messgröße wird der spezifizierte Sensor mit den Einträgen in der Zeile der Matrix abgeglichen. Stimmt der spezifizierte Sensor mit einem Element der Matrix überein oder mit der Kategorie, so ist der Sensor in der Systemstruktur richtig modelliert. In der Systemstruktur kann das Wirkprinzip dem Element vom Typ «Sensor» als Eigenschaft hinterlegt werden.

		Wirkprinzip					
		resistiv R	induktiv L	kapazitiv C	Spannung U	Strom I	Ladung Q
Messgröße	Position: Länge l Weg s Winkel ϕ	Potentiometer Magnetoresistiver Sensor, Gauß-Feldplatte	Differenzial- Transformator, Tauchanker- Wegsensor	Kapazitiver Wegsensor	Hall-Sensor, Op- toelektronischer Lichtschranken- Sensor	Wirbelstrom- Sensor	
	Dehnung $\varepsilon = \Delta l/l_0$	Dehnungsmess- streifen				Faseroptische Sensoren	
	Kraft F Moment F · l	Piezoresistiver Sensor, Dehnstoffe	Magnetoelasti- scher Sensor	Kraftkompensa- tions-Sensor	Federelemente oder Dehnungs- messstreifen, Rückführung auf s-, ε -Messung		Piezoelektri- scher Sensor
					

Bild 4-40: Ausschnitt aus der Sensoren-Matrix [Czi08, S.61]

Beispiel: In der Systemstruktur ist die Messgröße *Kraft* als eingehende Beziehung am Sensor *Dehnungsmessstreifen* spezifiziert. In der Matrix ist in der Zeile „Kraft“ der Dehnungsmessstreifen beim Wirkprinzip „Spannung“ aufgeführt. Die Kraft führt zu einer Dehnung, so dass sich der Widerstandswert verändert. Aus dem Messsignal kann die Dehnung und daraus die Kraft abgeleitet werden. Der in der Systemstruktur spezifizierte Sensor kann die benötigte Messgröße bestimmen und ist damit richtig modelliert.

4.4 Sichten auf die Systemstruktur

Die graphische Modellierung bringt den Vorteil mit sich, dass die Modelle übersichtlich und einfach zu lesen sind. Die Übersichtlichkeit sinkt jedoch mit der Anzahl an dargestellten Modellelementen. Diese Problematik ist in der Systemstruktur vorzufinden. Es sind viele Elemente durch viele Beziehungen miteinander verbunden. Die visuelle Wahrnehmung der Systemstruktur ist jedoch ein wichtiger Faktor, da das Modell unter anderem zur Kommunikation zwischen Fachdisziplinen dient. Leserlichkeit und Anschaulichkeit müssen bei der Modellierung berücksichtigt werden (Grundsatz der Klarheit [BPV12]). Die Leserlichkeit kann durch den Einsatz von Filtern erreicht werden. Dabei bildet der Filter eine Sicht auf die Systemstruktur, indem Modellelemente ein- oder ausgeblendet werden.

Die Hierarchisierung entspricht einem Filter, der die Systemebenen enthält. Hierdurch können Wirkzusammenhänge auf unterschiedlichen Systemebenen betrachtet werden. Die wahrgenommene visuelle Komplexität nimmt dabei ab, da die Anzahl an betrachteten Elementen reduziert wird. Darüber hinaus können Filter definiert werden, die unabhängig von der Systemebene, eine Sicht auf die Systemstruktur bieten. Die Voraussetzung hierfür ist Klassifikation der Elemente und Beziehungen nach Kapitel 4.1. Die Filterung kann durch beliebige Kombination der Klassen erzeugt werden. Dennoch ist es zweckmäßig, vordefinierte Filter anzubieten [Neg06]. Diese werden im Folgenden vorgestellt.

Die Filterung erfolgt dabei über den Typ der Beziehung. Je nachdem, ob es sich um eine Beziehung des Typs «Energie», «Information», «Stoff» oder «mechanische Verbindung» handelt, werden die Beziehungen und alle damit verbundenen Elemente angezeigt. Die nicht ausgewählten Beziehungen und Elemente werden dabei nicht komplett ausgeblendet, sondern optisch in den Hintergrund gerückt.

Energiespezifische Sicht: In der energiespezifischen Sicht werden alle Energiebeziehungen angezeigt (Bild 4-41). Die damit verbundenen Systemelemente sind Energieumsetzende Elemente, da die Energiebeziehung den Hauptfluss darstellt. Darüber hinaus sind auch andere Elemente eingebunden. Es werden zusätzlich Elemente angezeigt, die den Energiefluss als Nebenfluss haben. Auf Basis dieser Sicht kann eine Energiebilanz für das System aufgestellt und überprüft werden. Fehlende Beziehungen, wie zum Beispiel die Verlustleistung in Form von Wärme, kann nachgetragen werden. Diese Wirkung ist in den meisten Fällen wiederum eine störende Beziehung und wird als solche gekennzeichnet (z.B. rot markiert).

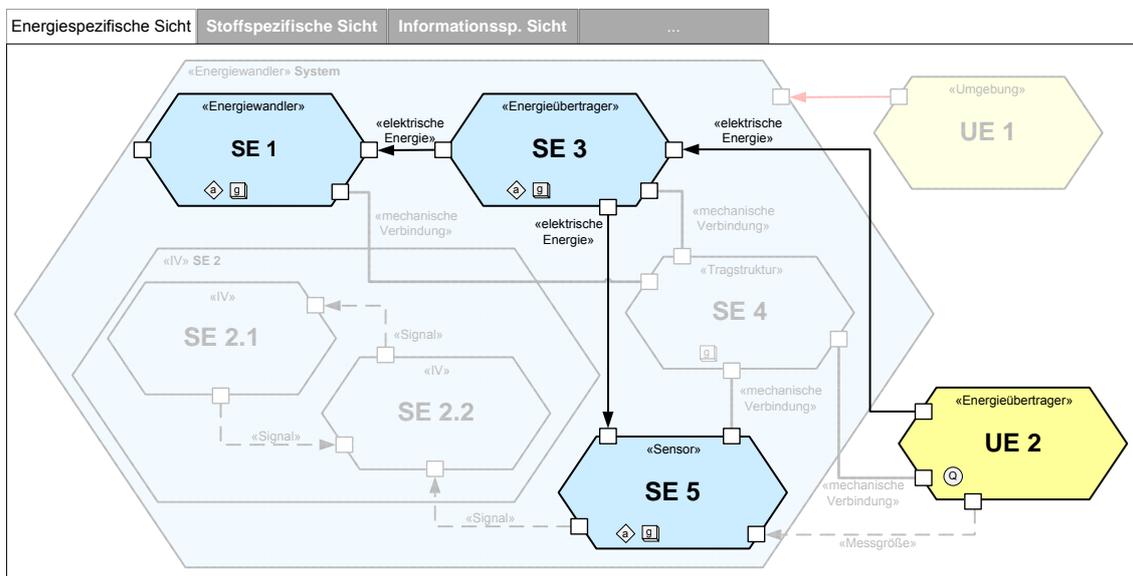


Bild 4-41: Schematische Darstellung der energiespezifischen Sicht

Durch die Unterteilung der Energie in die Energieformen kann je Form oder in Kombination aus mehreren Energieformen ein Filter generiert werden. Eine Sicht kann beispielsweise nur elektrische Energie enthalten, so dass die Fachdisziplin Elektrotechnik analysieren kann, ob alle Beziehungen berücksichtigt wurden. Aus dieser Sicht kann eine Aussage getroffen werden, inwieweit die vorgesehene Energiequelle ausreichend ist oder weitere Quellen, bzw. Transformatoren eingebracht werden müssen.

Stoffspezifische Sicht: Mit der stoffspezifischen Sicht wird der Fluss des Stoffs durch das System hervorgehoben. In dieser Sicht sind alle Stoffumsetzenden Elemente enthalten. Eine gefilterte Sicht ermöglicht dem Modellierer, den Stofffluss und damit die Eigenschaftsänderung näher zu untersuchen. Analog zur energiespezifischen Sicht, in der das Erstellen einer Energiebilanz ermöglicht wird, kann hier eine Stoffbilanz erfolgen.

Informationsspezifische Sicht: Elemente, die durch Informationsbeziehung verbunden sind, werden in der informationsspezifischen Sicht angezeigt. Damit sind alle Sensoren und Informationsverarbeitungen enthalten. Auf dieser Basis kann die zeitliche Abfolge der Kommunikation analysiert werden. Hierzu eignen sich andere Darstellungsformen, wie z.B. die Sequenzdiagramme (aus dem Aspekt Verhalten). Dazu werden die einzelnen Elemente aufgetragen und die Kommunikation zwischen diesen in einer Abfolge dargestellt. Aus den Sequenzdiagrammen kann wiederum abgeleitet werden, ob Beziehungen nicht detailliert genug spezifiziert wurden oder noch fehlen.

Messtechnische Sicht: Die Messgrößen-Erfassung, als Teil der informationsspezifischen Sicht, kann gesondert dargestellt werden. Diese enthält alle Beziehungen vom Typ Messgröße und die damit verbundenen Systemelemente. So entsteht eine Sicht mit Sensoren und den Quellen, an der die Messgröße erfasst werden soll. Messgenauigkeiten, die Realisierung der Sensoren, evtl. Erweiterungen um Digital-Analog-Wandler können auf Basis dieser Sicht diskutiert werden. Auf Systemebene betrachtet bietet diese Sicht darüber hinaus die Möglichkeit, Sensoren zu kombinieren oder evtl. redundante Sensoren zu eliminieren. Dies tritt dann auf, wenn Messgrößen aus bereits vorhandenen Sensoren gewonnen werden können. Ist im System z.B. ein Beschleunigungssensor vorhanden, der zur Berechnung der Geschwindigkeit genutzt wird, so kann dieser auch zur Bestimmung der Lage des Systems herangezogen werden.

Gestaltorientierte Sicht: In der gestaltorientierten Sicht werden alle gestaltbehafteten Elemente und deren mechanischen Verbindungen dargestellt. Diese Sicht bildet die Basis für die Erstellung einer Baustruktur, aus der sich das Produktionssystem ableiten lässt [Nor12]. Hierzu ist es wichtig, die mechanischen Verbindungen vollständig abzubilden. Die gestaltorientierte Sicht unterstützt dabei die Überprüfung und Ergänzung fehlender Beziehungen. Innerhalb des Systems sind viele konstruktive Beziehungen vorhanden. Diese in einer rein graphischen Sicht darzustellen, kann zu unübersichtlichen Modellen führen. Matrizen können bei der Modellierung unterstützen.

Bild 4-42 zeigt die Beziehungsmatrix mit allen gestaltbehafteten Elementen in der Zeile und Spalte. Die quadratische Matrix hat in der Diagonale keine Einträge. Alle anderen Kreuzungspunkte stellen die Verbindung zweier Elemente dar. Es handelt sich dabei um eine an der Diagonalen gespiegelten Matrix. Zur Unterscheidung der Verbindung wird ein Schema verwendet (K=Kraftschluss, F=Formschluss, S=Stoffschluss, X= keine Angabe zur Schlussart).

Beziehungsmatrix								
Fragestellung: „Ist das Element (Zeile) mit dem Element (Spalte) mechanisch Verbunden?“ S = Stoffschluss F = Formschluss K = Kraftschluss X = k. A. zur Schlussart		gestaltbehaftete Elemente	Systemelement 1	Systemelement 2	Systemelement 3	Systemelement 4	Umfeldelement 1	Umfeldelement 2
		Nr.	1	2	3	4	5	6
Systemelement 1	1						F	
Systemelement 2	2			X			K	
Systemelement 3	3		X			S	K	
Systemelement 4	4						K	
Umfeldelement 1	5			S				
Umfeldelement 2	6	F	K	K	K			

Bild 4-42: Beziehungsmatrix der gestaltbehafteten Elemente

4.5 Werkzeugunterstützung

Mit der Klassifikation der Elemente und Beziehungen werden die erstellten Modelle formalisiert dargestellt. Mit einem geeigneten Softwarewerkzeug können die Klassifikationen genutzt werden, die in vorherigen Kapiteln beschriebenen Konzepte automatisch umzusetzen. Durch die rechnerunterstützte Auswertung der Modelle wird der Benutzer auf fehlende oder fehlerhafte Darstellungen hingewiesen.

Zusätzlich kann die Klassifikation den Benutzer unterstützen, das Modell effizient rechnerintern zu erstellen. Im Kapitel 4.5.1 wird aufgezeigt, wie ein Werkzeug gestaltet sein muss, um diese Funktionen umzusetzen. Anschließend wird eine prototypische Implementierung der Plausibilitätsprüfung im Mechatronic Modeller und einem SysML-Werkzeug vorgestellt (Kap. 4.5.2).

4.5.1 Konzept der Werkzeugunterstützung

Für die Erstellung der Systemstruktur ist ein graphischer Editor mit einer entsprechenden Toolpalette notwendig. In dem graphischen Editor werden die Elemente angelegt und Beziehungen zwischen diesen gezogen. Neben der graphischen Darstellung müssen die Modellinhalte formalisiert als Daten abgespeichert werden. Dies unterscheidet die Modellierungswerkzeuge von Visualisierungsprogrammen (wie bspw. Microsoft Office Visio). Die formalisierten Modellinhalte können somit rechnerintern verarbeitet und ausgewertet werden.

In Anlehnung an bestehende Modellierungswerkzeuge, wie z.B. Mechatronic Modeller, wird die Benutzeroberfläche gestaltet (Bild 4-43). In der Toolpalette sind die Konstrukte *Umfeldelement*, *Systemelement* sowie die Ports zu den Beziehungen *Energie*, *Information*, *Stoff* und *mechanische Verbindung* vorhanden. Damit ist die Toolpalette auf die wesentlichen Konstrukte reduziert und übersichtlich gestaltet. Zusätzlich verfügt das Werkzeug über eine erweiterte Toolpalette, die alle Klassen enthält.

4.5.1.1 Arbeitsweise und Werkzeugfunktionen

Beim Anlegen eines Elements muss der Modellierer zu Beginn entscheiden, ob dieses Element innerhalb (Systemelement) oder außerhalb (Umfeldelement) der Systemgrenze liegt. Das Element wird anschließend im Editor als graphisches Konstrukt angelegt und der Modellierer über die Klassifikation des Elements abgefragt (Bild 4-43, Schritt 1).

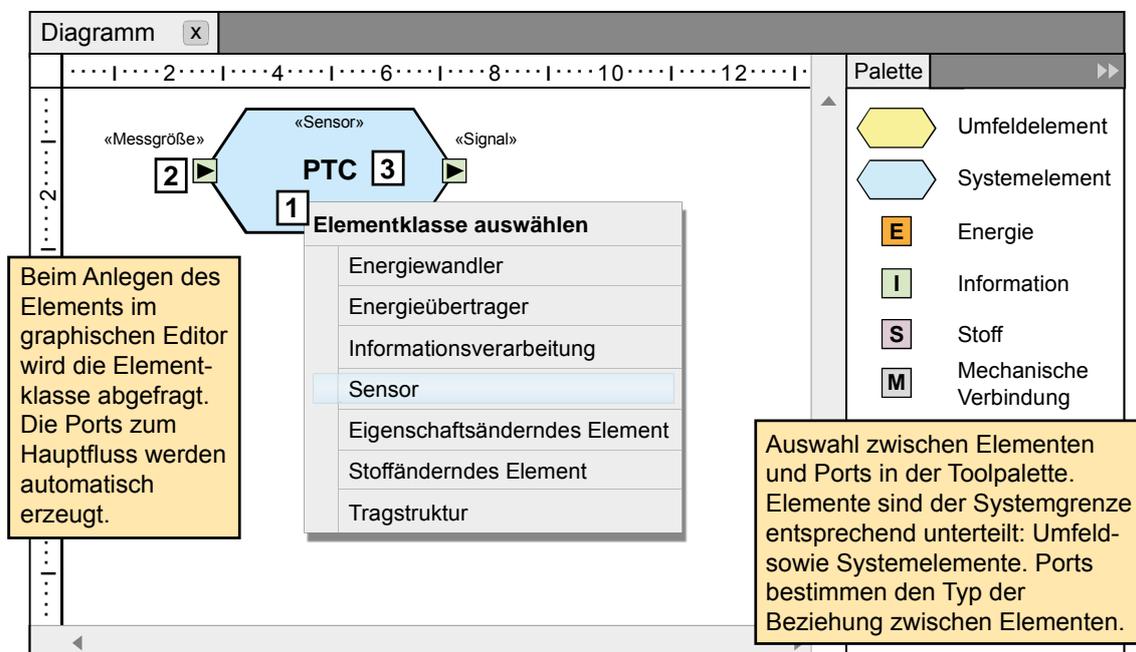


Bild 4-43: Anlegen eines Elements

Nach der Wahl der Klasse werden dem Element automatisch die entsprechenden Ports als Hauptfluss angelegt. Den Ports ist damit die Information hinterlegt, dass sie zum Hauptfluss des Elements gehören. Beispielsweise werden bei der Auswahl *Sensor* die Ports *«Messgröße»* IN und *«Signal»* OUT erzeugt (Bild 4-43, Schritt 2). Der Modellierer vergibt im nächsten Schritt die Bezeichnung des Elements (Schritt 3).

Der Typ der ein- und ausgehenden Beziehung wird nur bei **Informationsumsetzenden Elementen** direkt mit der Wahl der Elementklasse (Sensor oder IV) vergeben. Für **Energieumsetzende** und **Stoffumsetzende** Elemente wird eine weitere Abfrage an den Ports zum Hauptfluss gestartet: Bei **Energieumsetzenden Elementen** werden zu Beginn die Ports vom Typ *«Energie»* automatisch angelegt. Anschließend muss der Modellierer die Energieform wählen. Hierzu wird ihm am Port die Liste der möglichen

Energieformen angezeigt. Beim Wählen der Energieform (eingehende Beziehung) wird diese am Port der ausgehenden Beziehung direkt übernommen, wenn es sich beim Element um einen Energieübertrager handelt. Im anderen Fall (Energiewandler) wird am Port der ausgehenden Beziehung die Abfrage erneut gestartet, mit dem Unterschied, dass an diesem Port die zuvor ausgewählte Energieform ausgegraut ist, da die Bedingung Energieform IN \neq Energieform OUT eingehalten werden muss (Bild 4-44).

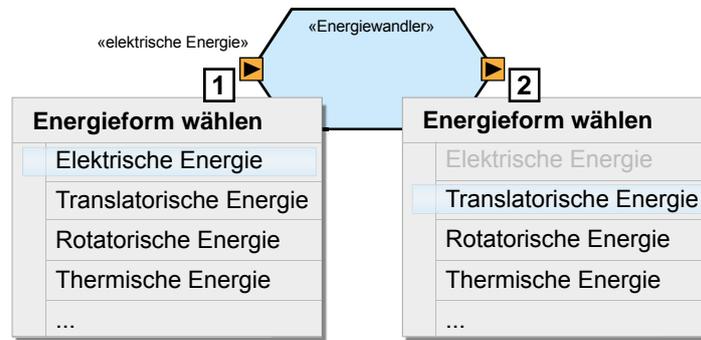


Bild 4-44: Automatische Einhaltung der Bedingung für Energiewandler bei der Wahl der Energieform

Bei **Stoffumsetzenden Elementen** wird neben der Port-Klasse der Stoff abgefragt. Für Eigenschaftsändernde Elemente wird am eingehenden Port der Stoff gewählt. Hierzu wird dem Benutzer die Liste der im Modell spezifizierten Stoff-Objekte angezeigt (Bild 4-45). Bei der Wahl des Objekts wird dieses automatisch am ausgehenden Port erzeugt (Schritt 2, Bild 4-45). Wurde kein Objekt angelegt, kann dies in diesem Schritt nachgeholt werden. Nicht immer ist gewünscht, alle Stoff-Objekte auch als Umfeldelemente graphisch im Modell abzubilden. Daher kann der Benutzer in diesem Schritt entscheiden, ob das Objekt auch graphisch angezeigt werden soll oder nicht.

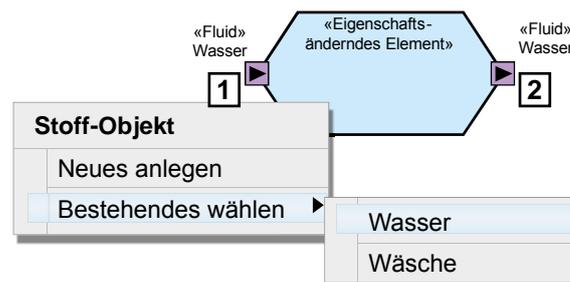


Bild 4-45: Wahl des Stoff-Objekts

Wählt der Benutzer die Klasse **Tragstruktur** aus, wird lediglich ein Port vom Typ «mechanische Verbindung» angelegt. In diesem Fall werden ebenfalls die Unterklassen (Schlussart) abgefragt. Dies kann jedoch abgebrochen werden, so dass der Port die Klasse «mechanische Verbindung» beibehält.

Umfeldelemente haben neben den technischen Elementen die nicht-technischen Elemente in der Auswahlliste. An den Umfeldelementen werden die Ports nicht automatisch generiert.

Beziehung zwischen Elementen ziehen

Sind zwei Elemente vorhanden, kann anschließend die Beziehung zwischen zwei Ports gezogen werden mit der Bedingung, dass die Ports vom gleichen Typ sind (hier «Signal»). Erst wenn die Beziehung angelegt wurde, wird dessen Benennung abgefragt (z.B. *Temperatur*, Bild 4-46). Weitere Ports können an den Elementen erzeugt werden. Dies erfolgt analog zum Anlegen eines Elements. Aus der Toolpalette wird ein Port-Typ gewählt (*Energie, Stoff, Information* oder *mechanische Verbindung*). Erst wenn es im Editor am Element erzeugt wurde, wird die Klassifikation über ein Auswahlménü abgefragt. Sollen nun zwei Elemente verbunden werden, wobei nur ein Port erzeugt wurde, werden die entsprechenden Modellierungsschritte automatisiert. Hierzu zieht der Benutzer die Beziehung ausgehend von einem Port zum zu verbindenden Element. An diesem wird automatisch ein Port vom gleichen Typ wie am Start-Element erzeugt und die Beziehung angelegt.

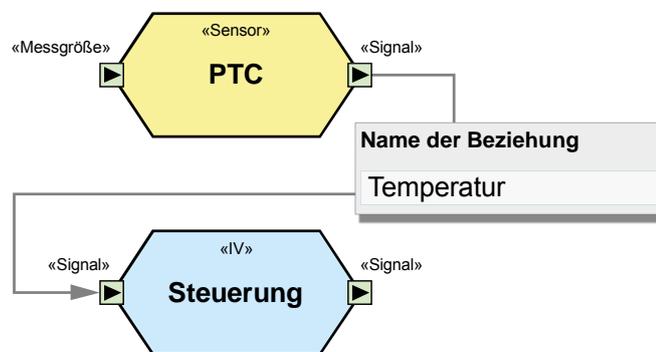


Bild 4-46: Beziehungen zwischen zwei Ports gleichen Typs

Integration der Matrix mit Energieumsetzenden Elementen und Sensoren

Die Modellierung der Energieumsetzenden Elemente und der Sensoren kann vom Werkzeug auch im konstruktiven Sinne unterstützt werden. Hierzu werden die Inhalte der Matrix im Werkzeug hinterlegt. Beim Anlegen eines Energieumsetzenden Elements wird nach der Wahl der eingehenden Beziehung die Matrix aufgerufen. Dem Benutzer werden die Inhalte der Zeile zur Auswahl angezeigt (Bild 4-47). Wird ein Energiewandler spezifiziert, werden die Inhalte der Diagonalen nicht mit angegeben. Im Gegensatz dazu wird beim Energieübertrager nur der Inhalt der Diagonalen angeboten. Der Modellierer wählt das entsprechende Element aus, woraufhin die ausgehende Beziehung automatisch erzeugt wird (Schritt 2 und 3 in Bild 4-47).

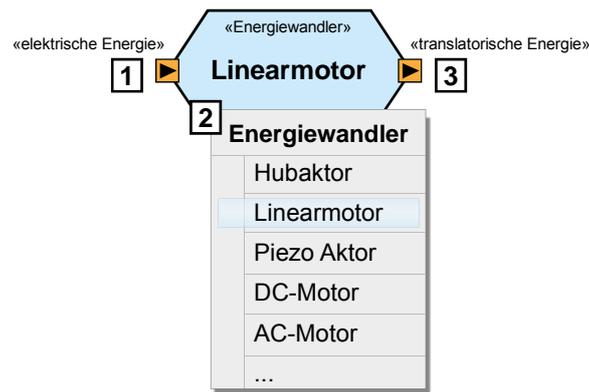


Bild 4-47: Auswahlliste der Energiewandler mit automatischer Typisierung der ausgehenden Beziehung

Ähnliches Vorgehen wird beim Spezifizieren von Sensoren vom Werkzeug angeboten. Hierbei wird jedoch nicht die ausgehende Beziehung automatisch erzeugt, da diese in der Matrix nicht enthalten sind. Dem Element selbst wird die Wirkbeziehung als Information hinterlegt, die bei Bedarf abgefragt und angezeigt werden kann.

Dekomposition von Elementen

Bei der Dekomposition wird die Klasse des Vater-Elements automatisch auf das Kind-Element übertragen. Hierzu wählt der Modellierer ein Element aus, das er dekomponieren möchte. Es wird eine neue Ebene erstellt mit einem Element, das die Klasse erbt. Dieses Element muss vom Modellierer als erstes weiter durch die Angabe der Bezeichnung spezifiziert werden. Danach können weitere Elemente hinzugefügt und Beziehungen gezogen werden. Die Ausnahme für Eigenschaftsändernde Elemente wird vom Werkzeug berücksichtigt. In diesem Fall wird die Ebene leer gelassen.

Filterfunktion

Die Information über die Klassen der Elemente und Beziehungen sind im Modell enthalten. Das Werkzeug bietet eine Maske an, die das Erstellen von spezifischen Filtern ermöglicht. Neben den Beziehungen können auch weitere Informationen über die Elemente (Klasse, Attribute) als Auswahlkriterium genutzt werden. Das Werkzeug lässt auch Kombinationen aus Klassifikationsmerkmalen zu. Die damit generierten Filter können auf die graphischen Modelle angewandt werden. Hierzu wird über Hierarchieebenen hinweg die Sicht generiert, in der Elemente und Beziehungen hervorgehoben werden, die zum Filter gehören, und andere Modellkonstrukte in den Hintergrund gesetzt werden.

Nicht graphische Modellierung von Beziehungen

In einer Systemstruktur ist eine Vielzahl an Beziehungen zwischen den Elementen vorhanden. Diese vollständig rein graphisch zu spezifizieren, kann aufwendig sein – vor allem dann, wenn die Beziehung über unterschiedliche Hierarchieebenen der Elemente gehen. In diesem Fall können Matrizen genutzt werden. Die Zeilen und Spalten der

Matrizen werden dabei automatisch vom Werkzeug generiert, je nachdem welche Beziehung gesetzt werden soll. Beispielsweise kann die mechanische Verbindung mit der Beziehungsmatrix (Kapitel 4.3.1) erstellt werden. Hierzu werden alle gestaltbehafteten Elemente des Modells identifiziert und in einer Matrix aufgeführt. Da es sich um eine an der Diagonalen gespiegelten Matrix handelt, wird der gespiegelte Eintrag automatisch übernommen.

Der Benutzer kann sich in dem graphischen Editor die Beziehungen aus der Matrix heraus generieren lassen. Dabei ist auf eine gute Layoutgestaltung der Verbindungslinien zu achten, um die Lesbarkeit des graphischen Modells sicherzustellen.

Analog hierzu kann eine Matrix generiert werden, die alle aktiven Elemente gegenüber den Energieumsetzenden Elementen stellt. Diese können die aktiven Elemente mit Energie versorgen. Die Beziehung wird in der Matrix hinterlegt (Bild 4-48). Die Matrix ist nicht quadratisch, da es mehr aktive Elemente geben kann als Energieumsetzende Elemente. Da Elemente sich nicht selbst mit Energie versorgen können, sind die Kreuzungspunkte, die Energieumsetzende Elemente sich selbst gegenüberstellen, deaktiviert.

Beziehungsmatrix		aktive Elemente	SE 1	SE 2	SE 3	UE 1	SE 4	SE 5
Fragestellung: „Versorgt das Element (Zeile) das Element (Spalte) mit Energie?“ X = ja								
energieumsetzende Elemente		Nr.	1	2	3	4	5	6
Systemelement SE1		1		X			X	
Systemelement SE2		2				X		
Systemelement SE3		3						X
Umfeldelement UE1		4			X			X

Bild 4-48: Beziehungsmatrix der elektrischen Energie

4.5.1.2 Automatische Plausibilitätsprüfung

Nach der Erstellung des Modells erfolgt die automatische Plausibilitätsprüfung. Hierzu werden die Bedingungen aus Kapitel 4.2 überprüft. Einige Bedingungen sind bereits erfüllt, da diese im Werkzeug zur Automatisierung von Modellierungsschritten genutzt werden. Die Überprüfung erfolgt nur anhand der Bedingungen, da sich die Richtlinien auf die Semantik der Elemente und Beziehungen beziehen und somit nicht rechnerunterstützt ausgewertet werden können.

Für die **Vergleichbarkeit** der Systemstruktur muss die Bedingung „*Jedes Element und jede Beziehung ist einer Klasse zuzuordnen*“ eingehalten werden. Dieses erfolgt bereits beim Erstellen der Elemente und Beziehungen, da dabei die Klassen abgefragt werden.

Die Prüfung der **Vollständigkeit** wird für jede Bedingung einzeln erläutert.

Elemente und Beziehungen müssen benannt werden: Der Name des Elements oder der Beziehung wird beim Erstellen abgefragt. Durch die Überprüfung wird dabei sichergestellt, dass das Feld für den Namen nicht leer gelassen wurde. Trifft dies zu, wird der Modellierer hingewiesen, das entsprechende Element oder die Beziehung zu benennen.

Energieumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Energie»: Beim Anlegen eines Elements werden die Ports automatisch erstellt, die zum Hauptfluss gehören. Damit wird nicht sichergestellt, dass auch eine entsprechende Beziehung vorhanden ist. Erst die Überprüfung zu dieser Bedingung liefert einen Hinweis auf fehlende Beziehungen. Dies erfolgt für alle technischen Elemente, die innerhalb der Systemgrenze liegen.

Alle Elemente und Beziehungen sind einer Klasse der höchsten Detailstufe zuzuordnen: Im Werkzeug wird beim Anlegen der Elemente die höchste Detailstufe der Klassen abgefragt, so dass die Bedingung erfüllt ist. Bei den Beziehungen sind beide Detailstufen zugelassen. Beim Anlegen eines Ports wird im ersten Schritt die unterste Detailstufe angeboten («Energie», «Information», «Stoff» und «mechanische Verbindung»). Erst im zweiten Schritt wird diese Unterteilung detailliert. Als Ergebnis der Überprüfung werden dem Anwender die Beziehungen aufgelistet, die noch nicht die höchste Detailstufe der Klassifikation besitzen.

Es besteht ein 1:m-Verhältnis, mit $m > 1$, zwischen dem Element der Ebene n und den Elementen der Ebene $n+1$: Ist diese Bedingung nicht erfüllt, wird der Modellierer darauf hingewiesen. Der Anwender muss dann entscheiden, ob ein Element nicht modelliert wurde, oder ob die eingeführte Ebene überflüssig ist.

Mindestens ein Element der Ebene $n+1$ erbt die Elementklasse des Vater-Elements. Die Ausnahme bildet die Elementklasse «Eigenschaftsänderndes Element»: Diese Bedingung wird beim Dekomponieren von Elementen berücksichtigt und ist damit erfüllt.

Gewollte Wirkbeziehungen vom Typ «Energie», «Information» oder «mechanische Verbindung» müssen bis zur untersten Hierarchieebene geführt werden: Beziehungen zwischen Elementen, die sich auf unterschiedlichen Hierarchieebenen befinden, werden über die Hierarchien weitergereicht. Fehlende Beziehungen werden mit dieser Bedingung identifiziert (Bild 4-49).

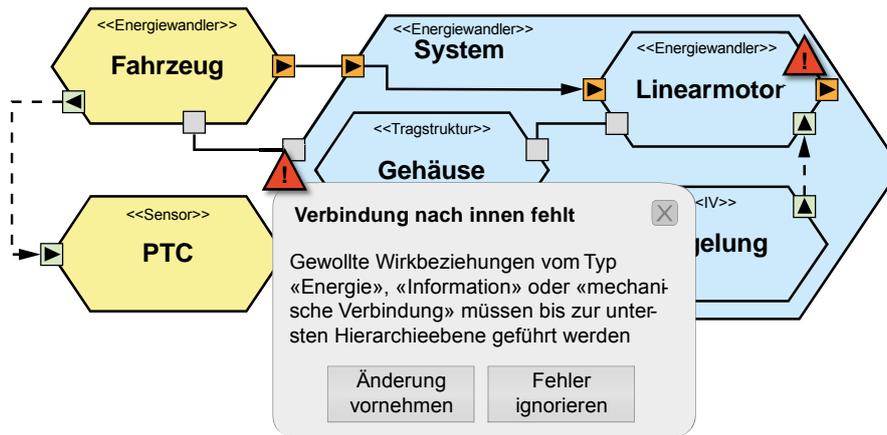


Bild 4-49: Prüfergebnis wird mit einem Warnhinweis angezeigt

In der Systemstruktur ist mindestens eine Energiequelle vorhanden; Elemente mit dem Attribut „aktiv“ besitzen mindestens eine eingehende Beziehung vom Typ «elektrische Energie»; Elemente mit dem Attribut „gestaltbehaftet“ haben mindestens eine Beziehung vom Typ «mechanische Verbindung»: Zur Überprüfung dieser Bedingungen müssen Elemente mit den Attributen *Energiequelle*, *aktiv* und *gestaltbehaftet* versehen werden. Damit wird im ersten Schritt geprüft, ob Elemente vorhanden sind, die mit den Attributen versehen sind. Trifft dies nicht zu, wird der Anwender hingewiesen, die Attribute zu vergeben. Im zweiten Schritt erfolgt die Überprüfung der Bedingungen.

Wird der Zustand eines Stoffes durch ein Eigenschaftsänderndes Element geändert, besitzt das Element mindestens eine eingehende Beziehung vom Typ «Energie»: Für diese Bedingung wird die Spezifikation des Stoffflusses untersucht. Durch den Vergleich der ein- und ausgehenden Beziehung am Eigenschaftsändernden Element kann auf die Zustandsänderung geschlossen werden. In diesem Fall greift die Bedingung und wird auf ihre Einhaltung überprüft.

In der Systemstruktur ist mindestens eine Beziehung vom System zu einem Umfeldelement vom Typ «mechanische Verbindung» vorhanden: Wird diese Bedingung verletzt, erhält der Anwender den Hinweis und muss entscheiden, ob eine Beziehung fehlt, oder ob die Bedingung für das betrachtete System nicht zutrifft.

Die **Richtigkeit** bezieht sich auf die richtige Modellierung der Zusammenhänge zwischen der ein- und ausgehenden Beziehung von Elementen. Die in Kapitel 4.2 aufgeführten Bedingungen werden für jedes Element überprüft. Bei den Energieumsetzenden Elementen wird in diesem Zusammenhang die im Werkzeug integrierte Matrix mit Energieumsetzenden Elementen eingesetzt. Damit wird die Überprüfung aus Kapitel 4.3 automatisiert. Der Kreuzungspunkt der Matrix ergibt sich aus den ein- und ausgehenden Beziehungen am Energieumsetzenden Element. Daraufhin erfolgt ein Abgleich des Elements mit den Elementen im Kreuzungspunkt. Stimmt keines der Matrixeinträge mit dem spezifizierten Element überein, wird dem Modellierer ein Hinweis gegeben und die Liste der Elemente angezeigt. Der Modellierer muss in diesem Fall entscheiden, ob er

lediglich eine andere Bezeichnung seines Elements gewählt hat oder die ein- und/oder ausgehende Beziehung falsch spezifiziert wurde. Da die Matrix keinen Anspruch auf Vollständigkeit hat, kann es vorkommen, dass das Element richtig spezifiziert wurde, jedoch nicht in der Matrix enthalten ist. Der Modellierer erhält die Möglichkeit, die Matrix unternehmensspezifisch auszuprägen und damit Elemente einzufügen oder zu entfernen. Die Matrix kann auch mit Lösungselementen gefüllt sein.

4.5.2 Prototypische Implementierung der Plausibilitätsprüfung

Die Überprüfung der Richtigkeit in Kombination mit der Matrix für energiemetzende Elemente wurde sowohl im Mechatronic Modeller als auch in einem SysML-Werkzeug prototypisch implementiert. Damit wird aufgezeigt, dass die Einführung der Klassifikation und die Auswertung der Systemstruktur sowohl mit CONSENS als auch SysML möglich sind.

Mechatronic Modeller

Das dedizierte Werkzeug Mechatronic Modeller wurde für die Sprache CONSENS entwickelt (Kapitel 3.1.2) [GLL12]. Das Metamodell sieht dabei die Unterscheidung der Elemente in Umfeld- und Systemelement vor. Bei den Beziehungen wird zwischen Energie, Information und Stoff unterschieden. Zusätzlich ist die logische Beziehung vorhanden, die im Folgenden für die mechanische Verbindung genutzt wird.

Im Werkzeug besteht die Möglichkeit projektweit Klassifikationsmerkmale zu definieren. Diese werden genutzt, um die Elementklassen und Beziehungsklassen zu definieren. Bei der Erstellung der Wirkstruktur können die Elemente und Beziehungen mit den entsprechenden Klassifikationsmerkmalen versehen werden. Bild 4-50 zeigt die Klassifikation des Elements *Akku* im Mechatronic Modeller.

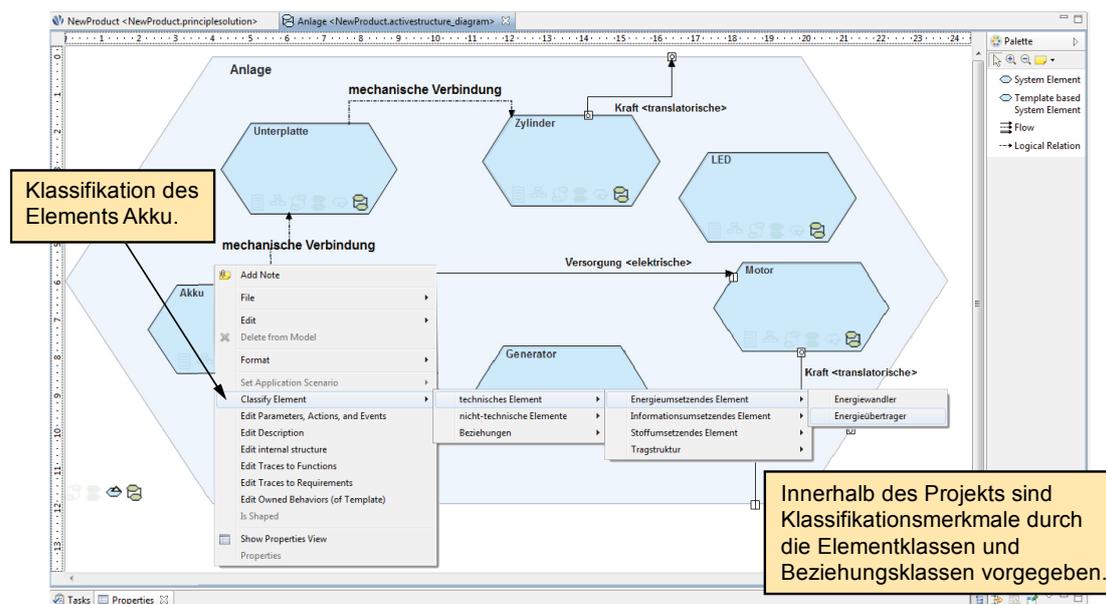


Bild 4-50: Klassifikation der Elemente und Beziehungen im Mechatronic Modeller

Die Modellierung der Systemstruktur erfolgt im Mechatronic Modeller. Anschließend werden die Modelldaten ausgelesen und ausgewertet. Die Ergebnisse der Auswertung der Energieumsetzenden Elemente sind in der Excel-Tabelle aufgezeigt (Bild 4-51). Die **Systemelemente** werden untereinander aufgelistet, die mit dem Klassifikationsmerkmal Energiewandler oder Energieübertrager versehen waren. Die **Elementklasse** sowie alle ein- und ausgehenden **Energiebeziehungen** sind aufgeführt. Dabei werden die eingehenden Beziehungen hellgrau und die ausgehenden Beziehungen dunkelgrau markiert dargestellt.

Systemelement	Energiebeziehung (Input/ Output)				Elementklasse (Link)	Hinweise
	elektrische	translatorische				
Motor	elektrische	translatorische			Energiewandler	
Zylinder	translatorische				Energiewandler	Input fehlt
LED					Energiewandler	Input und Output fehlen
Generator	elektrische	elektrische			Energiewandler	Bedingung Energietyp IN ≠ Energietyp OUT nicht erfüllt
Akku	elektrische	elektrische	elektrische		Energieübertrager	

		Energie			
		Translatorisch	Rotatorisch	Elektrisch	Pneumatisch
Energie	OUT				
	IN				
	Translatorisch	Hebel, Feder, Dämpfer, Hydraulik-Pressen, Keil	Zahnradstange	Piezo	Kolbenpumpe, Luftpumpe
	Rotatorisch	Gewindespindel, Nockengetriebe, Zahnradstange	Gebtriebe, Welle, Rad, Zahnrad	Generator	Verdichter
Elektrisch	Hubaktor, Linearmotor, Piezo, Zylinder	Rotationsmotor, DC-Motor, AC-Motor	Akkumulator, Leistungselektronik, Netzteil	Pneumatikpumpe	
Pneumatisch	Zylinder	Turbine, Windrad	Pneumatikgenerator	Akkumulator, Leistungselektronik, Netzteil	

Bild 4-51: Aus den Modelldaten des Mechatronic Modellers abgeleitete Tabelle mit Analyseergebnis und Verlinkung zur Matrix

In der Elementklasse ist die Matrix mit Energieumsetzenden Elementen hinterlegt. Dabei handelt es sich um einen Link, der auf Basis der ein- und ausgehenden Beziehungen zum entsprechenden Kreuzungspunkt in der Matrix führt. Konnte keine Verlinkung hergestellt werden, so ist die Elementklasse rot markiert. Der Grund für den fehlenden Link kann fehlende oder fehlerhafte Modellierung der ein- oder ausgehenden Beziehung sein und wird unter **Hinweis** aufgeführt. Bei fehlerhafter Modellierung wird die Bedingungen der Elementklasse (im Beispiel Energiewandler) Energietyp IN ≠ Energietyp OUT nicht erfüllt.

SysML-Werkzeug

Mit dem Profilmechanismus der SysML können die Klassen der Elemente und Beziehungen als leichtgewichtige Spracherweiterungen eingeführt werden. Dies wird durch Stereotype realisiert. Die Elementklassen bilden dabei die Spezialisierung des

SysML1.3::block Stereotyps (Bild 4-52). Für die Beziehungen wird das SysML Stereotyp SysML1.3::InterfaceBlock spezialisiert (Bild 4-53).

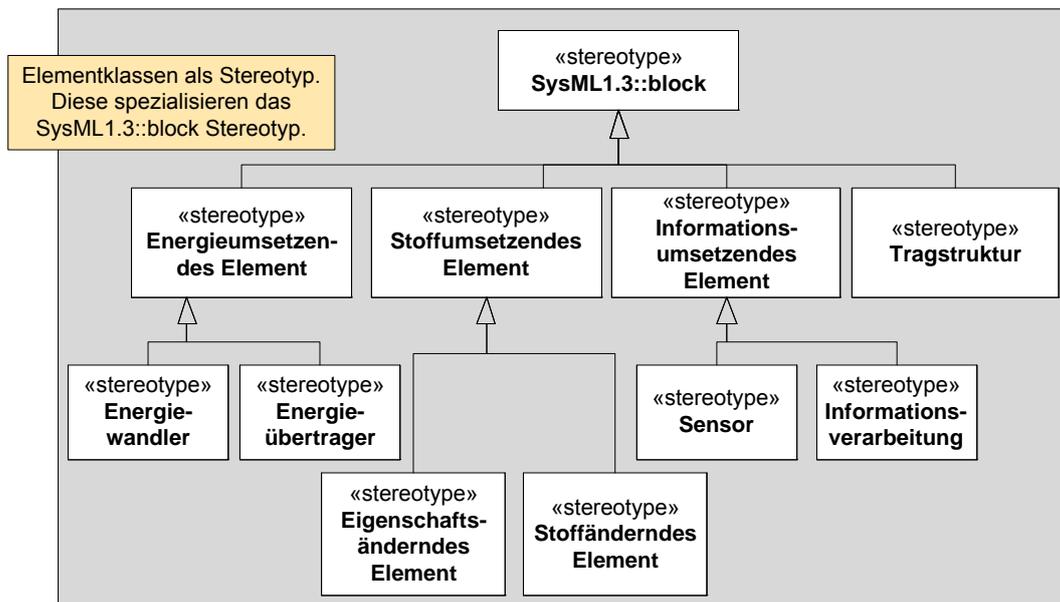


Bild 4-52: Elementklasse als Stereotyp in SysML (abstrakte Syntax)

Die Bedingungen an die ein- und ausgehenden Beziehungen von Elementen bilden die statische Semantik und können formalisiert im Werkzeug abgebildet werden. Zusätzlich wurden die Inhalte der Matrix mit Energieumsetzenden Elementen im Werkzeug hinterlegt. Am Beispiel der Elementklasse «Energiewandler» wird die implementierte Prüfung beschrieben.

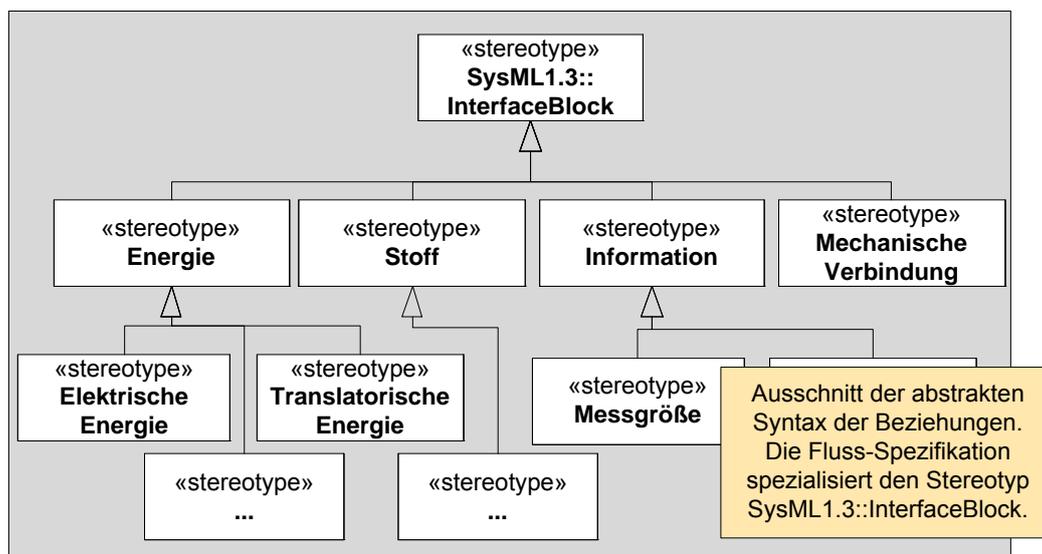


Bild 4-53: Beziehungen als Stereotyp in SysML (abstrakte Syntax)

Im ersten Schritt wird für jedes Element mit dem Stereotyp Energiewandler überprüft, ob dieses mindestens zwei Ports mit unterschiedlichen Richtungen (In und Out) besitzt.

Anschließend wird die Spezifikation des Ports näher betrachtet. Diese enthält die Information über den Stereotyp der Beziehung. Für den Energiewandler werden daraufhin die Stereotype der In- und Out-Ports verglichen. Diese müssen die Bedingung Stereotyp IN \neq Stereotyp OUT erfüllen. Trifft dies zu, wird der Kreuzungspunkt der Matrix identifiziert. Es folgt ein Abgleich des Elementtyps mit den Einträgen im Kreuzungspunkt. Stimmt das Element nicht mit den Einträgen überein, wird ein Hinweis generiert und der Anwender kann über ein Auswahlm Menü die Matrix-Einträge aufrufen (Bild 4-54).

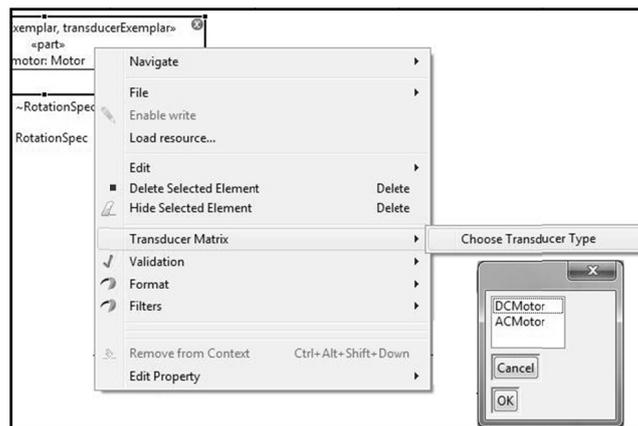


Bild 4-54: Matrixeinträge des Kreuzungspunkts werden in einem Auswahlm Menü angezeigt [KDH+13].

4.6 Vorgehensmodell zur Erstellung plausibler Systemstruktur

Das Vorgehensmodell beschreibt das Vorgehen zur Erstellung einer plausiblen Systemstruktur. Ziel der Systemstruktur ist das einheitliche Systemverständnis sowie eine disziplinübergreifende Diskussions- und Dokumentationsplattform. Die Systemstruktur ist ein Aspekt des Systemmodells. Dieses enthält neben der Strukturbeschreibung die Aspekte Anforderungen und Verhalten. In welcher Reihenfolge und Ausprägung die Aspekte entstehen, beschreibt die Modellierungsmethode. Das Vorgehensmodell zur Erstellung einer plausiblen Systemstruktur ordnet sich in die Modellierungsmethode ein und setzt auf den bereits erstellten Aspekten auf. Als Ausgangsbasis werden für die Strukturbeschreibung eine Anforderungsanalyse sowie eine lösungsneutrale Funktionsbeschreibung vorausgesetzt.

Das Vorgehen zur Erstellung einer plausiblen Systemstruktur erfolgt in fünf Phasen. Bild 4-55 zeigt diese mit den wesentlichen Tätigkeiten und den Resultaten in Form eines Phasen-/Meilenstein-Diagramms. Bei der Reihenfolge handelt es sich um eine idealtypische sequentielle Darstellung, die in der Praxis angepasst werden kann und damit auch Iterationen und Rücksprünge zulässt. Das Vorgehen sieht eine Trennung der Erarbeitung der Systemstruktur und der formalisierten rechnerinternen Abbildung vor.

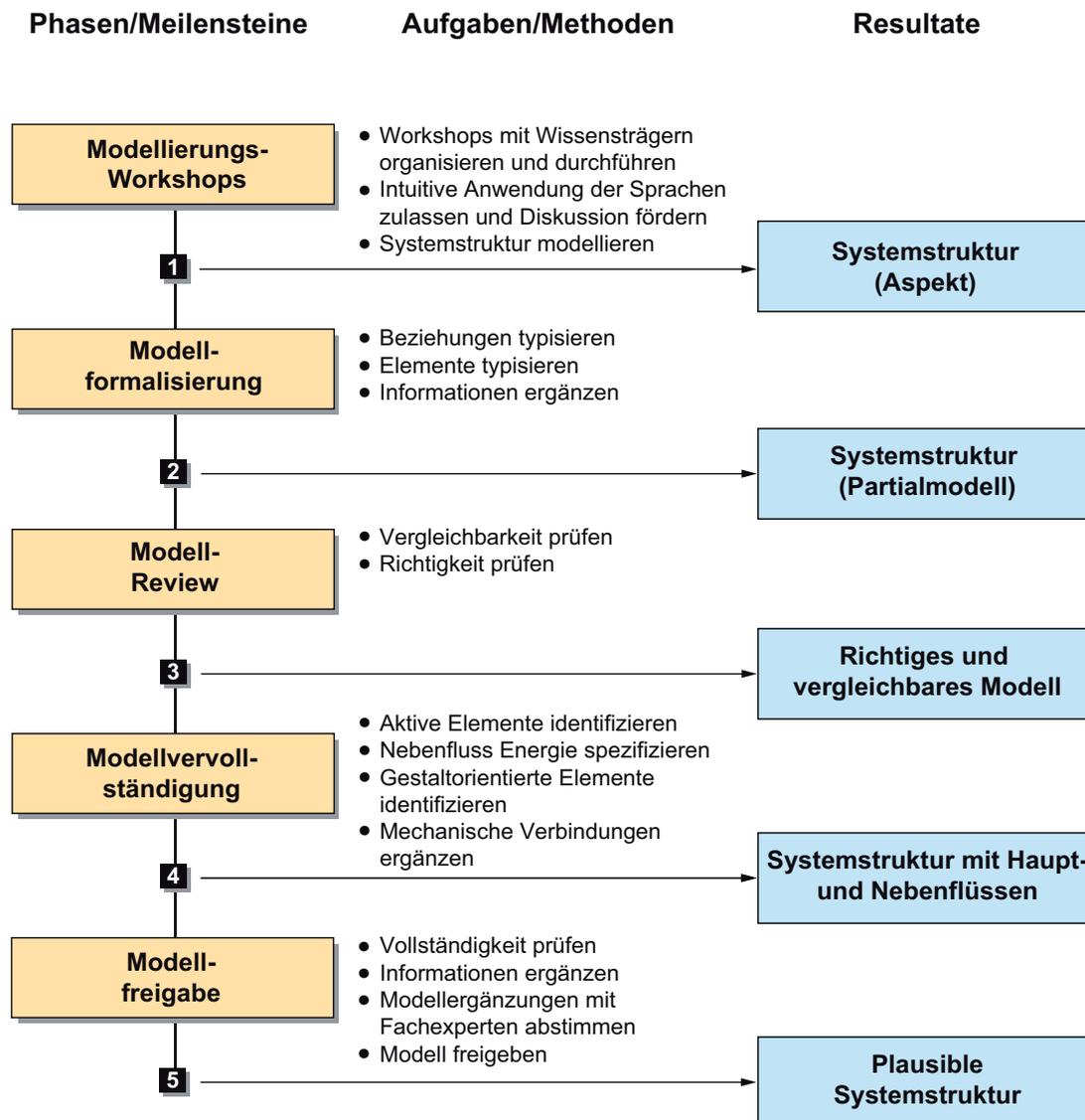


Bild 4-55: Vorgehensmodell zur Erstellung plausibler Systemstruktur

4.6.1 Phase 1: Modellierungs-Workshops

Innerhalb eines Projekts sind verschiedene Fachexperten beteiligt. Je nach Produkt und Projektgröße variiert die Anzahl an Fachexperten und die Vielfalt der Bereiche. Dabei stellen diese die Wissensträger des Projekts dar. Jeder der Fachexperten kennt das Produkt aus seiner Fachexpertise heraus am besten – er kennt die Restriktionen und die Best Practices der vorherigen Projekte. Mit der Systemmodellierung soll erreicht werden, die einzelnen Bereiche und Erfahrungen zu nutzen, um möglichst viele Erkenntnisse über das neue Projekt/Produkt frühzeitig zu berücksichtigen und Fehler zu vermeiden. Hierzu soll gemeinsam die Systemstruktur erarbeitet werden. Das gemeinsame Erarbeiten hat darüber hinaus den Vorteil, dass die Akzeptanz der Vorgehensweise und des Produktkonzepts steigt, wenn die Experten von Beginn an in den Prozess integriert werden [SMM+12].

Die Workshops werden von dem Systems Engineer¹⁴ organisiert. Hierzu zählt die Wahl der Wissensträger (Fachdisziplinen und bei Bedarf andere Organisationseinheiten, wie z.B. Einkauf, Produktion). Der Systems Engineer erläutert das Ziel der Modellierung und schult die Teilnehmer in der Sprache und Methode. Innerhalb der Workshops fördert er die Diskussion rund um das technische System und leitet an, das Modellierungsziel zu erreichen.

Durch die interdisziplinäre Vorgehensweise in der frühen Phase sehen sich die Entwickler vor einer Herausforderung gestellt: neue Modellierungssprache und Methode. Um dieser Herausforderung zu begegnen, wird die Modellierungssprache soweit vereinfacht, dass diese leicht zu erlernen ist, jedoch die notwendigen Informationen über die Systemstruktur liefert. Der Systems Engineer kennt die Richtlinien und Bedingungen zur Modellierung der Systemstruktur und achtet darauf, dass diese eingehalten werden. Die Teilnehmer erlernen die Richtlinien indirekt durch das Anwenden. Eine explizite Schulung ist nicht notwendig, da die Modellierung in Moderation durch den Systems Engineer erfolgt. Folgende Modellelemente sind auszuwählen, wobei zu beachten ist, dass die graphische Notation je Konstrukt eindeutig ist.

- Umfeldelemente (Elemente außerhalb der Systemgrenze)
- Systemelemente (Elemente innerhalb der Systemgrenze)
- Beziehungen: Stoff, Information, Energie, mechanische Verbindung

Die überschaubare Anzahl an Modellkonstrukten bietet den Entwicklern eine intuitive Herangehensweise. Durch die graphische Unterscheidung der Elemente wird die Darstellung anschaulich und gut lesbar. Die Workshops sind so organisiert, dass die Fachexperten aktiv an der Modellierung beteiligt sind. Die Modellkonstrukte dienen dabei als Ausdrucks- und Dokumentationsmittel. Die Modellierung erfolgt auf Papier – z.B. auf einer Moderationswand. Damit werden die Fachexperten nicht mit Tools konfrontiert, sondern können sich auf das technische System und die Systemstrukturbeschreibung konzentrieren.

Innerhalb der Workshops können viele disziplinübergreifende Fragestellungen erarbeitet und diskutiert werden – gelöst werden diese jedoch nicht auf der Abstraktionsebene der Systemstruktur. Die Lösung der Entwicklungsarbeit liegt weiterhin im Detail der entsprechenden disziplinspezifischen Modelle. Ein mehrfaches Durchlaufen der Workshops ist daher ein notwendiger Prozess. Damit die Fragestellungen aus den Workshops innerhalb der Disziplinen weiter genutzt werden, bietet sich hierbei die Moderationstechnik „Themenspeicher“ an. Auf einer separaten Wand werden aufgekommene Frage-

¹⁴ Durch das Paradigma des Model-Based Systems Engineering erweitert sich der Aufgabenbereich eines Systems Engineers. Eine Beschreibung der Aufgaben und des Fähigkeitsprofils eines Systems Engineers befindet sich im Anhang (A5).

stellungen mit Verantwortlichkeiten gesammelt. Haben diese Themen eine Auswirkung auf die Systemstruktur werden sie im Verlauf der Workshop-Reihe wieder aufgegriffen.

Die Modellierung der Systemstruktur erfolgt hierarchisch. Im ersten Schritt wird das System als Black-Box betrachtet. Hierzu muss eine Systemgrenze definiert werden. Dies erfolgt durch die Zuordnung der Elemente zu Systemelementen oder Umfeldelementen. Ziel der Black-Box Betrachtung ist eine Umfeldanalyse. Dabei werden die Wechselwirkungen des Systems mit den Umfeldelementen identifiziert und visualisiert. Die technischen Wirkbeziehungen stehen im Fokus. Damit werden Einflüsse auf die Entwicklung, wie Normen oder Gesetze, nicht als Elemente modelliert. Diese Punkte nimmt der Systems Engineer in den Themenspeicher auf und überführt diese anschließend in die Anforderungsliste. Einflüsse aus der Umgebung, die das System in seiner Wirkungsweise beeinträchtigen können, wie z.B. Temperaturschwankungen, Vibrationen, werden abgebildet. Der Systems Engineer lässt sich die Systemumgebung von den Teilnehmern schildern und benennen. Dieses wird durch Umfeldelemente modelliert, wobei die Einflüsse als Wirkbeziehungen dargestellt werden.

Die Black-Box Betrachtung liefert Aufschluss über die Wirkzusammenhänge und Bedingungen aus der Umgebung. Dabei werden viele Fragestellungen identifiziert, deren Ausprägung erst durch die White-Box Betrachtung möglich wird. Innerhalb der White-Box Betrachtung wird die nächste Hierarchieebenen betrachtet. Hierzu sind Funktionseinheiten zu identifizieren und zu benennen. Dabei wird empfohlen sich an der Funktionsbeschreibung zu orientieren. Diese enthält bereits eine Hierarchie (unabhängig davon, ob eine reine Funktionshierarchie oder Funktionsstruktur erstellt wurde). Die neue Ebene wird anschließend mit den Beziehungen aus der Umgebung verbunden. Da die Modelle auf unterschiedlichen Metaplanwänden entstehen, werden die Beziehungen in der Black-Box Ebene durchnummeriert und anschließend in der neuen Ebene mit den Systemelementen verbunden. Dieses Vorgehen wird für jede Ebene wiederholt. Der Systems Engineer muss auf Basis der Diskussionen entscheiden, wie viele Hierarchieebenen betrachtet werden sollen. Dabei darf nicht ausgeschlossen werden, dass neue Ebenen im Verlauf der Entwicklung hinzukommen.

4.6.2 Phase 2: Formalisierung des Modells

Das gemeinsam erstellte Modell wird im zweiten Schritt vom Systems Engineer formalisiert, indem es rechnerintern abgebildet wird. Hierzu muss eine Wahl des Modellierungswerkzeugs getroffen werden. Der Systems Engineer kennt die Unterschiede der Werkzeuge sowie die IT-Struktur des Unternehmens. Auf dieser Basis wählt er das entsprechende Werkzeug aus. Je nach Werkzeug ist dieser Prozess auf die toolspezifischen Besonderheiten anzupassen. Zur Formalisierung des Modells werden folgende Teilschritte vorgenommen:

Umfeldelemente klassifizieren – Die Umfeldelemente werden nach nicht-technischen und technischen Elementen unterschieden. Die nicht-technischen Elemente werden den

Klassen «Lebewesen», «Umgebung» oder «Stoff-Objekt» zugeordnet (Kap. 4.1.3). Bei den technischen Elementen sind für jedes Element die Hauptfunktion und damit der Hauptfluss zu bestimmen, so dass eine entsprechende Klassifikation vorgenommen werden kann (Kap. 4.1.2). Die «Energieübertrager» erhalten zusätzlich das Attribut *Energiequelle*, falls sie das System mit der Energie versorgen.

Beziehungen klassifizieren – Die Teilnehmer der Modellierungswshops haben die Beziehungen beliebig bezeichnet. Entscheidend für den Workshop war die Trennung in die Grundtypen (Energie, Stoff, Information und mechanische Verbindung). In diesem Schritt werden alle Beziehungen weiter unterteilt. Für die Energiebeziehungen wird die entsprechende Energieform gewählt und die Beziehung dementsprechend typisiert. Wurde beispielsweise eine Energiebeziehung von einer Energiequelle zum System gezogen mit der Bezeichnung *Versorgungsstrom*, so handelt es sich um elektrische Energie. Die Beziehung erhält die Klassifikation «elektrische Energie», sowie die bereits bestehende Bezeichnung *Versorgungsstrom*.

Bei der Stoffbeziehung ist zu unterscheiden, ob es sich um ein Fluid oder Festkörper handelt. Im Modell wird hierzu ein Objekt angelegt, das mit Name, Merkmal und Merkmalsausprägung definiert wird. Die Informationsbeziehungen, die eine Messgrößenerfassung darstellen, sind als solche zu modellieren. Die Kommunikationsbeziehung zwischen technischen Elementen wird mit dem Typ «Signal» angelegt. Mechanische Verbindungen können, soweit bekannt, mit der Angabe der Schlussart hinterlegt werden. Bei der mechanischen Verbindung können dabei mehr als eine Schlussart gewählt werden (z.B. Formschluss und Stoffschluss).

Systemelemente klassifizieren – Bei den Systemelementen handelt es sich um technische Elemente, die entsprechend klassifiziert werden. Den Elementen werden neben der Klasse die Attribute *aktiv* und/oder *gestaltbehaftet* vergeben. Der Hauptfluss kann zur besseren Erkennung hervorgehoben dargestellt werden.

4.6.3 Phase 3: Modell-Review

Durch die Angabe der Klassen für Elemente und Beziehungen führt der Systems Engineer eine erste Überprüfung des Modells durch. Anschließend kann das Modell auf Basis der Checkliste (Kap. 4.3) systematisch überprüft werden. Eine rechnerunterstützte Überprüfung hängt vom gewählten Tool und der implementierten Prüfungen ab. In diesem Schritt erfolgt die Überprüfung des Modells auf Vergleichbarkeit und Richtigkeit. Dies kann auf das bereits vorliegende Modell angewandt werden. Die Vollständigkeit sollte nach dem Schritt Modellvervollständigung überprüft werden.

Die Richtigkeit der Energieumsetzenden Elemente kann durch den Einsatz der Matrix (Kap. 4.3.1) erfolgen. Im Fall einer Überprüfung, werden die Energieumsetzenden Elemente schrittweise abgearbeitet. Hierzu wählt der Benutzer das Element aus und betrachtet die eingehende und ausgehende Beziehung. Diese definieren den Kreuzungs-

punkt in der Matrix. Der Systems Engineer schaut in der Matrix im entsprechenden Kreuzungspunkt die Elemente nach und entscheidet, ob das dargestellte Element dem Element der Matrix entspricht. Im Fall der Übereinstimmung, wurde das Element in der Systemstruktur richtig spezifiziert und er wählt das nächste Element zur Überprüfung aus. Trifft die Übereinstimmung nicht zu, muss der Systems Engineer abwägen, ob die Darstellung der Beziehungen nicht richtig angegeben ist, oder das Element in der Matrix noch nicht vorhanden ist. Fallspezifisch muss er die Anpassungen tätigen oder sich mit den entsprechenden Fachexperten abstimmen.

4.6.4 Phase 4: Modellvervollständigung

Bei der Erstellung der Systemstruktur hängt es stark von den Teilnehmern ab, welche Beziehungen modelliert werden. In den meisten Fällen achten die Beteiligten darauf, dass der Hauptfluss vorhanden ist. Nebenflüsse, wie z.B. elektrische Versorgung werden nicht immer dargestellt. Diese Informationen gehören jedoch ebenfalls in die Systemstruktur, da sie Schnittstellen der Elemente darstellen.

Die Modellvervollständigung kann vom Systems Engineer vorgenommen werden. Bei Unklarheiten sollten die Fachexperten wiederum mit einbezogen werden. Alle Änderungen müssen innerhalb der Projektgruppe weitergetragen werden. Hierzu bieten sich wiederum Workshops an. Zu diesen Treffen muss vorab eine Einführung und Erläuterung des ausgewählten Tools stattfinden. Bislang war den Teilnehmern nur die Papierversion der Systemstruktur bekannt. Zweck der Einführung ist nicht die Schulung der Teilnehmer in der Bedienung des Tools. Jedoch sollte jeder wissen, welches Tool verwendet wird, warum und wie die Modelle in dem Tool dargestellt und damit zu lesen sind.

Die Systemstruktur wird bereits für kleinere Systeme komplex und ist in der Darstellung nicht gut zu überblicken. In diesem Fall sollte zur Durchführung des gemeinsamen Reviews das Konzept der Filter genutzt werden (Kap. 4.4). Der Systems Engineer hat das Modell bereits formalisiert rechnerintern abgebildet und kann gezielt die Sichten generieren lassen. Dazu ist abzuwägen, welche Sicht für die angesprochenen Fachexperten sinnvoll ist. Zum Beispiel ist für ein Review mit der Konstruktion/Mechanik die gestaltorientierte Sicht zu wählen.

In dem gemeinsamen Review und der Vervollständigung werden aktive Elemente identifiziert. Aktive Elemente benötigen eine eingehende Beziehung vom Typ «elektrische Energie». Im Rahmen der Workshops werden fehlende Beziehungen ergänzt.

Tragstrukturen, wie Gehäuse, sind meist passive Elemente und werden teilweise in der Modellierung vernachlässigt. Im Rahmen der Vervollständigung ist daher ein Fokus auf die konstruktiven Elemente zu legen und bei Bedarf zu ergänzen. Damit geht die Verknüpfung der Systemelemente durch die Beziehung «mechanische Verbindung» einher.

4.6.5 Phase 5: Modellfreigabe

Im letzten Schritt erfolgt ein zweites Review des Modells. Hierzu wird die Checkliste erneut durchlaufen, wobei der Fokus auf die Fragen zur Vollständigkeit zu legen ist. Die ergänzten Systemelemente und Beziehungen werden einem Review unterzogen und evtl. noch fehlende Modellelemente identifiziert. Auch in diesem Review hängt die rechnerunterstützte Überprüfung von der Wahl des Tools ab.

Ist das zweite Review erfolgreich durchlaufen, kann die Systemstruktur freigegeben werden. Diese entspricht einem Stand 1.0 und wird im Verlauf der Entwicklung als Grundlage zur Abstimmung genutzt. Die Auswirkungen von Änderungsprozessen können auf Basis der Systemstruktur durchgeführt werden. Die Abstimmung der Fachdisziplinen sollte anschließend in regelmäßigen Abständen erfolgen. Hierzu wird die Systemstruktur als Basis genutzt. Änderungen und neue Erkenntnisse zum System werden diskutiert und in dem Modell übernommen. Die Pflege und die Dokumentation des Modells erfolgt durch den Systems Engineer.

Das Vorgehen beschreibt die Erstellung der Systemstruktur. Dabei ist bei den Workshops nur ein eingeschränkter Kreis des Projektteams beteiligt. Für das Projekt und die disziplinübergreifende Abstimmung ist es wichtig und notwendig, dass jeder Entwickler sowie nicht-technische Bereiche des Projekts stets einen Zugriff auf die Systemstruktur haben. Dabei dürfen sie die Inhalte lesen, jedoch nicht bearbeiten. Die Anpassungen finden nur durch den Systems Engineer statt.

5 Anwendung und Bewertung

In Kapitel 5 erfolgt die beispielhafte Anwendung der in Kapitel 4 eingeführten Konzepte. Hierzu wurden zwei Beispiele ausgewählt. Im ersten Beispiel wird an einem fiktiven Unternehmen das Vorgehen aus Kapitel 4.6 gezeigt. Dabei wird davon ausgegangen, dass das Produkt neu entwickelt wird. Damit wird die Systemstruktur gemeinsam in einer Gruppe erarbeitet und anschließend rechnerintern abgebildet und überprüft. Die Systemstruktur stellt die Visualisierung des Produktkonzepts dar und unterstützt bei Entscheidungen. Neben der Visualisierung wird ein formales Repository (Datenmodell) benötigt, damit die erarbeiteten Informationen über den gesamten Entwicklungsprozess genutzt werden können. Dieses Vorgehen wird in Kapitel 5.1 am Anwendungsbeispiel der Tretkraftunterstützung für ein Fahrrad beschrieben.

Im zweiten Anwendungsbeispiel wird ein System betrachtet, das bereits realisiert wurde und damit im Nachhinein modelliert wird. Dies stellt einen typischen Anwendungsfall einer Systemanalyse, z.B. im Rahmen eines Reverse Engineering, dar. Ein bestehendes System wird nachmodelliert und analysiert. Bei dem Beispiel handelt es sich um eine Sortieranlage. Die Systemstruktur wird in Kapitel 5.2 Schichtweise aus den verschiedenen Sichten (stoffspezifisch, energiespezifisch, informationsspezifisch) aufgebaut. Die Sortieranlage unterscheidet sich von der Tretkraftunterstützung zusätzlich in der Produktklasse, da es eine Produktionsanlage darstellt. Die Tretkraftunterstützung ist ein Energieumsetzendes Element, während die Sortieranlage einem Stoffändernden Element entspricht. Damit wird im zweiten Beispiel der Stofffluss näher betrachtet.

5.1 Anwendungsbeispiel: Tretkraftunterstützung für ein Fahrrad

Mechatronik hält in immer mehr Bereichen des alltäglichen Lebens Einzug. Die sonst mechanischen Produkte werden mit Aktoren und Sensoren ausgestattet und können damit neue Funktionen realisieren. Das Fahrrad stellt ein gutes Beispiel dar. Die Technologie rund ums Fahrrad ist ständig im Wandel, z.B. durch verbesserte mechanische Prinzipien (Federungen) oder neue Materialien (Kohlenstofffaserverstärkter Kunststoff). Mit dem neuen Werkstoff konnte z.B. das Gewicht des Fahrrads reduziert werden, mit besserer Festigkeit. Jedoch wird seit mehreren Jahren ein neuer Trend in der Fahrradindustrie deutlich: die Mechatronisierung des Fahrrads durch die Integration der Aktorik und Sensorik. Es wird ein steigender Absatzmarkt für Elektrofahrräder beobachtet [ZVE09].

Ein Unternehmen, das bislang Fahrräder produziert hat, entschließt in diesen Markt einzusteigen. Hierzu soll eine elektrische Tretkraftunterstützung entwickelt werden. Diese Fahrräder werden auch Pedelec (Pedal Electric Cycle) bezeichnet. Die Bedingungen an die Art der Unterstützung wird in den EU Richtlinien 2002/24EG sowie der DIN Norm DIN EN 15194 festgelegt [EU 02], [DIN15194]. Dabei wird die Motorleistung mit einer Obergrenze von 250 W vorgeschrieben. Die Tretkraftunterstützung darf bis zu einer

Geschwindigkeit von 25 km/h zur Verfügung stehen. Mit diesen Rahmenbedingungen unterliegen die Fahrräder keiner Helm- oder Versicherungspflicht.

Da es sich bei dem neuen Produkt um ein mechatronisches System handelt und die Entwicklung nicht mehr rein mechanisch geprägt ist, sollen die Ansätze des Systems Engineering greifen. Hierzu wurde ein Systems Engineer eingestellt. Für die Entwicklung wurden Experten der Sensortechnik und Antriebstechnik extern zugekauft. Ein Produktmanager hat im Vorfeld erste Anforderungen erarbeitet und die Funktionen beschrieben.

Das Konzept der Tretkraftunterstützung soll gemeinsam mit allen Fachdisziplinen erarbeitet und in einer Systemstruktur abgebildet werden. Die Systemgrenze wird so gelegt, dass das Fahrrad selbst nicht Teil der Entwicklung ist – das zu betrachtende System ist die Tretkraftunterstützung (TKU). Anpassungen am bestehenden Fahrrad sollen dennoch möglich sein.

An dem Beispiel der Tretkraftunterstützung wird das Vorgehen zur Erstellung einer plausiblen Systemstruktur vorgestellt. Die einzelnen Phasen werden an dem Beispiel erläutert und die Ergebnisse dargestellt.

5.1.1 Phase 1: Modellierungs-Workshops

Die Konzeptentwicklung wird vom Systems Engineer moderiert durchgeführt. Dieser beruft Workshops mit den Fachexperten ein und schult die Teilnehmer in dem Vorgehen. Hierzu werden mehrere Sitzungen eingeplant, die vom Systems Engineer vorbereitet werden. Durch die Workshops wird die Diskussion zwischen den Fachexperten gefördert und ein einheitliches Verständnis des Systems geschaffen. Offene Fragen werden im Themenspeicher gesammelt. Zusätzlich wird ein Glossar erstellt, der die verwendeten Abkürzungen in der Systemstruktur enthält. Die Darstellung der Systemstruktur erfolgt an Moderationswänden. Dazu werden die Sprachkonstrukte der Modellierungssprache CONSENS verwendet. Zur Informationsgewinnung reichen die sechs Konstrukte: Umfeldelement, Systemelement, mechanische Verbindung, Energie-, Informations- und Stofffluss aus (Bild 5-1). Mit wenigen Konstrukten und der graphischen Unterscheidung zwischen den darzustellenden Elementen und Beziehungen wird den Teilnehmern eine intuitive und einfache Sprache geboten. Die farbliche Trennung der Umfeldelementen und den Systemelementen zeigt indirekt die Systemgrenze auf.

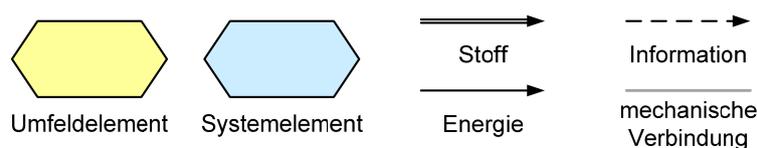


Bild 5-1: CONSENS-Modellkonstrukte zur Darstellung der Systemstruktur

Die Elemente werden von den Teilnehmern benannt und die Beziehungen auf der Moderationswand gezogen und beschriftet. Zu Beginn wird die Einbettung des Systems in

seine Umgebung genauer untersucht und modelliert. Hierzu wird das zu entwickelnde System TKU als Black-Box betrachtet (Bild 5-2). Die Umfeldelemente werden in der Diskussion identifiziert. Dazu gehören das *Fahrrad* sowie ein Benutzer – hier von den Teilnehmern als *Fahrer* bezeichnet. Die *Ladestation* stellt die externe Energiequelle dar und wird ebenfalls als Umfeldelement spezifiziert. Umwelteinflüsse, die bislang beim Fahrrad nicht sicherheitskritisch waren, wie *Regen* oder *Temperatur*, spielen bei einem elektrischen Antrieb eine entscheidende Rolle. Alle Einflüsse werden gesammelt und notiert (Bild 5-2). Das dazugehörige Umfeldelement wird von den Teilnehmern als *Umwelt* bezeichnet.

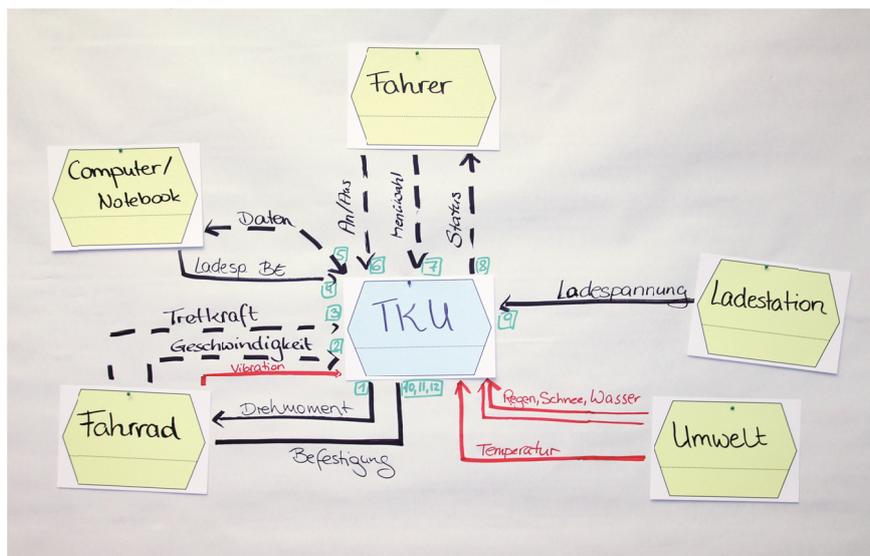


Bild 5-2: Umfeldmodell der Tretkräftunterstützung

Damit die TKU nur unterstützend wirkt, wenn der Fahrer in die Pedale tritt, muss dies durch Sensoren erfasst werden. Die Messgrößen *Geschwindigkeit* und *Tretkraft* werden dabei am Fahrrad bestimmt. Dies wird durch eine Informationsbeziehung mit den entsprechenden Bezeichnungen vom Fahrrad zum System angedeutet. Die Tretkräftunterstützung erfolgt durch die Übertragung eines Drehmoments vom System auf das Fahrrad (Energiefluss *Drehmoment*).

Die Interaktion mit dem Fahrer wird ausführlich diskutiert. Dieser soll das Gerät ein- und ausschalten können. Darüber hinaus sollen mehrere Programme zur Verfügung stehen, die vom Fahrer gewählt werden. Dies wird mit dem Informationsfluss *Menüwahl* dargestellt. Der Benutzer soll von der TKU eine Rückmeldung durch eine Statusanzeige erhalten. Welche Informationen sich im Speziellen dahinter verbergen, wird zu diesem Zeitpunkt noch nicht festgelegt. In diesem Zusammenhang kam jedoch die Idee auf, die TKU an einen Computer/Notebook anschließen zu können. Damit soll erreicht werden, dass Daten von der TKU auf den Rechner geladen werden können (z.B. gefahrene Kilometer/Geschwindigkeiten etc.).

Im anschließenden Workshop wird das Innere des Systems spezifiziert. Hierzu werden zu Beginn alle Flüsse an der Systemgrenze betrachtet und gemeinsam erarbeitet, welche

Systemelemente diese Interaktion mit den Umfilderelementen ermöglichen. Daraus entstand die nächste Hierarchieebene des Systems, die in Bild 5-3 vorzufinden ist.

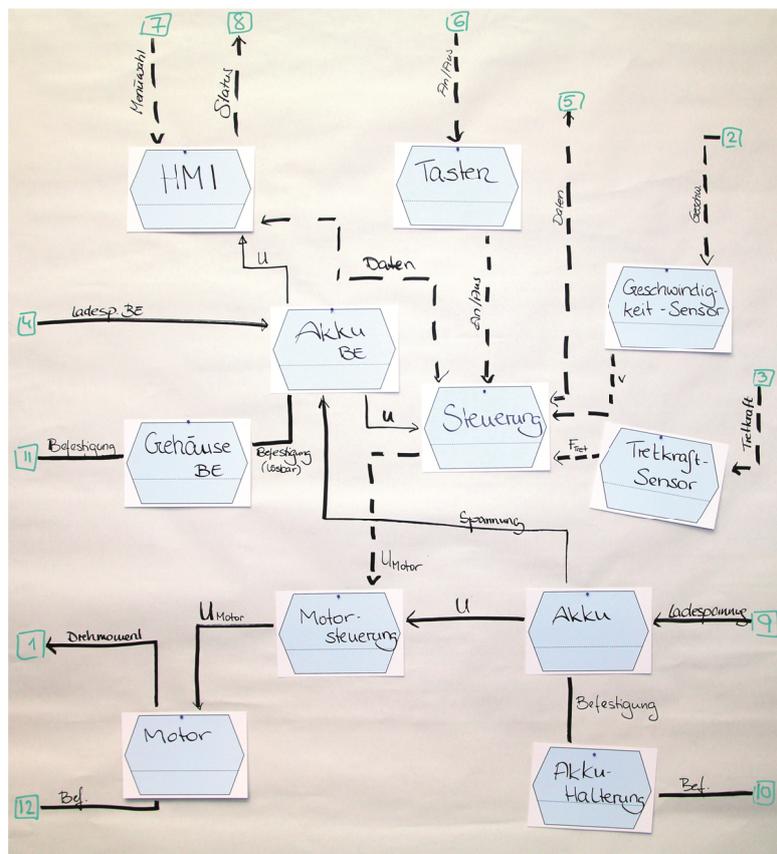


Bild 5-3: Wirkstruktur der Tretkraftunterstützung

Die Beziehungen des Umfeldmodells werden durchnummeriert und in der Wirkstruktur mit Systemelementen verbunden. Dabei wird festgelegt, dass es Tasten für die Funktion „Gerät ein-/ausschalten“ geben soll. Die Menüwahl und die Rückmeldung zum Benutzer über den Status hingegen soll durch ein HMI (Human Machine Interface) erfolgen.

Für die Messung der *Geschwindigkeit* und *Tretkraft* werden zwei Sensoren eingeplant. Eine *Steuerung* soll die Daten auswerten und daraus eine *Motorspannung* berechnen. Diese Information wird von einer zusätzlichen *Motorsteuerung* verarbeitet und die entsprechende Spannung (Energiefluss *Motorspannung*) am Motor angelegt.

Die Befestigung der Funktionsträger wird in dieser frühen Phase nur für die Elemente *Motor*, *Akku* und der Bedieneinheit (entspricht den Elementen im eingekreisten Bereich) spezifiziert. Daher wird ein Gehäuse als Element eingebracht, das die Bedieneinheit zusammenhält (*Gehäuse BE*). Die Befestigung erfolgt am Fahrrad. Die Bedieneinheit verfügt über eine eigene Energiequelle (*Akku BE*) und ist mit dem Gehäuse verbunden. Diese Beziehung muss lösbar sein, so dass der Akku ausgetauscht werden kann.

Die Befestigung des Hauptakkus wird ebenfalls in der Runde diskutiert. Hierzu wird vom Produktmanagement die Anforderung vorgegeben, dass der Akku vom Fahrrad

abgenommen werden kann, um an einer externen Station zu laden. Dies soll mit Hilfe einer eigenen Halterung umgesetzt werden.

Die vielen Diskussionen innerhalb der Workshops zeigen den direkten Handlungsbedarf auf. So ist im Gespräch deutlich geworden, dass die Kompetenzen im Bereich HMI im Unternehmen noch nicht vorhanden sind. Fragestellungen, wie z.B.: „Welche Knöpfe sollen mit welchen Funktionen vorgesehen werden?“, „Wie ist die Menüführung für den Benutzer?“ oder „Was wird, wann, wie angezeigt?“ konnten in der Runde nicht geklärt werden. Ein externer Auftrag ist im Anschluss in die Wege geleitet worden.

Die Workshops sind von allen Beteiligten als positiv bewertet worden. Die Diskussionen in der Gruppe führten zu weiteren Anforderungen an das Produkt und zeigten Handlungsbedarf auf. Die Systemstruktur diente dabei als ein Medium zur Visualisierung der besprochenen Wechselwirkungen im System. Das erarbeitete Modell wird dabei immer konkreter, so dass bereits in der White-Box Darstellung neue Beziehungen zwischen System und Umfeldelemente hinzugekommen sind, die vorher in der Black-Box Darstellung nicht bedacht wurden.

5.1.2 Phase 2: Formalisierung des Modells

Im Nachgang zu den Workshops werden die Modelle rechnerintern abgebildet und in diesem Zuge formalisiert. Der Systems Engineer wählt hierzu ein geeignetes Softwarewerkzeug aus. In diesem Fall wurde das Werkzeug Enterprise Architect von Sparx Systems mit der Modellierungssprache SysML ausgewählt. Dieses Werkzeug war bereits bei dem Unternehmen eingeführt worden, da es bei der Softwareentwicklung der Trekkraftunterstützung zum Einsatz kommt. Durch die SysML-Integration konnte dieses für die Systemmodellierung genutzt werden.

Die SysML bietet durch den Profilmechanismus die Definition von Stereotypen, die projektweit gelten. Diese werden genutzt, um die Klassen der Elemente und Beziehungen abzubilden. Die Stereotype werden einmal im Werkzeug angelegt und können für Folgeprojekte wiederverwendet werden.

Im ersten Schritt werden die Umfeldelemente angelegt und mit dem entsprechenden Stereotyp versehen. Dabei erfolgt die Trennung zwischen technischen und nicht-technischen Elementen. Das Umfeldelement *Umwelt* wird als nicht-technisches Element dem Stereotyp «*Umgebung*» zugeordnet; der *Fahrer* erhält den Stereotyp «*Lebewesen*». Für die technischen Elemente wie *Fahrrad*, *Ladestation* und *Computer/Notebook* wird die entsprechende Hauptfunktion bestimmt. Das *Fahrrad* erfüllt die Grundfunktion „Energie wandeln“ und erhält damit den Stereotyp «*Energiewandler*» – der *Computer* ist eine «*Informationsverarbeitung*» und die *Ladestation* die «*Energieübertragung*».

Nach den Umfeldelementen werden die Beziehungen klassifiziert. Der Systems Engineer geht jede Beziehung durch und entscheidet, welcher Klasse die Beziehung entspricht. Die *Versorgungsenergie* bspw. ist eine elektrische Energie und wird dement-

sprechend typisiert. Der *Antrieb*, der von der TKU auf das Fahrrad wirkt, ist wiederum eine mechanische Energie. Da ein Drehmoment auf das Getriebe des Fahrrads übertragen werden soll, handelt es sich um «rotatorische Energie». Bei den Informationsflüssen sind Messgrößen vorhanden, die als solche zu typisieren sind. Beispielsweise wird die Geschwindigkeit des Fahrrads nicht vom Fahrrad gesendet, sondern am Fahrrad erfasst. Die Messgröße *Geschwindigkeit* wird im Modell angelegt. Die Bedieneingaben des *Fahrers* werden nach Kap. 4.1.3.1 ebenfalls als «Messgröße» spezifiziert. Die restlichen Informationsbeziehungen sind vom Typ «Signal».

Die Stoffbeziehung *Regen, Schnee, Wasser* zeigt eine störende Wirkung auf. Diese Einflüsse haben eine Auswirkung auf die Auslegung der Dichtigkeit des Systems. In spezifischen Normen werden Anforderungen beschrieben, die für elektronische Bauteile gelten. Diese Normen wurden identifiziert und mit einer Verlinkung zur entsprechenden Anforderung der Beziehung hinterlegt.

In Bild 5-4 ist das Umfeldmodell im Werkzeug formalisiert abgebildet. Die Stereotype sind in spitzen Klammern über dem Namen des Elements zu finden. Das System selbst hat die Energiebeziehung als Hauptfluss (elektrische Energie IN und rotatorische Energie OUT) und stellt damit einen Energiewandler dar. Die Beziehungen werden in SysML über die Ports spezifiziert. Der Stereotyp ist vor dem Namen der Beziehung dargestellt (z.B. «rotatorische Energie» *Drehmoment*). Die mechanische Verbindung ist in der frühen Phase der Modellierung noch nicht weiter unterteilt. Die Schlussarten stehen noch nicht fest. Aus diesem Grund wird im Modell der Stereotyp «mechanische Verbindung» verwendet.

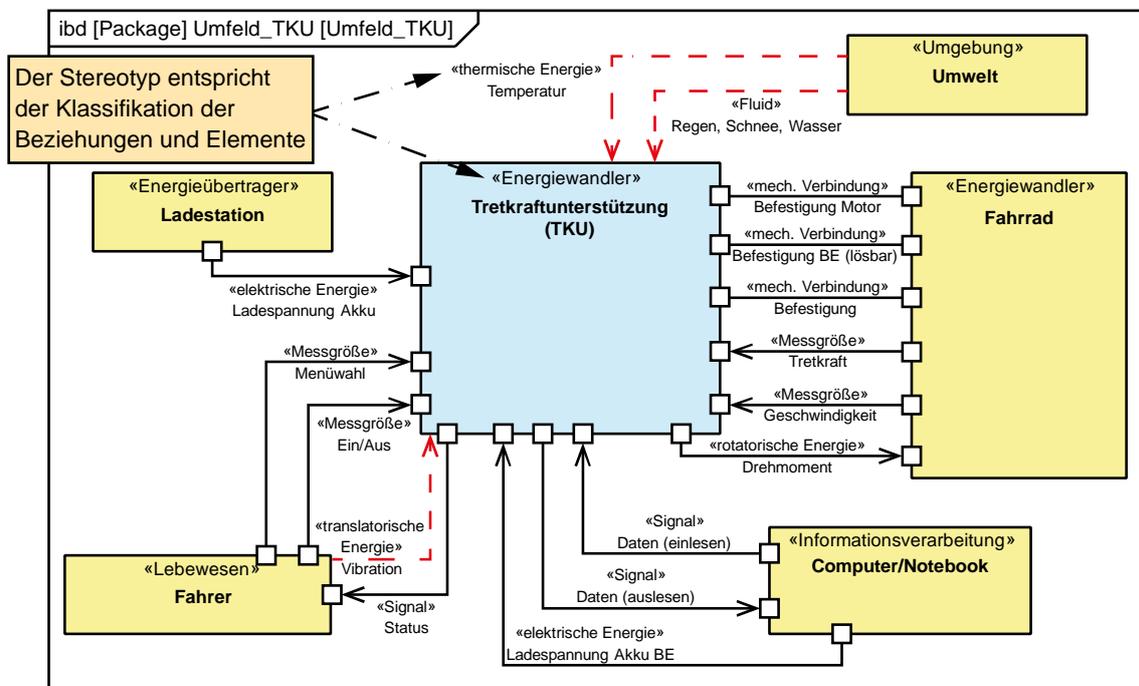


Bild 5-4: Formalisierte Darstellung des Umfelds der Tretkraftunterstützung im SysML-Werkzeug

Im nächsten Schritt ist eine Struktur des Systems zu definieren. Dies erfolgt in einem Blockdefinitionsdiagramm in Form einer Hierarchie (Bild 5-5). Die Systemhierarchie wurde, bis auf die Bedieneinheit, im Workshop nicht weiter berücksichtigt. Der Systems Engineer fasst funktionale Elemente zusammen. Daraus ergibt sich eine neue Ebene, bestehend aus den Systemelementen *Bedieneinheit*, *Sensoreinheit* und *Antriebseinheit*.

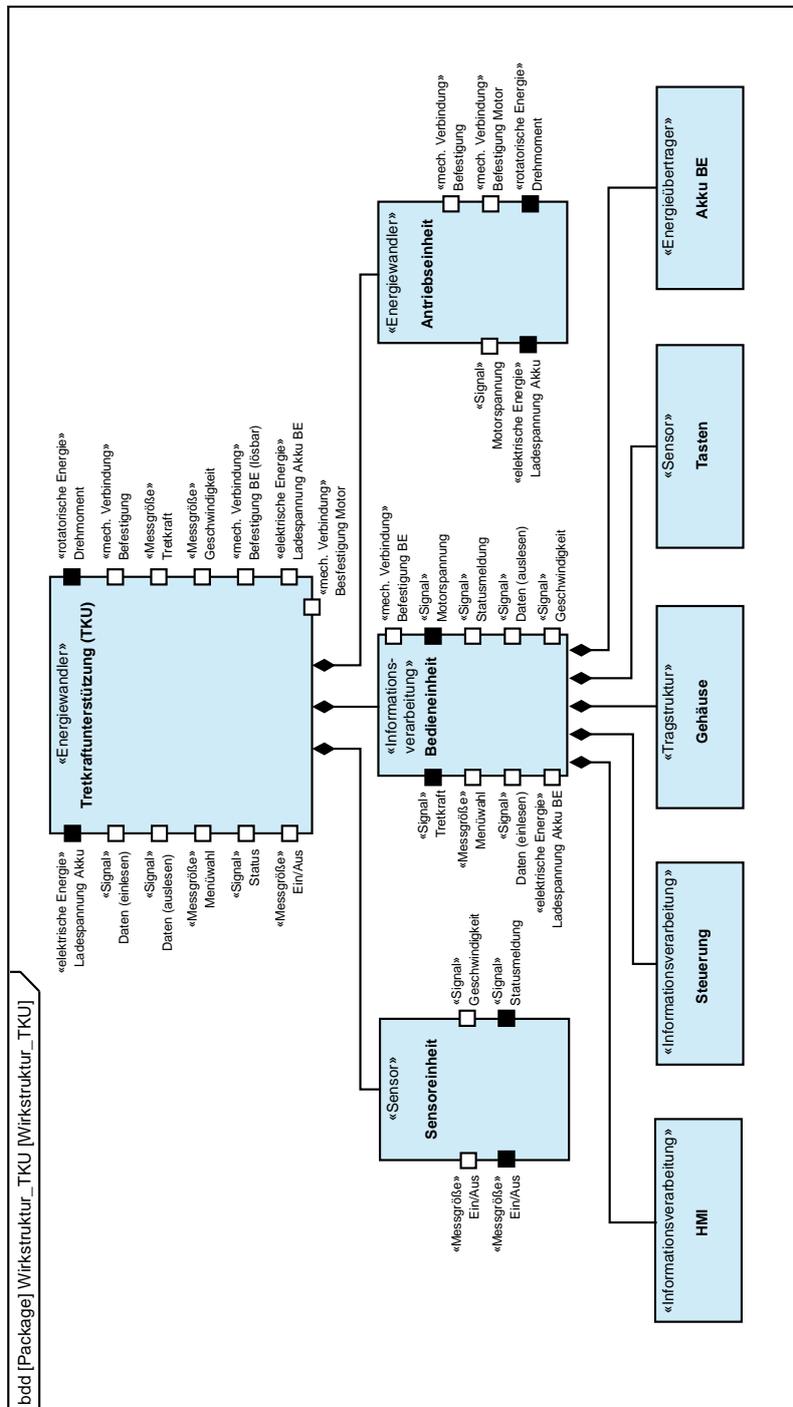


Bild 5-5: Ausschnitt des Blockdefinitionsdiagramms der Tretkraftunterstützung

Für jedes Element wird wiederum die Klasse definiert und als Stereotyp modelliert. Der Systems Engineer geht hierzu die Elemente durch und identifiziert die Hauptfunktion, respektive den Hauptfluss. Auf Basis des Hauptflusses entscheidet der Modellierer, ob es sich um ein *Energie-, Stoff- oder Informationsumsetzendes Element* oder eine *Tragstruktur* handelt. Im Bild 5-5 ist der Hauptfluss durch die eingefärbten Ports hervorgehoben dargestellt. Die *Sensoreinheit* ist vom Typ «*Sensor*» («*Messgröße*» IN und «*Signal*» OUT). Die *Bedieneinheit* hat als Hauptfluss «*Signal*» IN und OUT und wird als «*Informationsverarbeitung*» typisiert. Die *Antriebseinheit* erbt die Klasse «*Energiewandler*» der *Tretkraftunterstützung* («*elektrische Energie*» IN und «*rotatorische Energie*» OUT). In der Darstellung ist die weitere Ebene für die *Bedieneinheit* zu sehen, bestehend aus *HMI, Steuerung, Gehäuse, Tasten* und *Akku BE*, mit der entsprechenden Klassifikation. Die Beziehungen zwischen den Elementen werden in einem ibd (internes Blockdiagramm) der SysML spezifiziert.

Durch dieses Vorgehen erfolgt bereits ein erstes Review. Der Modellierer betrachtet den Hauptfluss und ergänzt, falls ein Hauptfluss nicht korrekt oder unvollständig spezifiziert wurde.

5.1.3 Phase 3: Modell-Review

Die Plausibilitätsprüfung erfolgt auf Basis der Checkliste. Dabei prüft der Systems Engineer die Systemstruktur im ersten Schritt auf die vergleichbare Darstellung. Im Rahmen der Workshops hat er bereits darauf geachtet, dass die Beziehungsart dem Zweck entspricht. Ausgehend von den Beziehungen konnte er bei der Formalisierung die Elementklassen bestimmen. Die Hierarchie der Systemstruktur wurde im Nachgang zu den Workshops eingefügt. Dies ist nach funktionalen Gesichtspunkten erfolgt. Nach dem Umfeldmodell wurde die Ebene mit den Funktionseinheiten *Sensor-, Bedien- und Antriebseinheit* eingefügt.

Zur Überprüfung der Richtigkeit werden die Matrizen der Energieumsetzenden Elemente und Sensoren genutzt. Hierzu geht der Systems Engineer erst die Energieumsetzenden Elemente der untersten Ebene durch und gleicht deren Darstellung mit der Matrix ab. Am Beispiel des Akkus wird dies veranschaulicht: Der *Akku* ist in der Matrix in der Diagonalen im Kreuzungspunkt elektrische Energie IN und elektrische Energie OUT vorhanden. Damit ist die Darstellung richtig vorgenommen und entspricht auch der vergebenen Klasse «*Energieübertrager*». Bei den Sensoren erfolgt die Abfrage analog. Hier ist die eingehende Beziehung vom Typ «*Messgröße*» ausschlaggebend. Die *Sensoreinheit* enthält einen Sensor zur Bestimmung der Messgröße *Geschwindigkeit*. In der Matrix sind mehrere Möglichkeiten beschrieben, diese Messgröße zu bestimmen. In diesem Fall soll die Geschwindigkeit über einen *Drehwinkelsensor* erfasst werden, der in der Spalte (induktiv L) enthalten ist. Die Information des Wirkprinzips (induktiv) wird dem Sensor als Zusatzinformation (Eigenschaft) hinterlegt.

5.1.4 Phase 4: Modellvervollständigung

Das bestehende Modell wird um die Attribute *aktiv*, *gestaltbehaftet* und *Energiequelle* erweitert. Diese Angabe wird im rechnerinternen Modell hinterlegt, indem *Tagged Values* definiert werden. Diese können in den Diagrammen angezeigt werden (Bild 5-6 am Beispiel des Gehäuses). Die *tags* werden auf *true* gesetzt, wenn das Element mit dem Attribut versehen werden kann. Das Gehäuse ist beispielsweise kein aktives Element und entspricht auch keiner Energiequelle (beide *false*). Da es eine Gestalt besitzt, wird der Wert *gestaltbehaftet* auf *true* gesetzt.

«Tragstruktur» Gehäuse
<i>tags</i> <i>aktiv = false</i> <i>Energiequelle = false</i> <i>gestaltbehaftet = true</i>

Bild 5-6: Vergabe der Attribute *aktiv*, *gestaltbehaftet* und *Energiequelle*

Beim Vergeben der Attribute prüft der Systems Engineer die damit verbundenen Bedingungen. Eine Bedingung besagt, dass es im Modell mindestens eine Energiequelle geben muss. Dies trifft in diesem Fall zu, da im System die beiden *Akkus* den internen Energiequellen entsprechen sowie die *Ladestation* der externen Energiequelle.

Mit dem Attribut *aktiv* ist die Bedingung verknüpft, dass es mindestens eine Beziehung vom Typ «elektrische Energie» an dem Element vorhanden sein muss. Bei der Überprüfung konnten fehlende Beziehungen identifiziert werden. So fehlte an den Sensoren die Energieversorgung. Diese sind nachträglich ergänzt worden.

Zur Überprüfung der Bedingung zum Attribut *gestaltbehaftet* (mindestens eine Beziehung vom Typ «mechanische Verbindung» vorhanden) wird eine **Beziehungsmatrix** generiert. Daraus können fehlende Beziehungen erkannt und ergänzt werden. Bild 5-7 zeigt die Matrix mit allen Elementen, deren Wert *gestaltbehaftet* auf *true* gesetzt ist.

Die nicht verbundenen Elemente sind grau hinterlegt. So ist erkennbar, dass die Befestigung der Sensoren im Modell noch nicht berücksichtigt wurde. Diese und die anderen fehlenden Beziehungen werden in einem Treffen mit den entsprechenden Fachexperten durchgesprochen. In diesem Zusammenhang wird die **Sichtenbildung** genutzt, um die Modellkomplexität visuell zu reduzieren und damit den Fokus auf die gestaltbehafteten Elemente mit den entsprechenden Beziehungen zu legen. Die einzelnen Elemente sind am Fahrrad befestigt. Hierzu wird das Fahrrad noch weiter unterteilt, so dass die Befestigungsorte (*Rahmen*, *Lenker*, *Rad*, etc.) deutlich werden.

Beziehungsmatrix											
Fragestellung: „Ist das Element (Zeile) mit dem Element (Spalte) mechanisch Verbunden?“ S = Stoffschluss F = Formschluss K = Kraftschluss X = k. A. zur Schlussart		gestaltbehafete Elemente									
		Gehäuse BE	Akku BE	Tasten	Geschwindigkeits-Sensor	Tretkraft-Sensor	Motor	Motorsteuerung	Akku	Akku-Halterung	Fahrrad
gestaltbehafete Elemente	Nr.	1	2	3	4	5	6	7	8	9	10
Gehäuse BE	1		X								X
Akku BE	2	X									
Tasten	3										
Geschwindigkeits-Sensor	4										
Tretkraft-Sensor	5										
Motor	6							X			X
Motorsteuerung	7						X				
Akku	8									X	
Akku-Halterung	9								X		X
Fahrrad	10	X					X			X	

Bild 5-7: Beziehungsmatrix der gestaltbehafeten Elemente

Bei dieser Betrachtung ist ein weiterer Handlungsbedarf identifiziert worden. Die Auslegung des Rahmens kann nicht ohne weiteres von bisherigen Fahrrädern übernommen werden. Durch die neu hinzugekommenen Elemente (*Akku, Antrieb, etc.*) sind zusätzliche Massen und damit zusätzliche Belastungen des Rahmens zu berücksichtigen. Die daraus resultierende Belastungsprüfung wird in einem Testfall dokumentiert.

5.1.5 Phase 5: Modellfreigabe

In der letzten Phase wird ein erneutes Review mit dem Fokus auf die Vollständigkeit des Modells durchgeführt. Die Fragen der Checkliste werden systematisch abgearbeitet und vom Systems Engineer beantwortet. Die vollständige Systemstruktur wird in einem abschließenden Workshop mit den Fachexperten besprochen. Dabei werden alle Änderungen zum Stand nach Abschluss der Phase 1 erläutert.

5.2 Anwendungsbeispiel: Sortieranlage

Im Folgenden wird die Systemstruktur einer Sortieranlage aufgezeigt. Zur Visualisierung werden in diesem Beispiel die CONSENS-Konstrukte gewählt. Die rechnerinterne Modellierung erfolgt in Visio. Die Software Visio stellt dabei kein Modellierungswerkzeug dar, da kein auswertbares Repository (Datenmodell) erzeugt wird. In diesem Beispiel reicht die graphische Darstellung aus, da mit dem Modell ein Gesamtverständnis der Anlage entstehen soll. Darüber hinaus sind keine weiteren Anwendungen einer datentechnischen Verwaltung oder Analyse vorgesehen.

Die Sortieranlage ist ein Versuchsaufbau der Universität Paderborn. Die Anlage wird in der Lehre zur Steuerungsprogrammierung eingesetzt. Das Bild 5-8 zeigt den Aufbau der Anlage. Die Hauptaufgabe der Anlage ist es, Probenkörper in vordefinierte Lager zu sortieren. Die Probenkörper haben alle die gleiche Form, unterscheiden sich jedoch in Farbe und Material. Es sind rote und schwarze Probenkörper sowie die Materialien Kunststoff und Metall vorhanden. Eine Kombination der Farben und Materialien kann vorkommen (z.B. rote Metall-Probenkörper oder rote-Kunststoff-Probenkörper). Der Benutzer startet die Anlage und führt die Probenkörper in das Führungsrohr ein, an dessen Ende Sensoren zur Auswertung der Farbe und Materialien vorhanden sind. Die Daten werden erfasst und ausgewertet. Ein Zylinder schiebt anschließend den Probenkörper auf ein Förderband. Dieses bewegt die Körper in Richtung der Lager. Vor jedem Lager sind Lichtschranken angebracht, so dass die Position des Körpers erfasst werden kann. Über die Information der Farbe und des Materials in Verbindung mit der Position werden Zylinder an den entsprechenden Lagern ausgefahren und der Probenkörper in das Lager geschoben. Hierzu sind drei Lager vorhanden. In Lager 1 werden alle roten Metall-Probenkörper, in Lager 2 alle schwarzen Metall-Probenkörper und in Lager 3 alle Kunststoff-Probenkörper gesammelt. Nachdem die Körper sortiert sind werden die Lager von dem Benutzer geleert und das Programm abschaltet. Die Zylinder werden mit Druckluft betrieben.

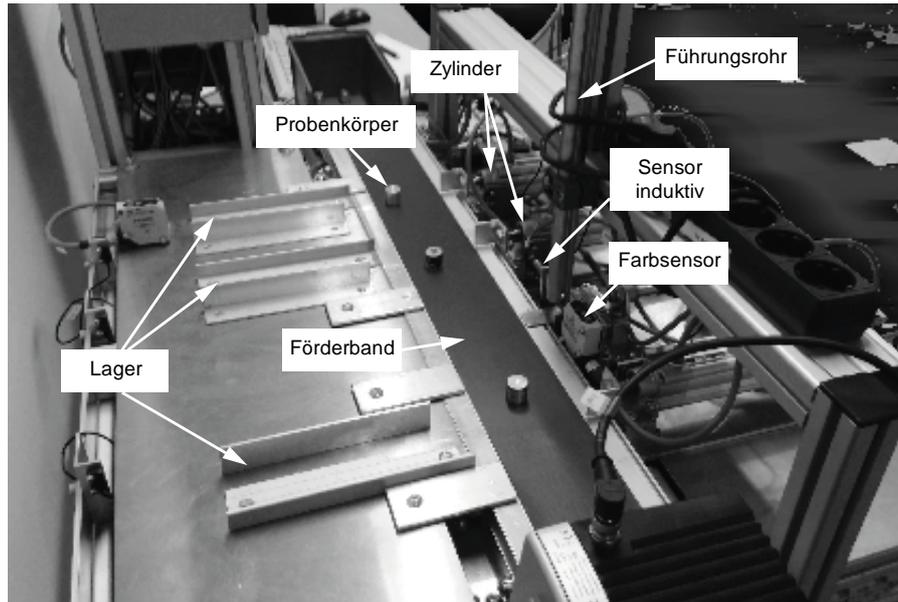


Bild 5-8: Aufbau der Sortieranlage

5.2.1 Oberste Hierarchieebene der Systemstruktur

Bild 5-9 zeigt die Systemstruktur der Sortieranlage auf der obersten Hierarchieebene – dem Umfeld. Im Folgenden wurde auf die Darstellung der Ports verzichtet. Für die Systemstruktur werden zu Beginn die Elemente identifiziert, die außerhalb der Systemgrenze liegen. Die Anlage stellt kein Subsystem eines übergeordneten Systems dar, so dass die nicht-technischen Elementklassen nach Kapitel 4.1.3 genutzt werden können, um die Elemente außerhalb der Systemgrenze zu finden. Der *Benutzer* (Typ «Lebewesen») interagiert auf zwei Weisen mit dem System. Zum einen führt er die Probenkörper ein und entleert die Lager (Stoffbeziehung zwischen Benutzer und System). Zum anderen startet und beendet er das Steuerprogramm (Informationsbeziehung zwischen Benutzer und System).

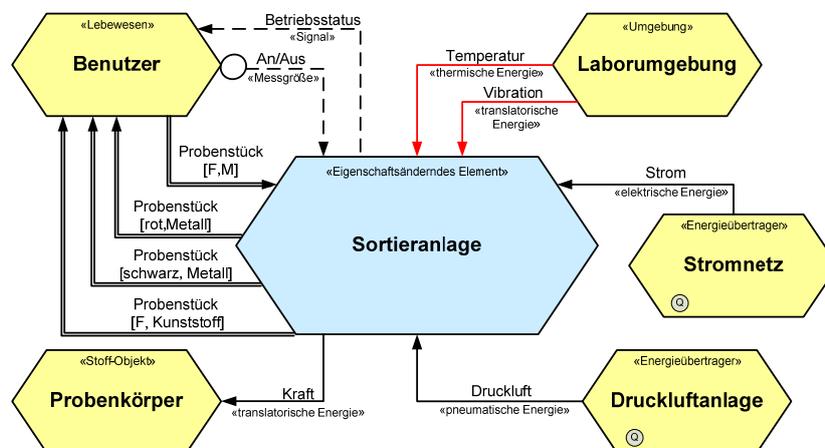


Bild 5-9: Systemstruktur der Sortieranlage – Umfeldmodell

Die Besonderheiten der Stoffbeziehung werden an dem Beispiel verdeutlicht. Das Probenstück interagiert mit dem System in vielfältiger Art und Weise (z.B. Transport, Krafteinwirkung durch die Zylinder). Zusätzlich stellt es einen Stofffluss durch das System dar. Die Modellierung erfolgt nach Bild 5-9. Das Element *Probenkörper* befindet sich außerhalb der Systemgrenze und erhält die Klassifikation «*Stoff-Objekt*». Mit dem Element werden die Merkmale und Merkmalsausprägungen definiert. Das Element hat die Merkmale *Farbe* und *Material*. Die Ausprägungen sind demnach *Farbe=rot, schwarz* und *Material=Kunststoff, Metall*.

Die Sortieranlage befindet sich in einem Universitätslabor. Damit ist es Umgebungsbedingungen, wie *Temperatur* und *Vibration* ausgesetzt. Diese werden als störende Beziehungen vom Element *Laborumgebung* (vom Typ «*Umgebung*») zum System dargestellt.

Unter Berücksichtigung der Richtlinien zur Vollständigkeit werden die Energiequellen ergänzt. Diese sind das *Stromnetz* mit der elektrischen Energieversorgung und die *Druckluftanlage* mit der pneumatischen Energieversorgung. Die Elemente erhalten zu dem Typ «*Energieübertrager*» das Attribut *Energiequelle* (Piktogramm Q).

Die Sortieranlage erfüllt die Hauptfunktionen „Stoff sortieren“. Dabei ändert sich die eingehende und ausgehende Stoffbeziehung hinsichtlich des Stoffs (*Probenkörper*) nicht. Es wird lediglich nach Merkmalsausprägung sortiert. Damit ist die Sortieranlage vom Typ «*Eigenschaftsänderndes Element*».

5.2.2 Sichten auf die Systemstruktur der Sortieranlage

Die Sicht in das System zeigt die Elemente des Systems auf. Dazu gehören das Führungsrohr, die Zylinder und Ventile, die Lager, das Förderband, die Steuerung und die Sensoren. Im Folgenden werden verschiedene Sichten nacheinander aufgebaut. Die komplette Systemstruktur als Kombination aller Sichten ist im Anhang enthalten (A6).

5.2.2.1 Stoffspezifische Sicht

Zu Beginn wird der Stofffluss durch das System modelliert. Dieses geht vom *Benutzer* über das *Führungsrohr*, dem *Auswerfzylinder* auf das *Förderband*. Dann startet der Sortiervorgang. Die Strukturdarstellung ist jedoch eine statische Sicht auf das System. Es werden alle Wechselwirkungen aufgezeigt, ohne den zeitlichen Zusammenhang. Die Abfolge wird im Aspekt Verhalten modelliert. Vom *Förderband* gibt es die Stoffbeziehung zu den *Zylindern* und anschließend in die *Lager*. Von da aus wieder zu dem *Benutzer*. Der Stofffluss durch das System ist in Bild 5-10 visualisiert. Die Klassifikation der Stoffbeziehung «*Festkörper*» ändert sich bei der Sortieranlage nicht, so dass sie in der graphischen Darstellung weggelassen wurde. Die Bezeichnung des Flusses enthält die Merkmalsausprägungen der Probenkörper. Dabei steht *F* für das Merkmal *Farbe* und *M* für *Material*. Ist keine Ausprägung gewählt, heißt das, dass alle Ausprägungen

vorkommen können. Bei einer Auswahl wird das gewählte Merkmal dargestellt (z.B. [schwarz, Metall]).

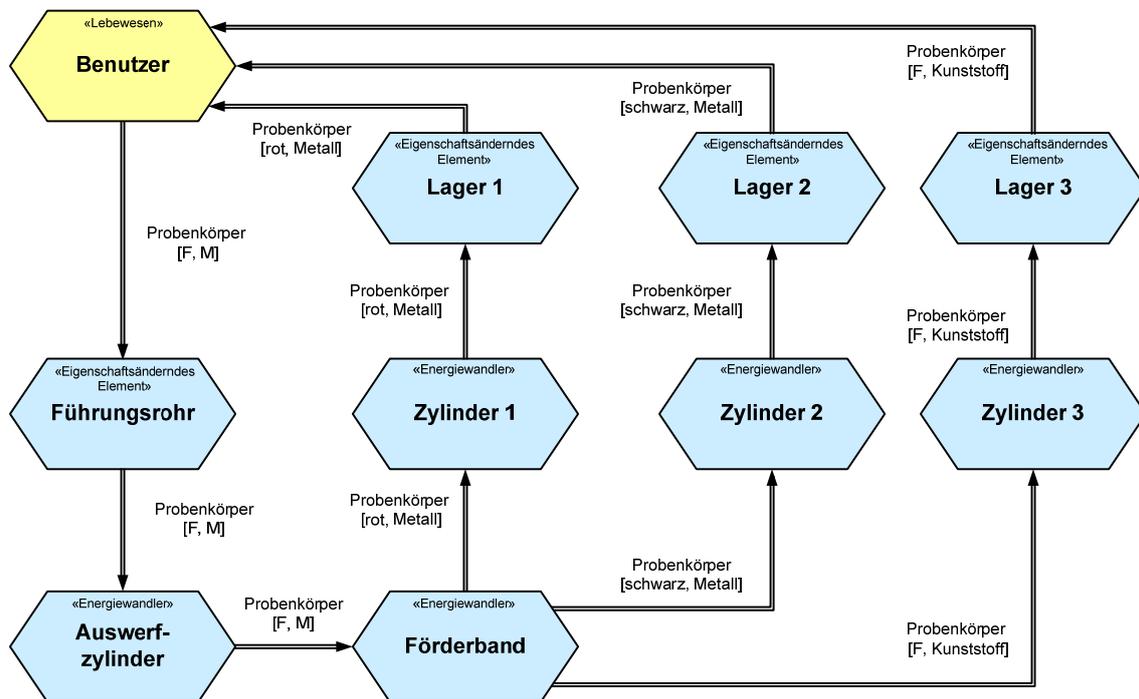


Bild 5-10: Systemstruktur der Sortieranlage – Stoffspezifische Sicht

In der Systemstruktur ist zu erkennen, dass für den Stofffluss nicht nur Stoffumsetzende Elemente gewählt wurden. Dies ist hier der Fall, da die Hauptaufgabe der Transport der Probenkörper ist. Wie in Kapitel 4.2.2 beschrieben, werden hierfür hauptsächlich Energieumsetzende Elemente benötigt, die eine Kraft oder Moment auf das zu transportierende Objekt ausüben. In diesem Fall sind es die *Zylinder*, die vom Typ *«Energiewandler»* sind und das *Förderband*, das vom Typ *«Energieübertrager»* ist. Für diese Elemente entspricht die Stoffbeziehung nicht dem Hauptfluss. Der Hauptfluss ist vom Typ Energie. Dieser wird im Folgenden in der energiespezifischen Sicht im Detail betrachtet.

5.2.2.2 Energiespezifische Sicht

Die energiespezifische Sicht zeigt eine Wirkkette von der *Druckluftanlage* bis zu dem *Probenstück* (Bild 5-11). Die Energieumsetzenden Elemente sind ein *Druckminderer*, *Ventile* und *Zylinder*. Dabei sind der *Druckminderer* und die *Ventile* vom Typ *«Energieübertrager»*. Die ein- und ausgehenden Beziehungen der Elemente sind vom Typ *«pneumatische Energie»*. Die Hauptfunktion dieser Elemente ist „Volumenstrom (dV/dt) und Druck (p) der Luft ändern“. Eine Energiewandlung findet dann bei den *Zylindern* statt. Hier wird pneumatische Energie in translatorische Energie gewandelt. Diese wird genutzt, um eine Kraft auf den *Probenkörper* auszuüben, die diesen von einer Position an eine andere Position schiebt.

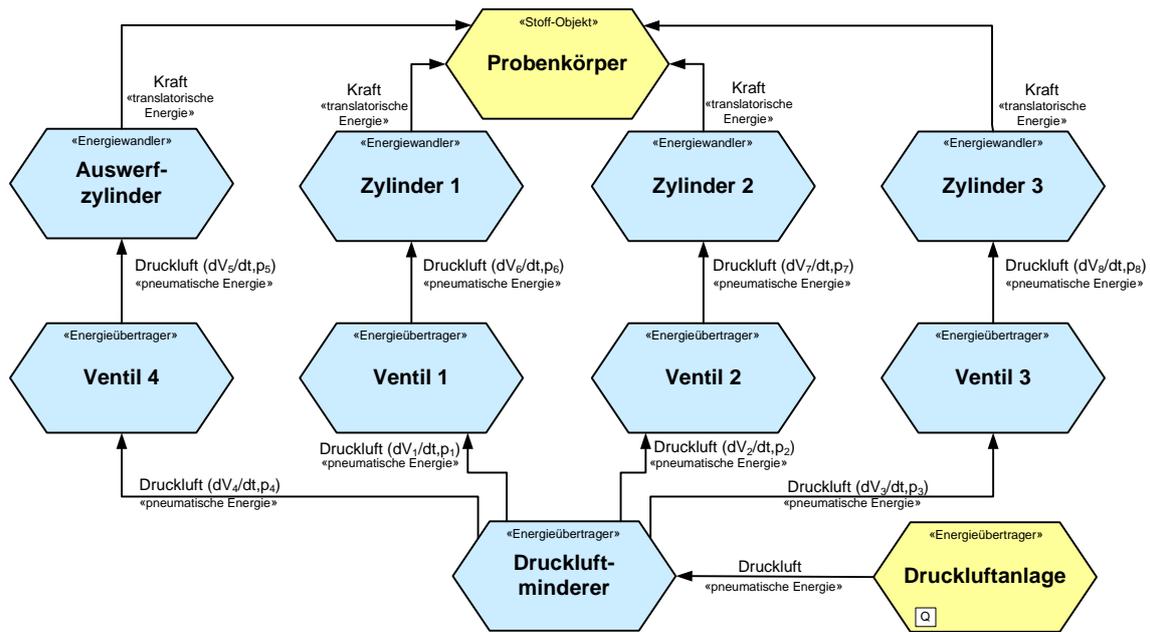


Bild 5-11: Systemstruktur der Sortieranlage – Energiespezifische Sicht

5.2.2.3 Informationsspezifische Sicht

Im nächsten Schritt wird die Steuerung der Anlage fokussiert betrachtet (Bild 5-12). Diese enthält die Sensoren und die Informationsverarbeitung *SPS*. Im System sind vier Arten von Sensoren vorhanden. Das *Material* der *Probenkörper* wird erfasst, während dieses im *Führungsrohr* ist. Dies ist durch die *«Messgröße»* am Element *Führungsrohr* dargestellt. Anschließend wird die *Farbe* bestimmt. Die *Lichtschranke* erfasst die *Position* des *Probenkörpers*, indem ein Lichtstrahl unterbrochen wird (*I/O Signal* in Bild 5-12). Der vierte Sensor ist am *Zylinder* angebracht und erfasst dessen Endlage. Insgesamt sind drei Lichtschranken (vor jedem Lager) und vier Endlagenschalter (an jedem Zylinder) vorhanden.

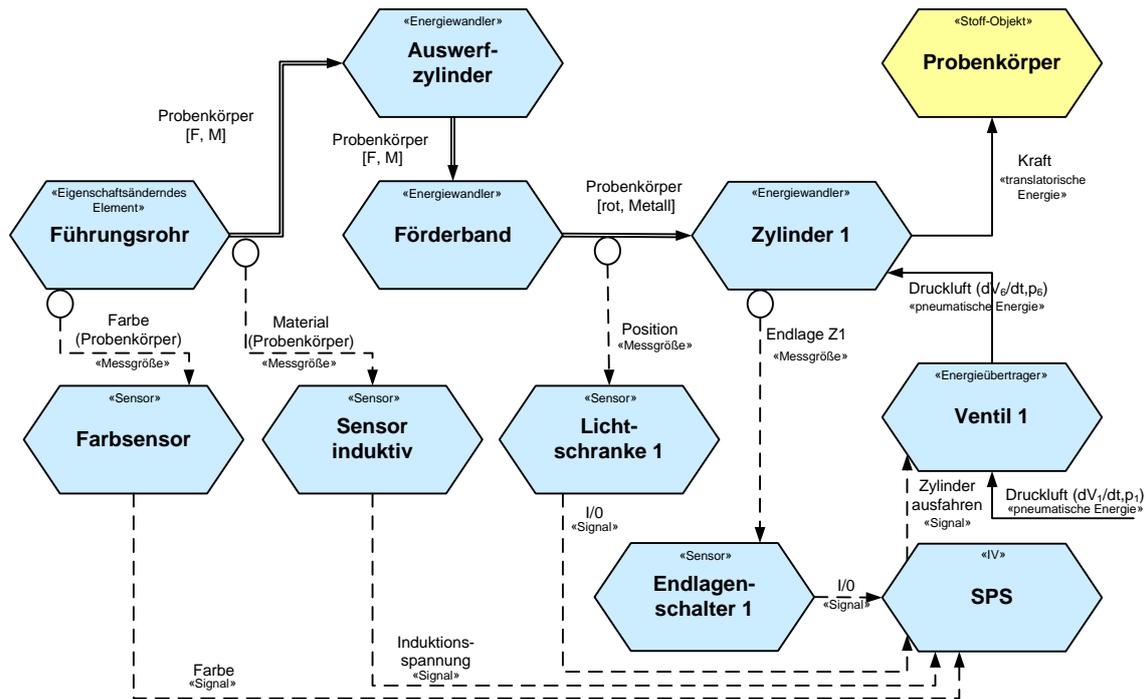


Bild 5-12: Systemstruktur der Sortieranlage – Informationsspezifische Sicht

5.2.2.4 Ergänzungen der Systemstruktur

Innerhalb der energiespezifischen Sicht (Bild 5-11) wurde die elektrische Energieversorgung noch nicht berücksichtigt. Um zu bestimmen, welche Elemente eine elektrische Versorgung benötigen, wird das Attribut *aktiv* (vgl. Kap. 4.2.2) vergeben. In diesem Fall trifft dies nur auf den *Motor* sowie auf die *Lichtschranken* zu. Der *Motor* hat jedoch durch den Hauptfluss bereits eine eingehende Beziehung vom Typ *«elektrische Energie»*. An die Elemente *Lichtschranke* werden die entsprechenden Beziehungen ergänzt. Die Energieversorgung erfolgt über das *Netzteil*. Bei den Ventilen ist in der Realisierung eine elektrische Verbindung vorhanden. Diese dient jedoch zur Ansteuerung der Ventile und wird daher durch den Informationsfluss beschrieben.

Zu den aufgeführten Elementen sind noch weitere Elemente vorhanden, die die Hauptfunktion „Bauteile tragen“ haben. Demnach sind diese Elemente vom Typ *«Tragstruktur»*. Hierzu gehören die *Unterplatte*, sowie die *Halterungsschiene* des *Führungrohrs*. Um die mechanischen Verbindungen weiter zu untersuchen, werden die Elemente um das Attribut *gestaltbehaftet* ergänzt. In diesem Fall erhalten alle Elemente das Attribut. Bei der Informationsverarbeitung wird dabei die Hardware im Schaltschrank mit berücksichtigt. Das gesamte Modell der Sortieranlage ist im Anhang dargestellt.

5.3 Bewertung der Arbeit an den Anforderungen

Das in Kapitel 4 vorgestellte *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme* wird nach der beispielhaften Anwendung (Kap. 5.1 und 5.2) anhand der aufgestellten Anforderungen (vgl. Kapitel 2.8) bewertet. Hierzu wird zu jeder Anforderung eine Übersicht gegeben, inwieweit das Rahmenwerk zur Erfüllung der Anforderung beiträgt.

A1) Sprachenunabhängig: Die gängigen Modellierungssprachen können eine Systemstruktur beschreiben. In dieser Arbeit wurde der Fokus auf die Sprachen SysML und CONSENS gelegt, da diese neben den Elementen die Wirkbeziehung abbilden können. Die in Kapitel 4.1 definierten Beziehungs- und Elementklassen lassen sich in beiden Sprachen modellieren. Hierzu kann die Angabe der Klasse gezielt durch Stereotype (SysML) oder über Klassifikationsmerkmale (CONSENS) erfolgen. An den Anwendungsbeispielen wurde gezeigt, dass die Konzepte mit beiden Sprachen eingesetzt werden können.

A2) Systemstruktur als Basis für ein einheitliches Systemverständnis: Die Definition der Systemstruktur legt den Fokus auf eine funktionale Beschreibung der Elemente und deren Wechselwirkungen. Eine Vermischung der Konzepte wird durch die eingeführte Klassifikation von Elementen und Beziehungen vermieden. An die Definition der Elementklassen sind zusätzlich Bedingungen der ein- und ausgehenden Beziehungen geknüpft.

Für die Klassifikation wurden bestehende Systemstrukturen analysiert, Grundfunktionen herangezogen sowie Begrifflichkeiten aus Mechatronik-Lehrbüchern verwendet. Damit wurde sichergestellt, dass keine Fachbegriffe einer spezifischen Fachdisziplin verwendet wurden. Mit den Steckbriefen werden beispielhafte Anwendungen aufgezeigt, so dass der Einstieg in die Modellierung erleichtert wird.

A3) Vergleichbarkeit: Zur Sicherstellung einer vergleichbaren Beschreibung der Systemstruktur wurden die darzustellenden Modellkonstrukte untersucht und klassifiziert. Darüber hinaus wurden Richtlinien formuliert, die anleiten, die Element- und Beziehungsklassen einheitlich zu wählen. Durch Richtlinien wird der Anwender angeleitet vergleichbar zu modellieren. Eine nachträgliche Prüfung erfolgt auf Basis der Checkliste.

A4) Vollständigkeit: Durch den Fokus auf mechatronische Systeme wurden Annahmen formuliert und in Richtlinien und Bedingungen überführt. Hierzu zählen die Annahmen, wie zum Beispiel, dass mindestens eine Energiequelle vorhanden sein muss. Dies führte dazu, dass zur Überprüfung dieser Bedingungen Attribute (aktiv, gestaltbehaftet und Energiequelle) eingeführt wurden. Über die Checkliste werden die Bedingungen systematisch abgefragt.

A5) Richtigkeit: Die Bedingungen an den ein- und ausgehenden Beziehungen der Elementklassen ermöglichen eine Überprüfung der Richtigkeit. Darüber hinaus wurden

Hilfsmittel bereitgestellt, die zur Prüfung der semantischen Richtigkeit herangezogen werden können. Hierzu zählt die Matrix der energieumsetzenden Elemente sowie der Sensoren.

A6) Klarheit: Die Klarheit wird mit dem Konzept der Sichtenbildung erreicht. Durch die Klassifikation der Elemente und Beziehungen können unterschiedliche Filter generiert werden, die spezifische Bestandteile der Systemstruktur visualisieren können. Beispielsweise kann eine rein gestaltorientierte Sicht erzeugt werden. Diese nimmt die visuell wahrgenommene Komplexität der vielen Elemente und Beziehungen heraus, indem nur gefilterte Elemente angezeigt oder hervorgehoben dargestellt werden. Erst die Klassifikation ermöglicht eine gezielte Filtergenerierung. Zusätzlich zu den Filtern werden Hierarchien zugelassen. Diese tragen ebenfalls zur Klarheit bei, indem pro Ebene die Informationen dargestellt werden. Die oberste Ebene stellt die Einbettung des Systems in seine Umgebung dar.

A7) Vereinbarkeit von intuitiver Anwendung und formalisierter Beschreibung: Das Vorgehensmodell zur Erstellung der Systemstruktur beschreibt ein Vorgehen, dass zu Beginn eine intuitive Anwendung der Modellierung beabsichtigt. So wird die interdisziplinäre Zusammenarbeit gefördert, indem Workshops durchgeführt werden. Der Aufwand, der hinter der formalisierten Beschreibung steckt, wird von den Fachexperten fern gehalten. Diese müssen die Klassifikation der Elemente nur auf oberer Ebene kennen und anwenden. Dabei wird die Rolle des Systems Engineers in die Modellierung einbezogen. Dieser leitet die Workshops und führt anschließend die Formalisierung und Überprüfung des Modells durch.

A8) Rechnerunterstützte Modellierung: Die Konzepte wurden so aufgebaut, dass die bestehenden Werkzeuge zur Modellierung genutzt werden können. Die Handhabung ist jedoch aufwendig und wenig anwendungsfreundlich. Daher wurde ein Konzept eines Softwarewerkzeugs vorgestellt, das die Klassifikation nutzt, um eine effiziente Modellierung zu unterstützen. Hierzu werden Modellierungsschritte automatisiert, die sich ausgehend von der Klassifikation bedingen. Zusätzlich zur Automatisierung wird damit von vornherein sichergestellt, dass die Vergleichbarkeit, Vollständigkeit und Richtigkeit sichergestellt werden. Über die automatische Plausibilitätsprüfung wird der Anwender auf Fehler hingewiesen.

Das erarbeitete *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme* erfüllt damit alle Anforderungen. In der Anwendung wurde darüber hinaus gezeigt, dass Bestandteile des Rahmenwerks aufeinander aufbauen, jedoch auch Einzelne eingesetzt werden können.

6 Resümee und Ausblick

Mechatronische Systeme zeichnen sich durch ihre Interdisziplinarität aus. Damit sind an der Entwicklung viele Fachdisziplinen beteiligt, die unterschiedliche Arbeits- und Denkweisen haben. Zur Vermeidung von Iterationen im Entwicklungsprozess und Unterstützung der besseren Zusammenarbeit, muss bei den beteiligten Entwicklern von Beginn an das Systemdenken gefördert werden. Hier greifen die Ansätze des Model-Based Systems Engineering. Die interdisziplinäre Systembeschreibung erfolgt in einem Systemmodell. Dieses steht im Mittelpunkt der Entwicklung und bietet den Fachdisziplinen ein Kommunikations- und Kooperationsmedium. Vor allem die Systemstruktur, als ein Aspekt des Systemmodells, bietet hohes **Nutzenpotential**. Dieses beschreibt die Elemente und Beziehungen des Systems und unterstützt damit ein einheitliches und ganzheitliches Systemverständnis bei den beteiligten Disziplinen. Darüber hinaus bietet die Systemstruktur eine Basis zur Dokumentation und Analyse von Auswirkungen bei disziplinübergreifenden Änderungen. Sie schafft Transparenz durch Visualisierung. Dadurch können frühzeitig Fehler identifiziert und vermieden werden.

Die Systemstruktur muss dabei disziplinübergreifend erstellt werden. Dies erfordert eine einfache und intuitive Anwendung von Methoden und Sprachen der Modellierung. Im Gegensatz dazu steht die rechnerbasierte Beschreibung zur besseren Auswertbarkeit und Handhabung der Systemstruktur. Der unterschiedliche Formalisierungsgrad stellt eine **Herausforderung** bei der Einführung und Anwendung des MBSE-Ansatzes. Dies wirkt sich auch auf die Inhalte der Systemstruktur aus. An der Modellerstellung können andere Personen beteiligt sein, als diejenigen, die die Systemstruktur lesen und nutzen werden. Daher muss die Systemstruktur plausibel sein und damit Sachverhalte vergleichbar, vollständig und richtig abbilden.

Im Rahmen dieser Arbeit wurden existierende MBSE-Ansätze untersucht. Dabei zeigte sich, dass die Modellierungsmethoden und -werkzeuge eine Strukturbeschreibung beinhalten, diese jedoch unterschiedlich ausgeprägt ist. Die Struktur wird teilweise funktional, logisch, physisch oder wirkungsorientiert beschrieben. Innerhalb einer Methode können verschiedene Formen (z.B. funktional und physisch) vorkommen. Eine Vorgabe der abzubildenden Elemente und Beziehung findet dabei unzureichend statt. So entstehen Modelle, die nicht vergleichbar sind. Zudem können daraus keine Aussagen zu Vollständigkeit oder Richtigkeit gemacht werden. Die Inhalte des Systemmodells sind somit von der Erfahrung des Modellierers abhängig. Damit wird das Lesen und Interpretieren der Modelle durch Externe (nicht an der Modellerstellung beteiligt) erschwert. Die untersuchten Ansätze gehen dabei nicht auf die Vereinbarkeit von intuitiver Anwendung und formalisierter Beschreibung ein. Daraus wurde der Handlungsbedarf an ein Rahmenwerk zur Modellierung einer plausiblen Systemstruktur für mechatronische Systeme abgeleitet.

Das Rahmenwerk umfasst im Kern die folgenden Inhalte

- **Vorgaben zur Modellierung:** Die Bestandteile einer Systemstruktur sind Elemente und deren Beziehungen. Hierzu wurden ausgehend von der Fokussierung auf mechatronische Systeme diese untersucht und klassifiziert. Daraus abgeleitet wurden Vorgaben an die Beschreibung der Systemstruktur erarbeitet und in Form von Richtlinien und Bedingungen formuliert. Damit wird der Modellierer angeleitet, bei der Erstellung auf Plausibilität zu achten.
- **Überprüfung der Modellinhalte:** Mit der Klassifikation der Elemente wurde eine Basis geschaffen, die Modellinhalte auszuwerten. Dabei dient eine Checkliste zur Überprüfung der Systemstruktur auf die Einhaltung der Richtlinien und Bedingungen. Damit kann eine Aussage über die Vergleichbarkeit, Vollständigkeit und Richtigkeit der Systemstruktur gemacht werden. Ergänzend hierzu wurden Hilfsmittel bereitgestellt, die bei der Überprüfung der Richtigkeit genutzt werden können.
- **Softwaretechnischen Unterstützung:** Das Konzept zur Werkzeugunterstützung beschreibt die rechnerbasierte Modellierung der Systemstruktur. Dabei sind die Element- und Beziehungsklassen sowie die Bedingungen im Metamodell implementiert. Hierdurch wird eine effiziente Modellierung erreicht, da Modellierungsschritte sowie die Plausibilitätsprüfung automatisiert werden.
- **Strukturiertes Vorgehensmodell:** Das Vorgehensmodell beschreibt die Erstellung der plausiblen Systemstruktur. Dabei findet eine Trennung zwischen der Erarbeitung der Systemstruktur im Rahmen von disziplinübergreifenden Workshops und der formalisierten, rechnerbasierten Modellierung statt.

Die **Anwendung** des Rahmenwerks erfolgte an zwei Beispielen. Das erste Anwendungsbeispiel zeigte die Erstellung der Systemstruktur anhand des aufgestellten Vorgehensmodells. Das zu entwickelnde Produkt war eine Tretkraftunterstützung für ein Fahrrad. Die Erfahrungen aus ähnlichen Industrieprojekten flossen in die Beschreibung des Vorgehens ein. Das zweite Beispiel war eine Sortieranlage – ein bereits realisiertes System. Die verschiedenen Sichten auf die Systemstruktur wurden nacheinander vorgestellt und daraus eine gesamte Systemstruktur erarbeitet. In der Bewertung der Arbeit konnte aufgezeigt werden, dass die aus der Problemanalyse abgeleiteten Anforderungen erfüllt wurden.

Für eine effiziente Entwicklung mechatronischer Systeme besteht weiterer Forschungsbedarf. Das Ziel zukünftiger Forschungsarbeiten ist die *neue Schule des Entwurfs für Intelligente Technische Systeme*. Sie umfasst die starke Verzahnung aller am Entwicklungsprozess beteiligter Stakeholder zur ganzheitlichen und durchgängigen Arbeitsweise in der Produktentstehung. Die vorliegende Arbeit liefert einen Beitrag zur Befähigung Dritter, das Systems in einer plausiblen Systemstruktur ganzheitlich zu beschreiben. Die Systemstruktur ist Ausgangsbasis für die Entwicklungstätigkeiten in den Disziplinen. Dabei müssen die Werkzeuge zur Systemmodellierung eine Verzahn-

nung der disziplinspezifischen Modelle ermöglichen. Dieses ist das Bindeglied zwischen allen Fachdisziplinen. Die datentechnische Kopplung führt zur durchgängigen Virtualisierung des Produktentstehungsprozesses. Damit wird unter anderem das disziplinübergreifende Abspeichern und Wiederverwenden von bewährten Lösungen ermöglicht.

Darüber hinaus ist dies der Schritt von intelligenten Produkten zur intelligenten Entwicklung. Hierzu können die bereits in der Technik eingesetzten Methoden der künstlichen Intelligenz auf die Entwicklung übertragen werden. So können Verfahren, wie Mustererkennung eingesetzt werden, um so z.B. wiederkehrende Gesetzmäßigkeiten disziplinübergreifend zu identifizieren.

Mit den neuen Ansätzen sind Änderungen in den bisherigen Denk- und Arbeitsweisen verbunden. Der Weg zur intelligenten Entwicklung erzeugt Akzeptanzprobleme. Die Entwickler sehen ihre Rolle, respektive ihre Stellung im Unternehmen gefährdet. Auch die Auswirkungen auf die Organisationsstruktur sind nicht ersichtlich. Umso wichtiger ist eine ganzheitliche Betrachtung der Faktoren: Mensch, Organisation, Technik.

7 Abkürzungsverzeichnis

BE	Bedieneinheit
bspw.	beispielsweise
bzw.	beziehungsweise
CA	Classified Ads Systems Engineering
CAD	Computer Aided Design
CI	Customer Interface
CPS	Cyber-Physical Systems
CO	Coordinator
CONSENS	CONceptual design Specification technique for the Engineering of complex Systems
engl.	Englisch
EA	Enterprise Architect
EVA	Eingabe-Verarbeitung-Ausgabe
FAS	Functional Architecture for Systems
FEM	Fenite Elemente Methode
FMEA	Failure Mode and Effects Analysis
G	Glue
IM	Information Manager
INCOSE	International Council on Systems Engineering
IT	Informationstechnik
it's OWL	Intelligente Technische Systeme OstWestfalenLippe
HMI	Human Machine Interface
Kap.	Kapitel
LO	Logistics and Operations
MBSE	Model-Based Systems Engineering (modellbasiertes Systems Engineering)
MCD	Mechatronic Concept Designer
MKS	Mehrkörpersimulation

SA	System Analyst
SD	System Designer
SE	Systems Engineering
SysML	Systems Modeling Language
OOSEM	Object Oriented Systems Engineering Method
OPM	Object-Process Methodology
PDM	Product Data Management
PE	Process Engineer
PLM	Product Lifecycle Management
QFD	Quality-Function-Deployment
RO	Requirements Owner
TKU	Tretkraftunterstützung
TM	Technical Manager
TU	Technische Universität
vgl.	vergleiche
VireS	Virtuelle Synchronisation von Produktentwicklung und Produktionssystem-entwicklung
V&V	Verifikation und Validierung, engl.: Verification and Validation
z.B.	zum Beispiel

8 Literaturverzeichnis

- [aca09] acatech – Deutsche Akademie der Technikwissenschaften: Intelligente Objekte - klein, vernetzt, sensitiv – Eine neue Technologie verändert die Gesellschaft und fordert zur Gestaltung heraus. Springer, Berlin, Heidelberg, 2009
- [Abr05] ABRAMOVICI, M.: PLM: Kern künftiger Unternehmensstrategien – Engineering-Prozesse auf dem Prüfstand. Intelligenter produzieren, 2/2005, VDMA-Verlag, S. 6-9
- [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 – Definitionen, Anwendungen, Konzepte. HNI-Verlagsschriftenreihe, Band 234, Paderborn, 2009
- [ALR12] ALBERS, A.; LOHMEYER, q.; RADIMERSKY, A.: Individuelle und organisatorische Akzeptanz von Methoden des Systems Engineering. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [AZ11] ALBERS, A.; ZINGEL, C.: Interdisciplinary Systems Modeling Using The Contact & Channel-Model for SysML. In: Proceedings of the 18th International Conference on Engineering Design (ICED'11), Copenhagen, 2011
- [Alt12a] ALT, O.: Bessere Systemspezifikationen durch Einsatz von SysML-Wirkkettenarchitekturen. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [Alt12b] ALT, O.: Modell-basierte Systementwicklung mit SysML – In der Praxis. Carl Hanser Verlag, München, 2012
- [AHJ+09] ANDERSSON, H.; HERZOG, E.; JOHANSSON, G.; JOHANSSON, O.: Experience from introducing unified modeling language/systems modeling language at Saab Aerosystems. Systems Engineering, Wiley Online Library2009
- [BBK+09] BALZERT, H.; BALZERT, H.; KOSCHKE, R.; LÄMMEL, U.; LIGGESMEYER, P.; QUANTE, J.: Lehrbuch der Software-Technik. Spektrum, Heidelberg, 3. Auflage, 2009
- [BPV12] BECKER, J.; PROBANDT, W.; VERING, O.: Grundsätze ordnungsmäßiger Modellierung – Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement. Springer Gabler, Berlin, Heidelberg, 2012
- [Ben05] BENDER, K.: Embedded Systems – Qualitätsorientierte Entwicklung. Springer, Berlin, 2005
- [BGJ+09] BERTSCHE, B.; GÖHNER, P.; JENSEN, U.; SCHINKÖTHE, W.; WUNDERLICH, H.-J.: Zuverlässigkeit mechatronischer Systeme. Springer, Berlin, 2009
- [Bir80] BIRKHOFFER, H.: Analyse und Synthese der Funktionen technischer Produkte. VDI-Verlag, Düsseldorf, Nr. 70, 1980
- [BC10] BONE, M.; CLOUTIER, R. J.: The Current State of Model Based Systems Engineering. In: Proceedings of the 8th Conference on Systems Engineering Research, 2010
- [BVB12] BONNEMA, G. M.; VEENVLIET, K. T.; BROENINK, J. F.: Systems Design and Engineering – Lubricating Multidisciplinary Development Projects. University of Twente, Twente, 2012
- [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>, Oktober 2013

- [BFG+08] BROY, M.; FEILKAS, M.; GRÜNBAUER, J.; GRULER, A.; HARHURIN, A.; HARTMANN, J.; PENZENSTADLER, B.; SCHÄTZ, B.; WILD, D.: Umfassendes Architekturmodell für das Engineering eingebetteter Software-intensiver Systeme. Institut für Informatik der Technischen Universität München, München, 2008
- [Czi08] CZICHOS, H.: Mechatronik – Grundlagen und Anwendungen technischer Systeme. Vieweg+Teubner Verlag, Wiesbaden, 2. Auflage, 2008
- [DIN1319] Grundlagen der Messtechnik. Teil 1: Grundbegriffe, 1995
- [DIN15194] Fahrräder – Elektromotorisch unterstützte Räder – EPAC-Fahrräder, 2009
- [Dor02] DORI, D.: Object-process methodology – A holistics systems paradigm. Springer, Berlin, New York, 2002
- [Dud13-ol] Duden: online – Rechtschreibung, Grammatik und Bedeutung eines Wortes, unter: <http://www.duden.de/woerterbuch>, Oktober 2013
- [Dum11] DUMITRESCU, R.: Eine Entwicklungssystematik zur Integration kognitiver Funktionen in fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 286, Paderborn, 2011
- [EGG+12] EIGNER, M.; GERHARDT, F.; GILZ, T.; MOGO NEM, F.: Informationstechnologie für Ingenieure. Springer Vieweg, Berlin, 2012
- [ES09] EIGNER, M.; STELZER, R.: Product Lifecycle Management – Ein Leitfaden für Product Development und Life Cycle Management. Springer, Berlin, 2. neu bearbeitete Auflage, 2009
- [Est08] ESTEFAN, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. In: INCOSE MBSE Initiative, California, 2008
- [EU 02] Eu Richtlinie 2002/24/EG: Über die Typgenehmigung für zweirädrige oder dreirädrige Kraftfahrzeuge und zur Aufhebung der Richtlinie 92/61/EWG des Rates. Europäisches Parlament und Rat, 2002
- [ES05] EVERSHEIM, W.; SCHUH, G.: Integrierte Produkt- und Prozessgestaltung. Springer, Berlin, 2005
- [Fra06] FRANK, U.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 175, Paderborn, 2006
- [FHH+05] FRANKE, H.-J.; HUCH, B.; HERMANN, C.; LÖFFLER, S.: Kooperationsorientiertes Innovationsmanagement – Ergebnisse des BMBF-Verbundprojektes GINA "Ganzheitliche Innovationsprozesse in modularen Unternehmensnetzwerken". Logos-Verl, Berlin, 2005
- [FMS12] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A practical guide to SysML – The systems modeling language. Morgan Kaufmann, Waltham, 2. Auflage, 2012
- [FK13-ol] Fritz, J.; Kickstein, J.: Systems Engineering und PLM – Modellzentriertes Vorgehen im Innovationsprozess, unter: http://www.contact-software.com/fileadmin/user_upload/de/Medien/White_Paper/pdf/WP_Systems_Engineering_PLM_de.pdf, Oktober 2013
- [Gau10] GAUSEMEIER, J. (Hrsg.): Frühzeitige Zuverlässigkeitsanalyse mechatronischer Systeme. Carl Hanser Verlag, München, 2010
- [GAC+13] GAUSEMEIER, J.; ANACKER, H.; CZAJA, A.; WASSMANN, H.; DUMITRESCU, R.: Auf dem Weg zu intelligenten technischen Systemen. In: Gausemeier, J.; Dumitrescu, R.; Rammig, F.-J.; Schäfer, W.; Trächtler, A. (Hrsg.): 9. Paderborner Workshop Entwurf mechatronischer Systeme, HNI Verlagsschriftenreihe, Band 310, 18.-19. April, Paderborn 2013
- [GCW+13] GAUSEMEIER, J.; CZAJA, A.; WIEDERKEHR, O.; DUMITRESCU, R.; TSCHIRNER, C.; STEFFEN, D.: Systems Engineering in der industriellen Praxis. In: Gausemeier, J.; Dumitrescu, R.; Rammig, F.-J.; Schäfer, W.; Trächtler, A. (Hrsg.): 9. Paderborner Workshop

- Entwurf mechatronischer Systeme, HNI Verlagsschriftenreihe, Band 310, 18.-19. April, Paderborn 2013
- [GEK01] GAUSEMEIER, J.; EBBESMEYER, P.; KALLMEYER, F.: Produktinnovation – Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag, München, 2001
- [GFD+08] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus. In: Konstruktion, Ausgabe 7/8-2008 und 9/-2008, Springer VDI-Verlag, Düsseldorf, 2008
- [GFD+09] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Specification technique for the description of self-optimizing mechatronic systems. In: Research in Engineering Design, Vol. 20, Number 4, November 2009, Springer, London, 2009
- [GKP09] GAUSEMEIER, J.; KAISER, L.; POOK, S.: FMEA von komplexen mechatronischen Systemen auf Basis der Spezifikation der Prinziplösung. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 11/2009, S. 1011–1017
- [GLL12] GAUSEMEIER, J.; LANZA, G.; LINDEMANN, U.: Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung. Hanser Verlag, München, 2012
- [GPW09] GAUSEMEIER, J.; PLASS, C.; WENZELMANN, C.: Zukunftsorientierte Unternehmensgestaltung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. Hanser, München, 2009
- [GSG+09] GAUSEMEIER, J.; SCHÄFER, W.; GREENYER, J.; KAHL, S.; POOK, S.; RIEKE, J.: Management of Cross-Domain Model Consistency During the Development of Advanced Mechatronic Systems. In: Proceedings of 17th International Conference On Engineering Design, 24-27 August, Stanford, USA, 2009
- [Geh05] GEHRKE, M.: Entwurf mechatronischer Systeme auf Basis von Funktionshierarchien und Systemstrukturen. Dissertation, Institut für Informatik, Universität Paderborn, 2005
- [GG05] GERLICH, R.; GERLICH, R.: 111 thesen zur erfolgreichen softwareentwicklung – Argumente und entscheidungshilfen für manager konzepte und anleitungen für praktiker. Springer, Berlin, 2005
- [Göh13] GÖHNER, P.: Agentensysteme in der Automatisierungstechnik. Springer, Berlin, 2013
- [HWF+12] HABERFELLNER, R.; WECK DE, O. L.; FRICKE, E.; VÖSSNER, S.: Systems Engineering – Grundlagen und Anwendung. Orell Füssli, Zürich, 2012
- [HTF96] HARASHIMA, F.; TOMIZUKA, M.; FUKUDA, T.: Mechatronics – "What Is It, Why and How?" An Editorial. IEEE/ASME Transactions on Mechatronics, Volume 1, Nr. 1, 1996
- [HMW02] HIRTZ, JULIE, STONE, ROBERT B.; MCADAMS, DANIEL A., SZYKMAN, SIMON; WOOD, K.: A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. In: Research in Engineering Design, Springer, 2002
- [Hit07] HITCHINS, D. K.: Systems engineering – A 21st century systems methodology. John Wiley, West Sussex, England, 2007
- [HN09] HOGREBE, F.; NÜTTGENS, M.: Rahmenkonzept zur Messung und Bewertung der Gebrauchstauglichkeit von Modellierungssprachen. In: Nüttgens, M. (Hrsg.): Arbeitsberichte zur Wirtschaftsinformatik, Nr. 7, 2009
- [Hon04] HONOUR, E.: Understanding the Value of Systems Engineering. In: INCOSE (Hrsg.): Proceedings of the 14th Annual INCOSE International Symposium, 2004
- [Hua02] HUANG, M.: Funktionsmodellierung und Lösungsfindung mechatronischer Produkte. Dissertation, Fakultät für Maschinenbau, Universität Karlsruhe, Forschungsberichte aus dem Institut für Rechneranwendung in Planung und Konstruktion, Shaker Verlag, Aachen, 2002

- [INC07-ol] INCOSE: Systems Engineering Vision 2020, unter: http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf, Oktober 2013
- [INC12] INCOSE: INCOSE Systems Engineering Handbuch – Deutsche Übersetzung. GfSE, 2012
- [Ise02] ISERMANN, R.: Mechatronische Systeme – Grundlagen. Springer, Berlin, 2002
- [ISO25010] Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE), 2011
- [Jan10] JANSCHKE, K.: Systementwurf mechatronischer Systeme – Methoden, Modelle, Konzepte. Springer, Heidelberg, 2010
- [JLS11] Ji, H.; LENORD, O.; SCHRAMM, D.: A Model Driven Approach for Requirements Engineering of Industrial Automation Systems. In: 4th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, ETH Zürich, Switzerland, 2011
- [JWK+08] JOHANSSON, L.-O.; WÄRJA, M.; KJELLIN, H.; CARLSSON, S.: Graphical modeling techniques and usefulness in the Model Driven Architecture: Which are the criteria for a “good” Computer independent model? In: Proceedings of The 31st Information Systems Research Seminar in Scandinavia (IRIS31), 2008
- [Jür00] JÜRGENS, U.: New product development and production networks – Global industrial experience. Springer, Berlin, 2000
- [KDH+13] KAISER, L.; DUMITRESCU, R.; HOLTMANN, J.; MEYER, M.: Automatic Verification of Modeling Rules in Systems Engineering for Mechatronic Systems. In: Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 4.-7. August, Portland, Oregon, 2013
- [Kal98] KALLMEYER, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 42, Paderborn, 1998
- [KWH+11-ol] KARBAN, R.; WEILKIENS, T.; HAUBER, R.; ZAMPARELLI, M.; DIEKMANN, R.; HEIN, A.: MBSE Initiative – SE2 Challenge Team – Cookbook for MBSE with SysML, unter: <http://www.mbse.gfse.de>, August 2013
- [Kle13] KLEINER, S.: Modellbasierte Entwicklung mechatronischer Systeme auf Basis V6 Systems. In: Gausemeier, J.; Dumitrescu, R.; Rammig, F.-J.; Schäfer, W.; Trächtler, A. (Hrsg.): 9. Paderborner Workshop Entwurf mechatronischer Systeme, HNI Verlagsschriftenreihe, Band 310, 18.-19. April, Paderborn 2013
- [KK94] KOLLER, R.; KASTRUP, N.: Prinziplösungen zur Konstruktion technischer Produkte. Springer, Berlin, 1994
- [KLW11] KORFF, A.; LAMM, J. G.; WEILKIENS, T.: Werkzeuge für den Schmied funktionaler Architekturen. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2011
- [KS03] KOSSIAKOFF, A.; SWEET, W.: Systems Engineering – Principles and practice. Wiley-Interscience, Hoboken, N.J., 2003
- [LW10] LAMM, J. G.; WEILKIENS, T.: Funktionale Architekturen in SysML. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2010
- [Lan99] LANGLOTZ, G.: Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte. Dissertation, Fakultät für Maschinenbau, Universität Karlsruhe, 1999
- [LGS+11] LI, F.; GILZ, T.; STEINHAUER, M.; VOGEL-HEUSER, B.; EIGNER, M.; SHEA, K.: Supporting the multi-domain plant engineering process using engineering knowledge from formalized model-based libraries. In: Spath, D.; Ilg, R.; Krause, T. (Hrsg.): ICPR 21, Fraunhofer IAO, Stuttgart, 2011

- [Lin09] LINDEMANN, U.: Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden. Springer, Berlin, 3. Auflage, 2009
- [LSO06] LINHARES, M.; SILVA, A.; OLIVEIRA, R.: Empirical Evaluation of SysML through the Modeling of an Industrial Automation Unit. In: 11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic, 20-22 September, 2006
- [Möh12] MÖHRINGER, S.: Systems Engineering im Mittelstand – ein flexibles Modell zur Rollverteilung – Praxiserfahrungen aus Entwicklungsprojekten. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [Moo09] MOODY, D.: The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering 6/2009, S. 756–779
- [Moo07] MOODY, D. L.: What Makes a Good Diagram? Improving the Cognitive Effectiveness of Diagrams in IS Development. In: Advances in information systems development. New methods and practice for the networked society, Springer, New York, 2007
- [Neg06] NEGELE, H.: Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung. Herbert Utz Verlag, München, 2. Auflage, 2006
- [Nor12] NORDSIEK, D.: Systematik zur Konzipierung von Produktionssystemen auf Basis der Prinziplösung mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 304, Paderborn, 2012
- [OMG12] OMG: Systems Modeling Language (OMG SysML™) – Version 1.3, 2012
- [PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung. Springer, Berlin, 7. Auflage, 2007
- [Pat82] PATZAK, G.: Systemtechnik, Planung komplexer innovativer Systeme – Grundlagen, Methoden, Techniken. Springer, Berlin, New York, 1982
- [PM06] PETRASCH, R.; MEIMBERG, O.: Model Driven Architecture – Eine praxisorientierte Einführung in die MDA. dpunkt Verlag, Heidelberg, 2006
- [PHA+12] POHL, K.; HÖNNINGER, H.; ACHATZ, R.; BROY, M.: Model-based engineering of embedded systems – The SPES 2020 methodology. Springer, Berlin, 2012
- [PL11] PONN, J.; LINDEMANN, U.: Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltlösungen. Springer, Berlin, Heidelberg, 2. Auflage, 2011
- [RFB12] RAMOS, A. L.; FERREIRA, J.; BARCELO, J.: Model-Based Systems Engineering: An Emerging Approach for Modern Systems. In: Proceedings of IEEE Transactions on Systems, Man and Cybernetics, 2012
- [RPC+12] REICHWEIN, A.; PAREDIS, C. J.; CANEDO, A.; WITSCHER, P.; STELZIG, P. E.; VOTINTSEVA, A.; WASGINT, R.: Maintaining Consistency between System Architecture and Dynamic System Models with SysML4Modelica. In: Proceedings of 6th International Workshop on Multi-Paradigm Modeling - MPM'12, Innsbruck, Austria, 2012
- [Rod12] RODDECK, W.: Einführung in die Mechatronik. Vieweg+Teubner Verlag, Wiesbaden, 4. Auflage, 2012
- [Rop75] ROPOHL, G.: Systemtechnik – Grundlagen und Anwendung. Carl Hanser Verlag, München, 1975
- [Rop09] ROPOHL, G.: Allgemeine Technologie – Eine Systemtheorie der Technik. Universitätsverlag Karlsruhe, Karlsruhe, 2009

- [Rot00] ROTH, K.: Konstruieren mit Konstruktionskatalogen. Springer, Berlin, Band 1, 3. Auflage, 2000
- [SA00] SAGE, A. P.; ARMSTRONG, J. E.: Introduction to systems engineering. Wiley, New York, 2000
- [SRC10] SCHALLES, C.; REBSTOCK, M.; CREAGH, J.: Ein generischer Ansatz zur Messung der Benutzerfreundlichkeit von Modellierungssprachen. In: Engels, G. (Hrsg.): Modellierung 2010, GI, Bonn, 2010
- [SFP+09] SCHAMAI, W.; FRITZSON, P.; PAREDIS, C.; POP, A.: Towards Unified System Modeling and Simulation with ModelicaML: Modeling of Executable Behavior Using Graphical Notations. In: Proceedings of the 7th Modelica Conference, Como, Italy, 2009
- [SBD+10] SCHATTEN, A.; BIFFL, S.; DEMOLSKY, M.; GOSTISCHA-FRANTA, E.; ÖSTREICHER, T.; WINKLER, D.: Best Practice Software-Engineering – Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen. Spektrum, Heidelberg, 2010
- [Sen13] SENDLER, U.: Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM. Springer, München, 2013
- [She96] SHEARD, S. A.: Twelve Systems Engineering Roles. In: INCOSE (Hrsg.): Proceedings of the 6th INCOSE Annual International Symposium, Boston, Massachusetts, USA, 1996
- [SB09] Stetter, R.; Blum, T.: Verschläft der Deutsche Maschinenbau seine Chancen? – Mechanische Möglichkeiten im internationalen Wettbewerb, 2009
- [Sta73] STACHOWIAK, H.: Allgemeine Modelltheorie. Springer, Wien, 1973
- [SV07] STAHL, T.; VÖLTER, M.: Modellgetriebene Softwareentwicklung – Techniken, Engineering, Management. dpunkt Verlag, Heidelberg, 2. Auflage, 2007
- [SMM+12] STIEGLER, A.; MALETZ, M.; MROTZEK, M.; WECK, T.: Generierung eines multiperspektiven Systemmodells in der automobilen Antriebsstrangentwicklung – Herausforderungen und Erfahrungen. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [TG10] TRETOW, G.; GÖPFERT, J.: Systems-Engineering in der frühen Phase der Produktentwicklung. CAD-CAM Report 4/2010, S. 58–61
- [TGH08] TRETOW, G.; GÖPFERT, J.; HEESE, C.: In sieben Schritten systematisch entwickeln. CAD-CAM Report 8/2008, Hoppenstedt Publishing, S. 772–778
- [UP95] ULRICH, H.; PROBST, GILBERT J. B.: Anleitung zum ganzheitlichen Denken und Handeln – Ein Brevier für Führungskräfte. Haupt, Bern, 4. unveränd. Auflage, 1995
- [VDI2222] Konstruktionsmethodik - Methodisches Entwickeln von Lösungsprinzipien, 1997
- [VDI2206] Entwicklungsmethodik für mechatronische Systeme, 2004
- [Wei06] WEILKIENS, T.: Systems Engineering mit SysML/UML – Modellierung, Analyse, Design. Dpunkt-Verlag, Heidelberg, 2006
- [ZVE09] ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie (Hrsg.): Nationale Roadmap Embedded Systems. Kompetenzzentrum Embedded Software & Systems, Frankfurt, 2009

Anhang

Inhaltsverzeichnis	Seite
A1 Steckbriefe der Elementklassen	A-1
A2 Liste der Richtlinien und Bedingungen	A-9
A3 Matrix der energieumsetzenden Elemente	A-12
A4 Matrix der Sensoren	A-13
A5 Aufgaben und Fähigkeiten eines Systems Engineers	A-14
A6 Systemstruktur der Sortieranlage	A-15

A1 Steckbriefe der Elementklassen

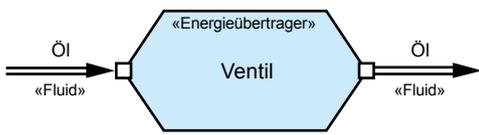
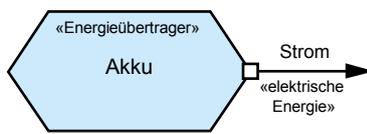
Klassifikation «Energieübertrager»	
<p>Beschreibung</p> <p>Ein Energieübertrager hat eine eingehende und ausgehende Beziehung vom Typ Energie. Die Energieform am Ein- und Ausgang ist identisch.</p>	<p>Beispielhafte Anwendung</p> 
<p>Grundfunktion</p> <ul style="list-style-type: none"> - Wert einer Energiekomponente verändern - Richtung einer Energiekomponente ändern - Sammeln/Trennen von Energiekomponenten 	
<p>Häufige Fehler</p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;">  </div> <div style="width: 50%;"> <p>Die ein- und ausgehende Beziehung ist nicht vom Typ «Energie». Das Öl ist das Medium zur Übertragung der Energie und wird daher in der Systemstruktur nicht als Stoff-Beziehung dargestellt.</p> </div> </div> <hr/> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;">  </div> <div style="width: 50%;"> <p>Die eingehende Beziehung fehlt. Der Akku kann wieder geladen werden und benötigt daher auch eine eingehende Energie-Beziehung vom Typ «elektrische Energie». Durch die statische Sicht auf die Struktur wird der zeitliche Zusammenhang zwischen ein- und Ausgang nicht beschrieben.</p> </div> </div>	

Bild A-1: Steckbrief Energieübertrager

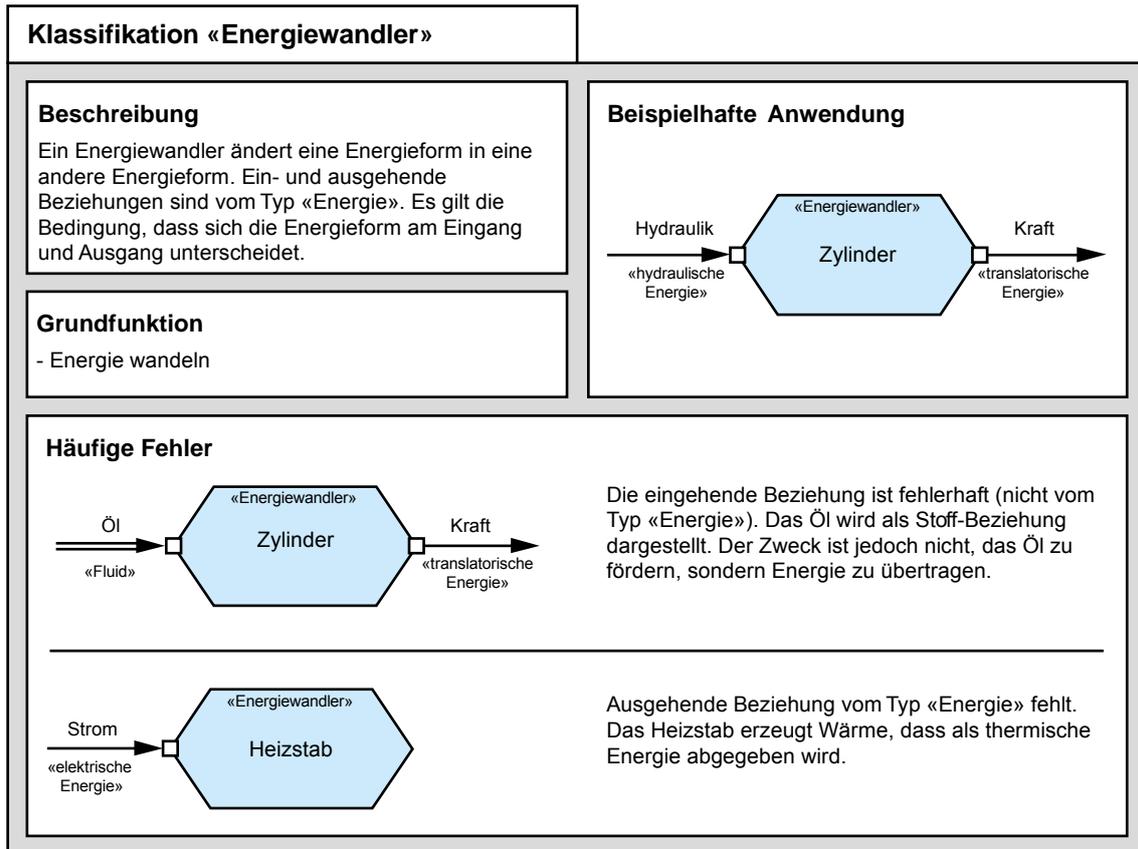


Bild A-2: Steckbrief Energiewandler

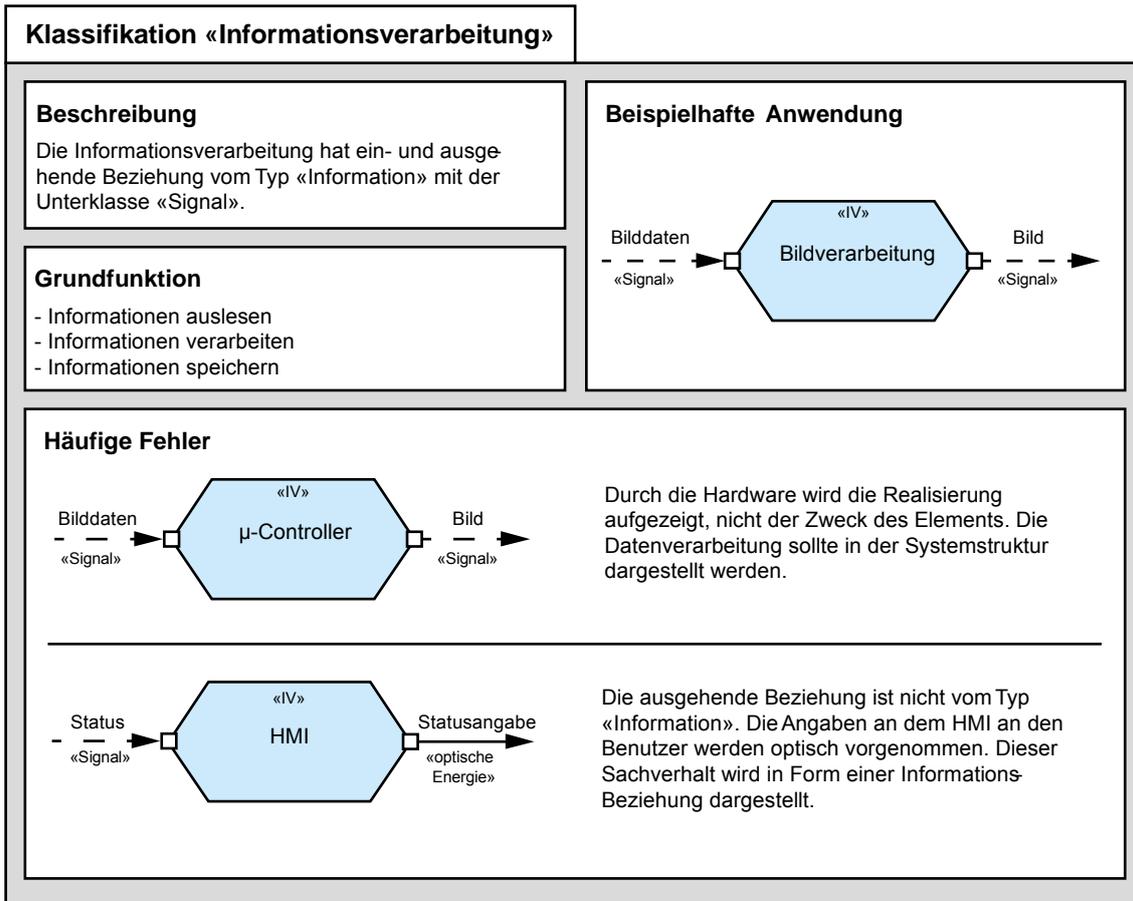


Bild A-3: Steckbrief Informationsverarbeitung

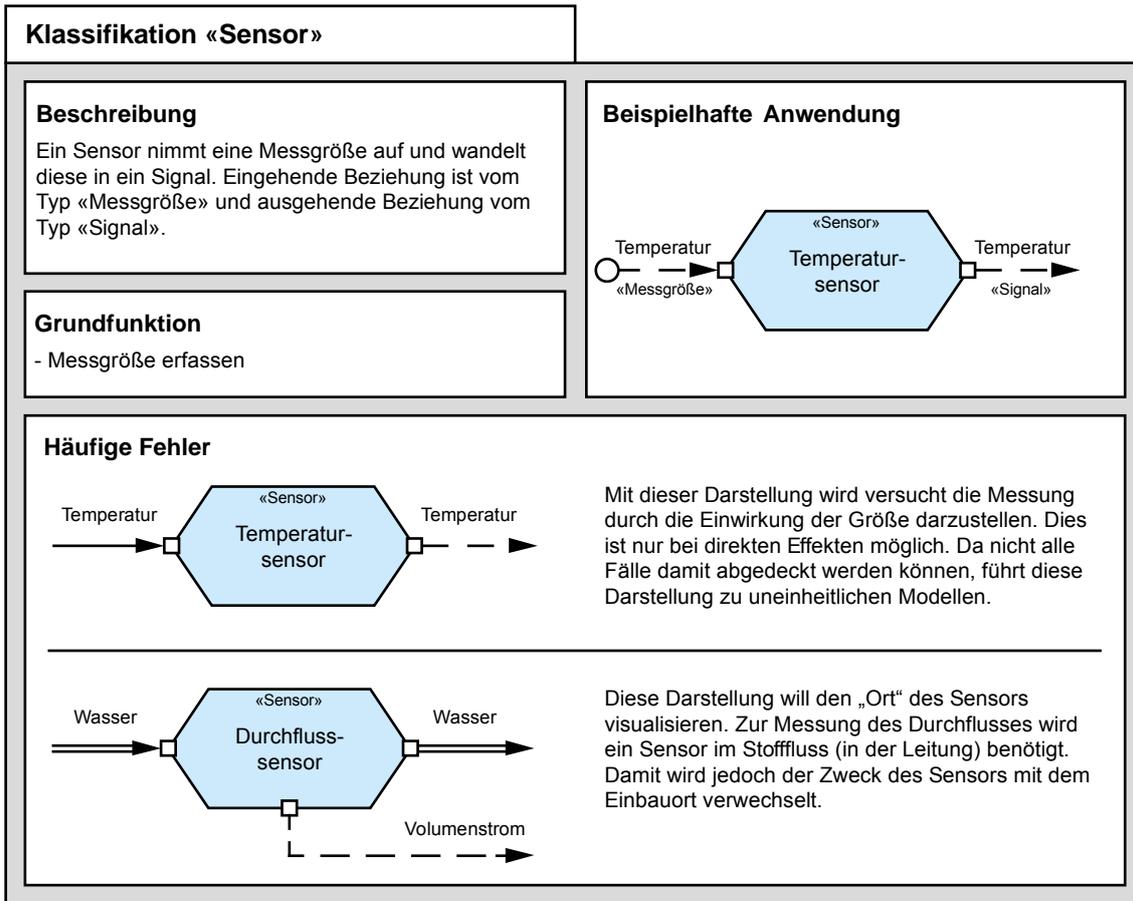


Bild A-4: Steckbrief Sensor

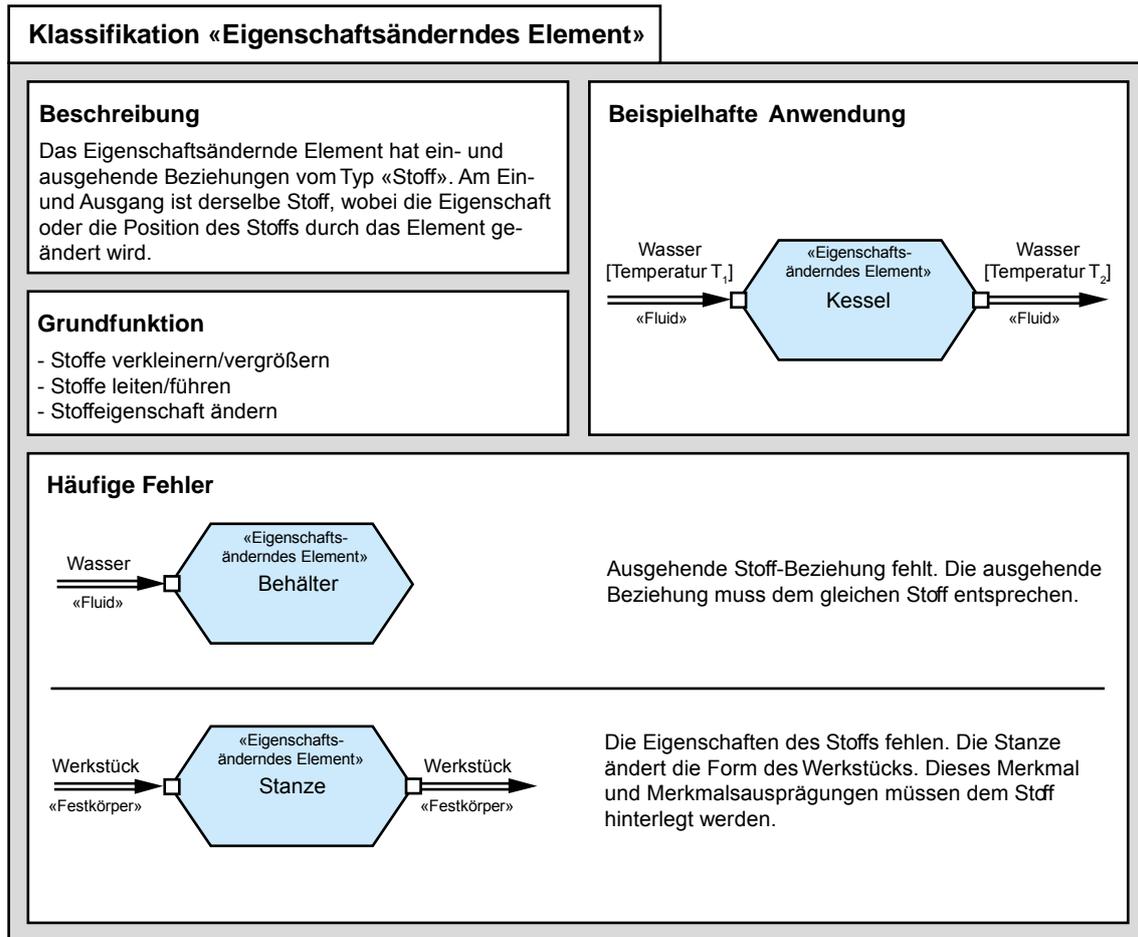


Bild A-5: Steckbrief Eigenschaftsänderndes Element

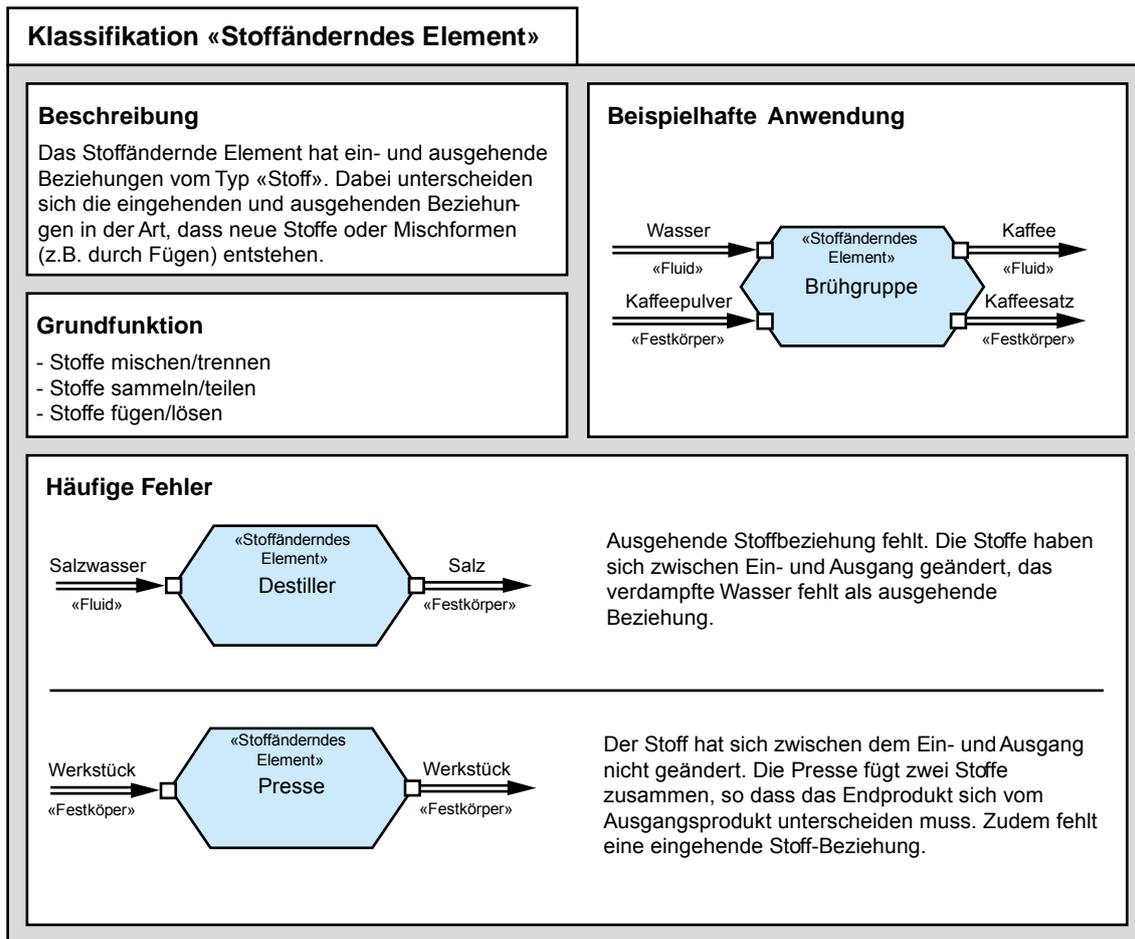


Bild A-6: Steckbrief Stoffänderndes Element

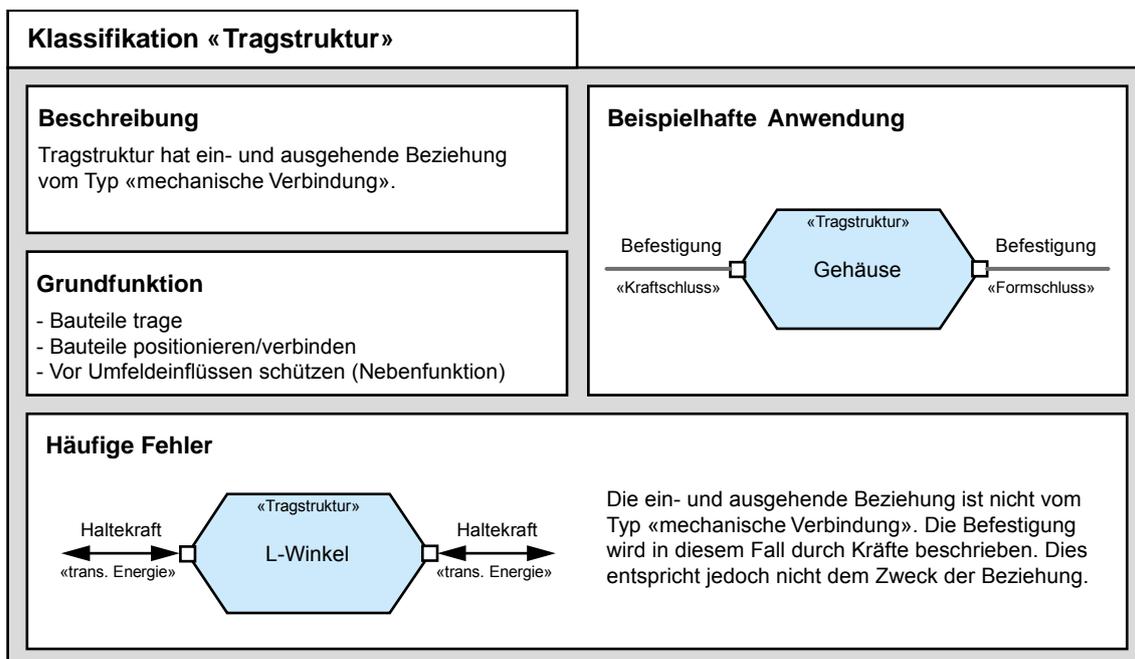


Bild A-7: Steckbrief Tragstruktur

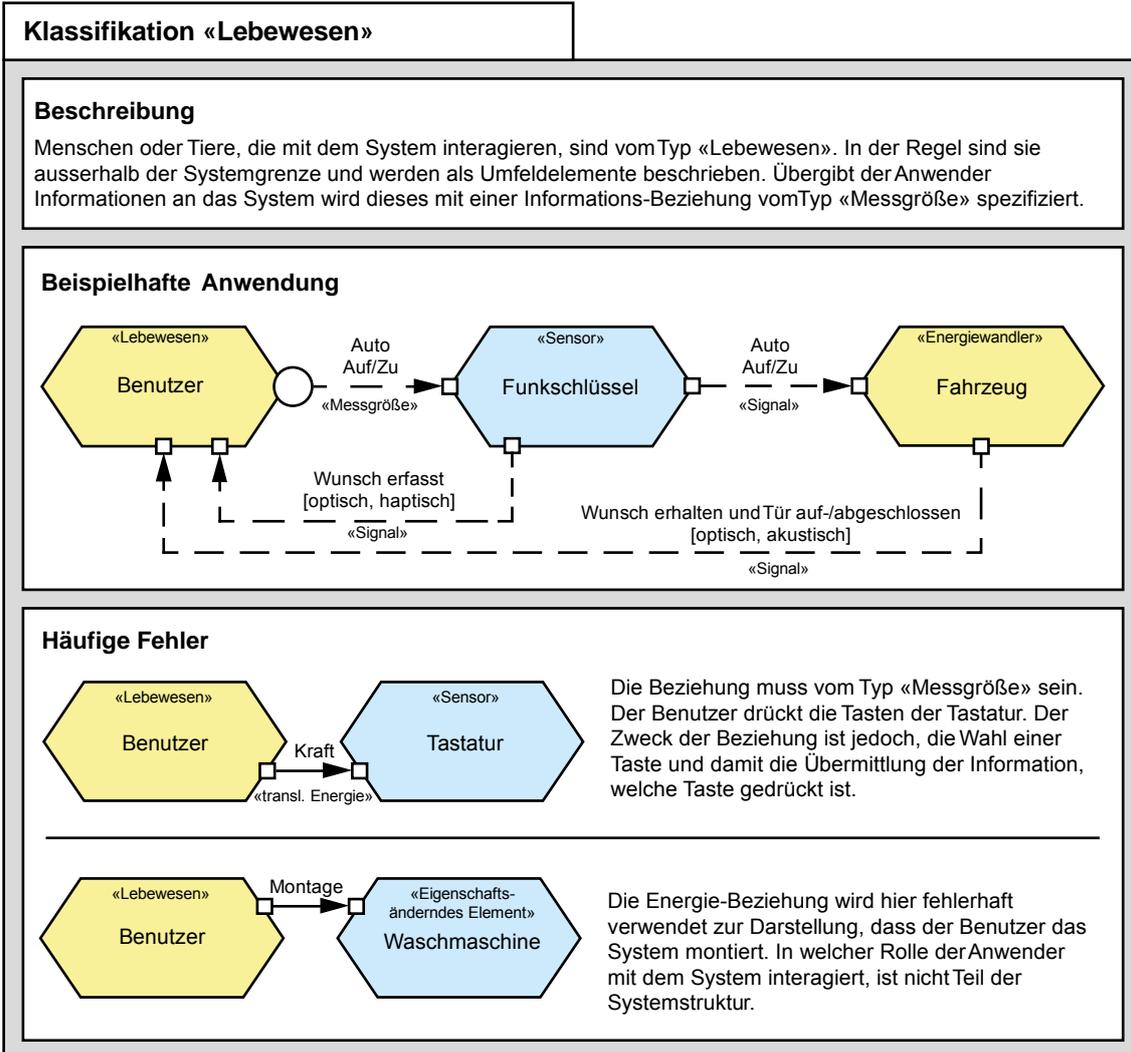


Bild A-8: Steckbrief Lebewesen

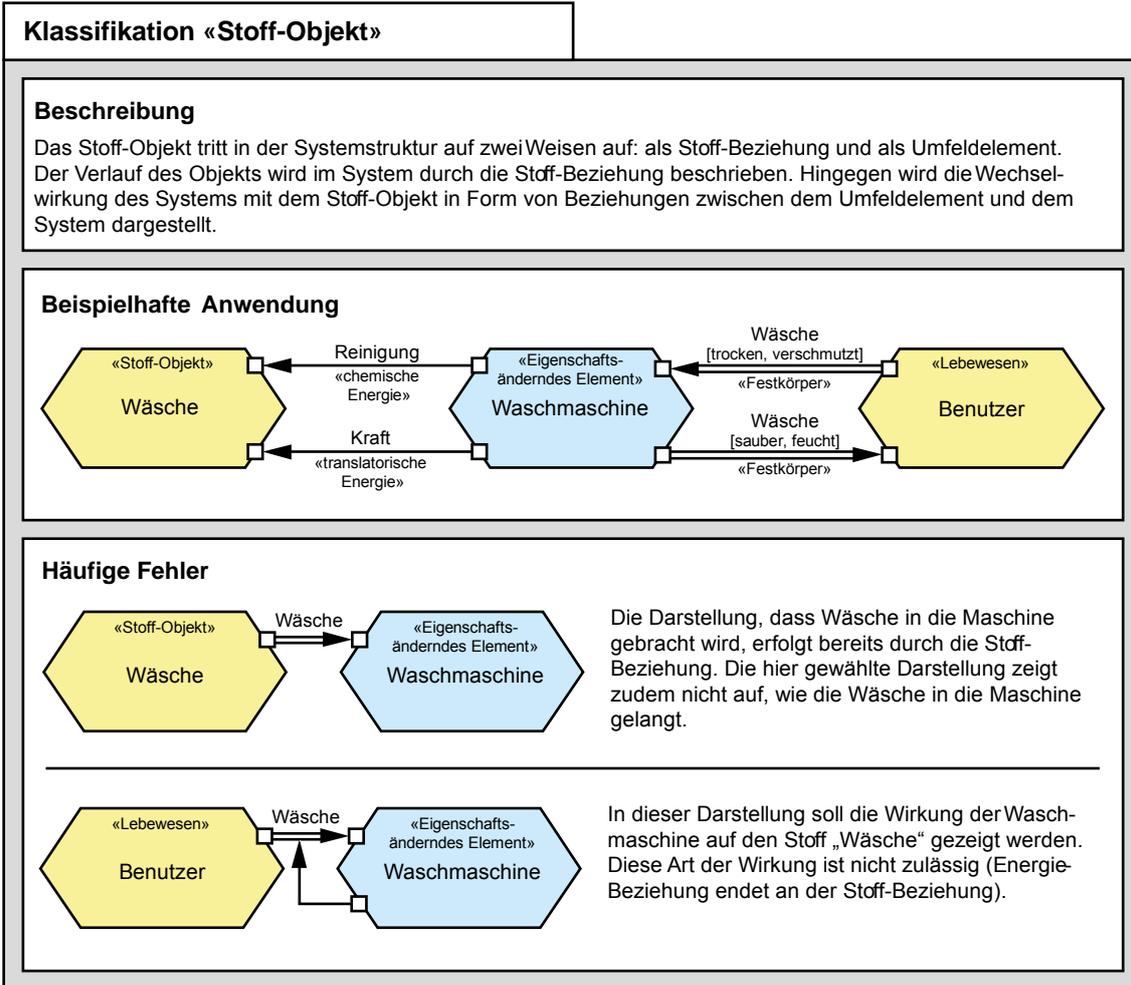


Bild A-9: Steckbrief Stoff-Objekt

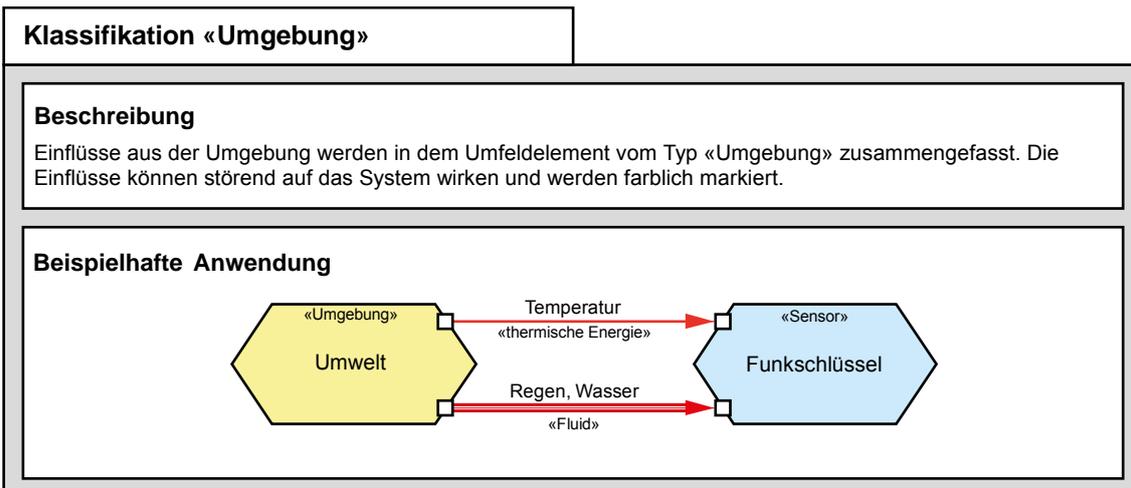


Bild A-10: Steckbrief Umgebung

A2 Liste der Richtlinien und Bedingungen

Vergleichbarkeit

- Bedingung:** Jedes Element und jede Beziehung ist einer Klasse zuzuordnen.
- Richtlinie:** Die Klasse (Energie, Information, Stoff, mechanische Verbindung) einer Beziehung wird durch den Zweck der Verbindung bestimmt.
- Richtlinie:** Die Elementklasse der technischen Elemente wird auf Basis der Hauptaufgabe des Elements festgelegt (Energie umsetzen, Information umsetzen, Stoff umsetzen oder Element halten/schützen).
- Richtlinie:** Menschen oder Tiere, die mit dem System interagieren, werden durch Elemente repräsentiert mit der Elementklasse «Lebewesen».
- Richtlinie:** Stoffe, die durch das System umgesetzt werden, werden als Elemente vom Typ «Stoff-Objekt» repräsentiert.
- Richtlinie:** Einflüsse aus dem Systemumfeld (z.B. Umwelteinflüsse, Raumbedingungen) werden als Beziehung dargestellt. Diese starten am Element vom Typ «Umgebung».
- Richtlinie:** Dekomposition von Elementen erfolgt nach funktionalen Gesichtspunkten.
- Richtlinie:** Ist ein Element in weitere Elemente dekomponiert, darf das übergeordnete Element keinem Bauteil entsprechen.

Vollständigkeit

- Bedingung:** Elemente und Beziehungen müssen benannt werden.
- Richtlinie:** Ein projektweites Glossar enthält mindestens die Abkürzungen und die entsprechenden ausgeschriebenen Elementbezeichnungen.
- Bedingung:** Energieumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Energie».
- Bedingung:** Informationsumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Information».
- Bedingung:** Stoffumsetzende Elemente haben eine ein- und ausgehende Beziehung vom Typ «Stoff».
- Bedingung:** Tragstrukturen haben mindestens eine Beziehung vom Typ «mechanische Verbindung».

- Bedingung:** Alle Elemente und Beziehungen sind einer Klasse der höchsten Detailstufe zuzuordnen.
- Bedingung:** Es besteht ein 1:m-Verhältnis, mit $m > 1$, zwischen dem Element der Ebene n und den Elementen der Ebene $n+1$.
- Bedingung:** Mindestens ein Element der Ebene $n+1$ erbt die Elementklasse des Vater-Elements. Die Ausnahme bildet die Elementklasse «Eigenschaftsänderndes Element».
- Bedingung:** Gewollte Wirkbeziehungen vom Typ «Energie», «Information» oder «mechanische Verbindung» müssen bis zur untersten Hierarchieebene geführt werden.
- Richtlinie:** Ungewollte Wirkbeziehungen (störende Einflüsse) können auf beliebiger Hierarchieebene enden mit der Semantik, dass die Wirkung sich auf alle innenliegenden Elemente bezieht.
- Richtlinie:** Gewollte Wirkbeziehungen vom Typ «Stoff» können am Stoffumsetzenden Element der Ebene n enden, wenn keine Stoffumsetzenden Elemente auf der Ebene $n+1$ enthalten sind.
- Richtlinie:** Die Energiequellen des Systems sind mit dem Attribut „Energiequelle“ zu versehen.
- Bedingung:** In der Systemstruktur ist mindestens eine Energiequelle vorhanden.
- Richtlinie:** Elemente, die elektrische Energie benötigen, sind mit dem Attribut „aktiv“ zu versehen.
- Bedingung:** Elemente mit dem Attribut „aktiv“ besitzen mindestens eine eingehende Beziehung vom Typ «elektrische Energie».
- Bedingung:** Wird der Zustand eines Stoffes durch ein Eigenschaftsänderndes Element geändert, besitzt das Element mindestens eine eingehende Beziehung vom Typ «Energie».
- Richtlinie:** Elemente, die einem physischen Element entsprechen, erhalten das Attribut „gestaltbehaftet“.
- Bedingung:** Gestaltbehaftete Elemente haben mindestens eine Beziehung vom Typ «mechanische Verbindung».
- Bedingung:** In der Systemstruktur ist mindestens eine Beziehung vom System zu einem Umfелеlement vom Typ «mechanische Verbindung» vorhanden.

Richtigkeit

- Bedingung:** Für Elemente der Klasse «Energieübertrager» gilt: Energie-Typ IN = Energie-Typ OUT.
- Bedingung:** Für Elemente der Klasse «Energiewandler» gilt: Energie-Typ IN \neq Energie-Typ OUT.
- Bedingung:** Für Elemente der Klasse «Sensor» gilt: «Messgröße» IN und «Signal» OUT.
- Bedingung:** Für Elemente der Klasse «Informationsverarbeitung» gilt: «Signal» IN und «Signal» OUT.
- Bedingung:** Für Elemente der Klasse «Eigenschaftsänderndes Element» gilt: Stoff IN = Stoff OUT.
- Bedingung:** Für Elemente der Klasse «Stoffänderndes Element» gilt: Stoff IN \neq Stoff OUT.
- Bedingung:** Für Elemente der Klasse «Lebewesen» gilt: ausgehende Informationsbeziehung ist vom Typ «Messgröße».
- Bedingung:** Der Objektfluss durch das System wird durch die Stoffbeziehung spezifiziert. Hierzu wird ein Element vom Typ «Stoff-Objekt» angelegt und mit Merkmalen und Merkmalsausprägungen beschrieben.
- Bedingung:** Ein Element vom Typ «Stoff-Objekt» wird als Umfeldelement visualisiert, wenn Wechselwirkungen des Systems mit dem Objekt spezifiziert werden.
- Bedingung:** Die Stoffbeziehung, die den Objektfluss durch das System beschreibt, darf nicht am entsprechenden Element «Stoff-Objekt» beginnen.

A3 Matrix der energieumsetzenden Elemente

Energie										
	Translatorisch	Rotatorisch	Elektrisch	Pneumatisch	Hydraulisch	Magnetisch	Optisch	Akustisch	Chemisch	Thermisch
Translatorisch	Hebel, Feder, Dämpfer, Hydraulik-Pressse, Keil	Zahnradstange	Piezo	Kolbenpumpe, Luftpumpe	Kolbenpumpe			Trommel		
Rotatorisch	Gewindespindel, Nockengetriebe, Zahnradstange	Gebtriebe, Welle, Rad, Zahnrad	Generator	Verdichter						Widerstandsbremse
Elektrisch	Hubaktor, Linearmotor, Piezo, Zylinder	Rotationsmotor, DC-Motor, AC-Motor	Akkumulator, Leistungselektronik, Netzteil	Pneumatikpumpe	Hydraulikpumpe	Stromdurchfl. Leiter, Spule, Elektromagnet	LED, Lampe, OLED, Laser	Lautsprecher	Akkumulatorenzellen	Heizstab
Pneumatisch	Zylinder	Turbine, Windrad	Pneumatikgenerator	Ventil, Druckminderer						
Hydraulisch	Zylinder	Turbine, Schaufelrad, Hydraulikmotor	Hydraulikgenerator		Ventil, Druckminderer					
Magnetisch						Dauermagnet				
Optisch			Solarzelle, Photovoltaik				Spiegel, Linse, optische Gitter			Solaranlage
Akustisch								Resonanzkörper		Thermoakustische Maschine
Chemisch		Verbrennungsmotor	Akkumulatorenzellen							Ölheizung, Gasheizung
Thermisch	Wärmeleistungsmaschine	Wärmeleistungsmaschine					Glühbirne	Thermoakustische Maschine		Wärmetauscher, Wärmeleiter, Wärmepumpe

Bild A-11: Mit Beispielen gefüllte Matrix der energieumsetzenden Elemente (Aufbau nach [Czi08])

A4 Matrix der Sensoren

		Wirkprinzip					
		resistiv R	induktiv L	kapazitiv C	Spannung U	Strom I	Ladung Q
Messgröße	Position: Länge l Weg s Winkel ϕ	Potentiometer Magnetoresistiver Sensor, Gauß-Feldplatte	Differenzial- Transformator, Tauchanker- Wegsensor	Kapazitiver Wegsensor	Hall-Sensor, Op- toelektronischer Lichtschranken- Sensor	Wirbelstrom- Sensor	
	Dehnung $\epsilon = \Delta l/l_0$	Dehnungsmess- streifen				Faseroptische Sensoren	
	Geschwindig- keit $v=ds/dt$	Magnetoresisti- ver Drehwinkel- Sensor	Induktiver Dreh- winkel-Sensor		Magnetpol-Dreh- zahl-Sensor	Optoelektroni- scher Drehzahl- Sensor	Gyrometer/ Piezo-Sensor
	Beschleuni- gung $a=dv/dt$	Seismische Masse-Dämpfer-Feder-Systeme, Rückführung auf Wegmessung: resistiv, induktiv, kapazitiv, optoelektronisch, piezoresistiv; Hall/Gauß-Sensorik oder Dehnungsmessung (DMS)					
	Kraft F Moment F · l	Piezoresistiver Sensor, Dehnstoffe	Magnetoelasti- scher Sensor	Kraftkompensa- tions-Sensor	Federelemente oder DMS, Rückführung auf s-, ϵ -Messung		Piezoelektri- scher Sensor
	Druck $p = \text{Kraft}/\text{Flä-che}$	Piezoresistiver Dehnstoff- Sensor	Magnetoelasti- scher Sensor	Kapazitiver Drucksensor	Federelemente oder DMS, Rückführung auf s-, ϵ -Messung		Piezoelektri- scher Sensor
	Temperatur T	NTC/PTC- Widerstände			Thermo- elemente	Optoelektroni- sches Pyrometer	
	Feuchte f_{abs}, f_{rel}	Resistives Hyg- rometer		Kondensator Hygrometer			

Bild A-12: Sensoren-Matrix [Czi08]

A5 Aufgaben und Fähigkeiten eines Systems Engineers

Mit der Einführung der Systemmodellierung sind Aufgaben verbunden, die in den Rol-
lendefinitionen des Systems Engineering nicht berücksichtigt werden. Hierzu zählen
alle Aufgaben rund um die Modellierung, Dokumentation und Pflege des Systemmo-
dells. Dabei organisiert und moderiert der Systems Engineer Workshops mit den Fach-
disziplinen. Die Ergebnisse aus den Workshops werden von ihm aufbereitet und Un-
klarheiten oder Änderungen mit den jeweiligen Experten abgestimmt. Er beruft Sitzun-
gen ein, wenn disziplinübergreifende Änderungen entstanden sind und am Systemmo-
dell diskutiert werden müssen. Er ist zuständig, die Grundsätze ordnungsmäßiger Mo-
dellierung einzuhalten. Bild A-13 zeigt die Aufgaben und das Fähigkeitsprofil eines
Systems Engineers.

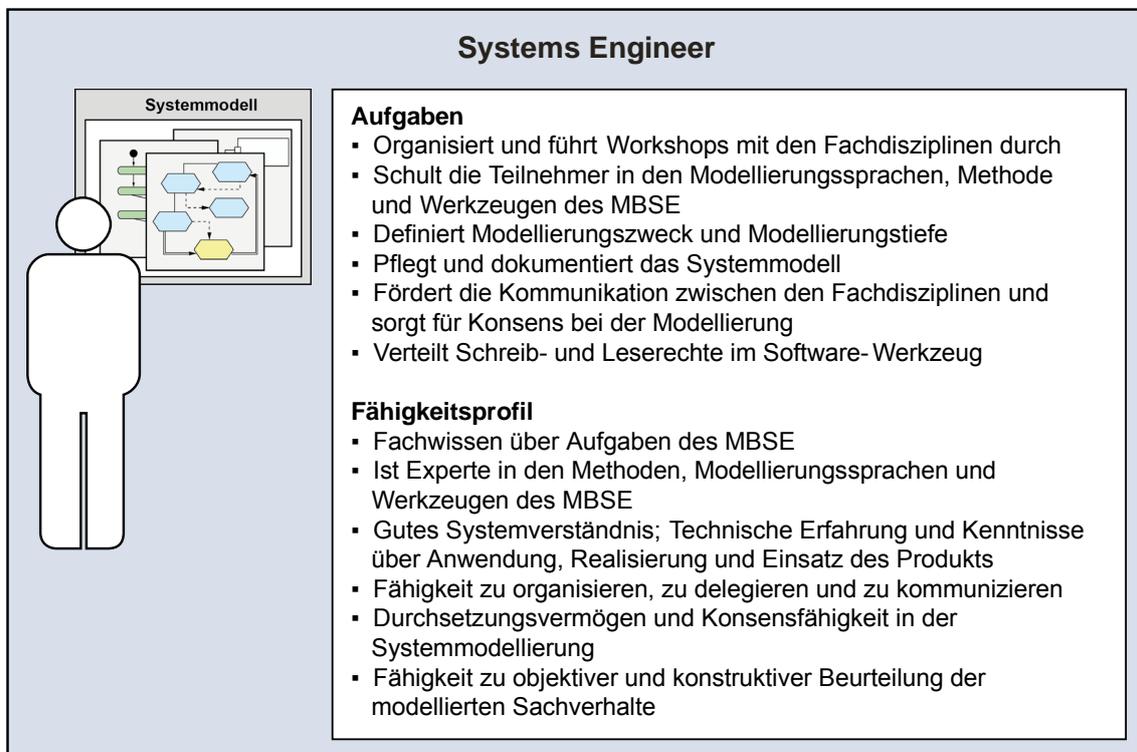


Bild A-13: Aufgaben und Fähigkeitsprofil eines Systems Engineers im Kontext MBSE

A6 Systemstruktur der Sortieranlage

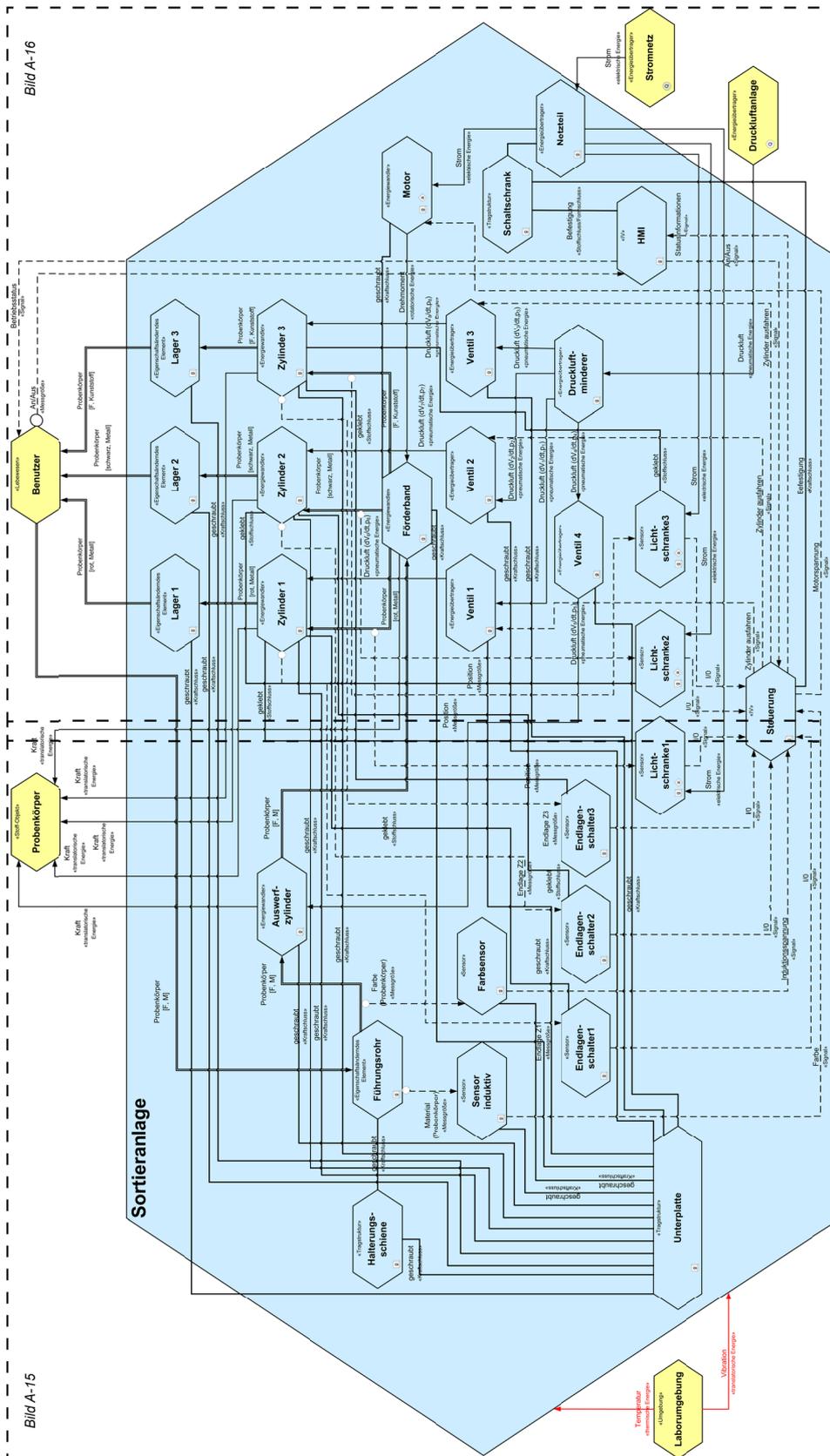


Bild A-14: Systemstruktur der Sortieranlage in Übersicht

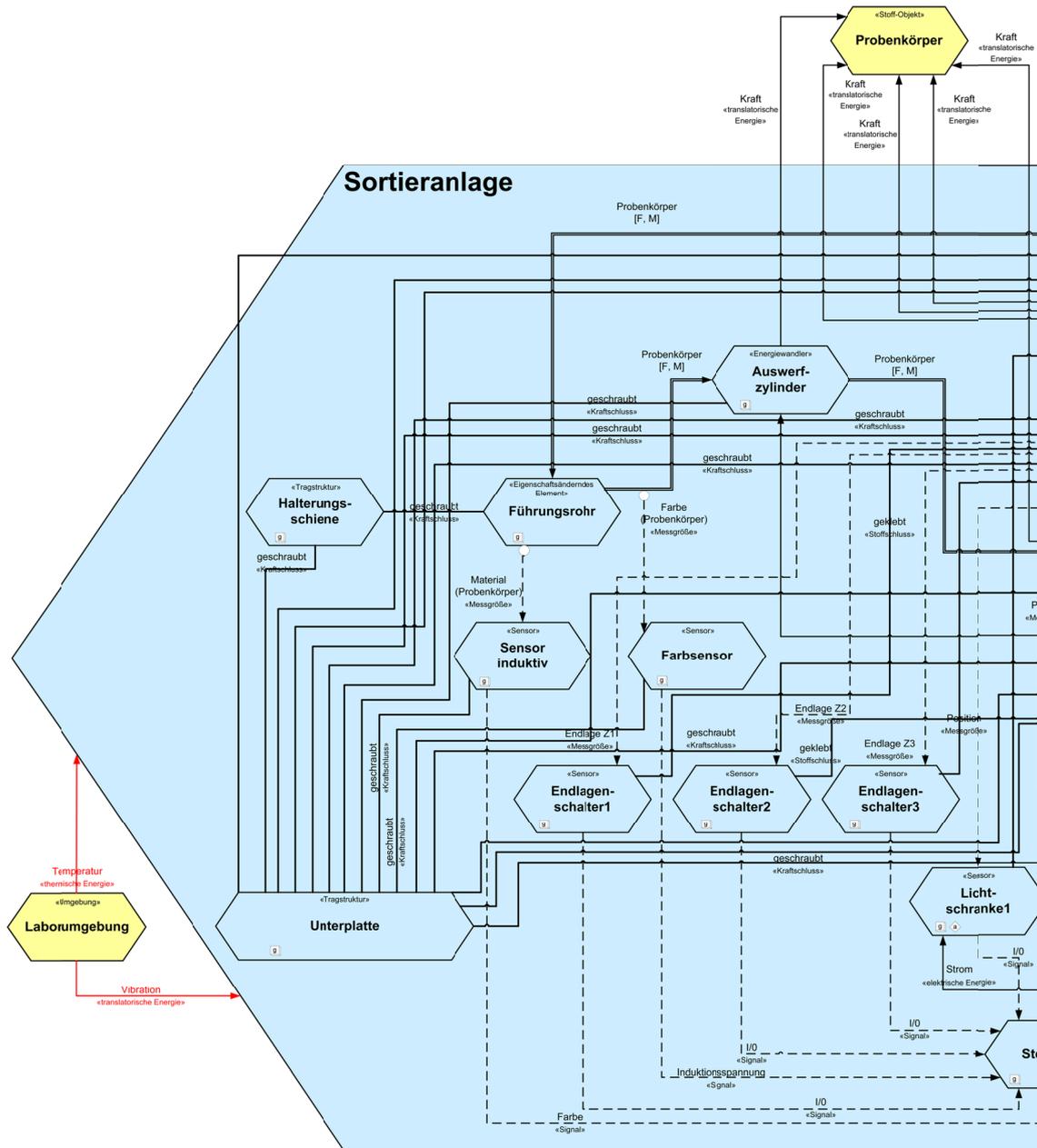


Bild A-15: Systemstruktur der Sortieranlage – linker Ausschnitt

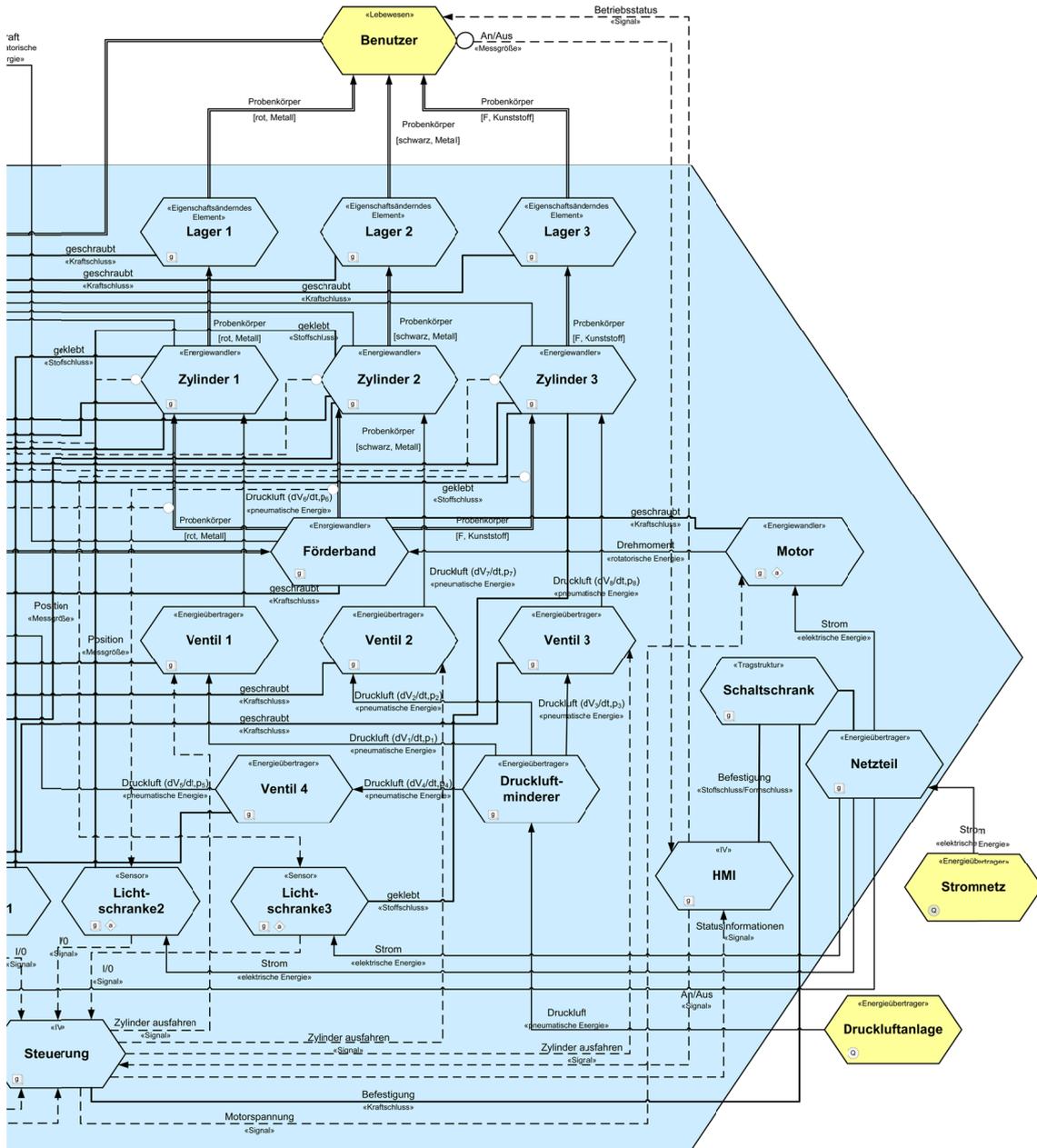


Bild A-16: Systemstruktur der Sortieranlage – rechter Ausschnitt