

Florian Munker

**Ein Ansatz zur anwenderorientierten
Systemmodellierung für die
interdisziplinäre Produktentwicklung**

A User-Oriented Concept of Systems Modeling
for Interdisciplinary Product Engineering

Band 98

Systeme ■ Methoden ■ Prozesse

Hrsg.: Prof. Dr.-Ing. Dr. h.c. A. Albers



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung –
Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz
(CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>

Forschungsberichte



Florian Munker

**Ein Ansatz zur anwenderorientierten
Systemmodellierung für die interdisziplinäre
Produktentwicklung**

A User-Oriented Concept of Systems Modeling for
Interdisciplinary Product Engineering

Band 98

Herausgeber: Prof. Dr.-Ing. Dr. h.c. A. Albers

Copyright: IPEK ▪ Institut für Produktentwicklung, 2016
Karlsruher Institut für Technologie (KIT)
Die Forschungsuniversität in der Helmholtz-Gemeinschaft

Alle Rechte vorbehalten

Druck: Stolzenberger Druck und Werbung GmbH & Co. KG, Leimen
06224-7697915

ISSN 1615-8113

Ein Ansatz zur anwenderorientierten Systemmodellierung für die interdisziplinäre Produktentwicklung

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation

von

Dipl.-Ing. Florian Munker
aus Lauf a. d. Pegnitz

Tag der mündlichen Prüfung: 3. November 2016
Hauptreferent: Prof. Dr.-Ing. Dr. h.c. A. Albers
Korreferent: Prof. Dr.-Ing. M. Geimer

Vorwort des Herausgebers

Wissen ist einer der entscheidenden Faktoren in den Volkswirtschaften unserer Zeit. Der Unternehmenserfolg wird in der Zukunft mehr denn je davon abhängen, wie schnell ein Unternehmen neues Wissen aufnehmen, zugänglich machen und verwerten kann. Die Aufgabe eines Universitätsinstitutes ist es, hier einen wesentlichen Beitrag zu leisten. In den Forschungsarbeiten wird ständig Wissen generiert. Dieses kann aber nur wirksam und für die Gemeinschaft nutzbar werden, wenn es in geeigneter Form kommuniziert wird. Diese Schriftenreihe dient als eine Plattform zum Transfer und macht damit das Wissenspotenzial aus aktuellen Forschungsarbeiten am IPEK - Institut für Produktentwicklung Karlsruhe¹ am Karlsruher Institut für Technologie (KIT) verfügbar.

Die Forschungsfelder des Institutes sind die methodische Entwicklung und das Entwicklungsmanagement, die rechnergestützte Optimierung von Strukturen und Systemen, die Antriebstechnik mit einem Schwerpunkt auf den Gebieten Antriebsstrang-Engineering und Tribologie und Monitoring von Lager- und Funktionsreibsystemen, die NVH mit dem Fokus auf Schwingungen und Akustik an Komponenten und am Gesamtfahrzeug, die Mikrosystemtechnik mit dem Fokus auf die zugehörigen Entwicklungsprozesse sowie die Mechatronik. Die Forschungsberichte werden aus allen diesen Gebieten Beiträge zur wissenschaftlichen Fortentwicklung des Wissens und der zugehörigen Anwendung – sowohl den auf diesen Gebieten tätigen Forschern als auch ganz besonders der anwendenden Industrie – zur Verfügung stellen. Ziel ist es, qualifizierte Beiträge zum Produktentwicklungsprozess zu leisten.

Albert Albers

¹ Eh.: Institut für Maschinenkonstruktionslehre und Kraftfahrzeugbau, Universität Karlsruhe (TH)

Vorwort zu Band 98

In den letzten 20 Jahren sind die Produkte im Maschinen- und Fahrzeugbau zunehmend komplexer und komplizierter geworden. Hier sind die Integration von elektronischen Teilsystemen sowie informatorischen Teilsystemen - durch entsprechende Embedded Softwarelösungen - ein wesentlicher Treiber dieses Prozesses. Ziel ist und bleibt es natürlich, mit jeder Produktgeneration den Kundennutzen zu erhöhen, bei gleichzeitiger Sicherung und Realisierung des Anbieternutzens. Dies in globalisierten, hoch kompetitiven Käufermärkten zu realisieren, stellt eine besondere Herausforderung für die moderne Produktentwicklung dar. Um hier zu einer höheren Effizienz und Effektivität zu kommen, ist es zwingend notwendig, von der komponentenorientierten Entwicklungsstrategie hin zu einer ganzheitlichen Entwicklungsstrategie zu kommen. Für diesen systembasierten Ansatz wurden in den letzten Jahren einige relevante Forschungen auf dem Gebiet der Systemtheorie, deren Umsetzung in das sogenannte Systems-Engineering und die Unterstützung dieses Systems-Engineering-Prozesses durch Software-Werkzeuge – unter anderem auch in meiner Forschungsgruppe zu Entwicklungs- und Managementmethoden in der Produktentwicklung geleistet, um so die Herausforderung der Komplexität mit neuen Ansätzen zu berücksichtigen. Diese Forschungsarbeiten, die ganz wesentlich auf Ansätzen des Systems-Engineerings aus der Entwicklung der Raumfahrt in den USA in den 60er- und 70er-Jahren basieren, und die unter anderem durch die grundlegenden Arbeiten von ROPOHL zur Systemtheorie befruchtet wurden, führen heute zu neuen Möglichkeiten einer Verknüpfung von Wissen aus den verschiedenen Domänen in einer neuen Qualität. Auch die Hersteller von CAD- und CAE-Systemen nehmen seit Kurzem diese Konzepte auf und versuchen sie, in ihre Software-Lösungen zu integrieren. Eine sehr große Herausforderung ist dabei allerdings immer noch die Akzeptanz beim Entwickler im Unternehmen. Der Anwender wird oft durch zu generische Konzepte überfordert, die dann auch noch sehr wenig intuitiv und effektiv in Software-Tools umgesetzt wurden.

An dieser Stelle setzt die Arbeit von Herrn Dr.-Ing. Florian Munker an. Er hat sich zum Ziel gesetzt, die Gründe für die Anwenderskepsis herauszuarbeiten und auf der Basis dieser Analyse Vorschläge zu einer anwenderorientierten Systemmodellierung für die interdisziplinäre Produktentwicklung zu erstellen. Die Motivation und Zielsetzung der wissenschaftlichen Arbeit von Herrn Dr.-Ing. Florian Munker ergibt sich aus den oben genannten Herausforderungen der interdisziplinären Produktentwicklung und den langjährigen Forschungsarbeiten in der KaSPro – Karlsruher Schule für Produktentwicklung, in die sich die Arbeit hervorragend

integriert. Die sehr gelungene Analyse heutiger realisierter Lösungen und die bis zu einem Funktionsprototypen für die Werkzeugunterstützung erarbeiteten Lösungskonzepte für eine erfolgreiche Nutzung von MBSE liefern wertvolle Beiträge für die Forschung und die Praxis.

Oktober, 2016

Albert Albers

Kurzfassung

Die Welt verändert sich, auch für den Menschen in der Produktentwicklung. War früher die Kompetenz eines Entwicklers durch disziplinspezifisches Fachwissen geprägt, ist heute und in Zukunft eine interdisziplinäre Ausrichtung des Kompetenzprofils erforderlich. Die Karlsruher Schule der Produktentwicklung begreift diesen Menschen als denkendes und handelndes Zentrum in der Produktentstehung.

Die dieser Arbeit zugrunde liegende Motivation besteht darin, den Entwickler durch interdisziplinäre Systemmodellierung in seinem Denken und Handeln bei der Lösung seiner interdisziplinären Entwicklungsaufgaben zu unterstützen, indem durch diese Modellierung die technischen Aspekte eines Systems kompetenzübergreifend modellbasiert abgebildet und zugänglich gemacht werden.

Die Idee der interdisziplinären Systemmodellierung ist nicht neu und findet beispielsweise mit der Systems Modeling Language (SysML) Anwendung. Während deren Potenzial anerkannt und geschätzt wird, findet ein in der Breite praxistauglicher Einsatz jedoch noch nicht statt. Das Ziel dieser Arbeit ist, einen neuen Ansatz zur interdisziplinären Systemmodellierung zu konzipieren. Die Grundlage dieses Entwurfs bilden die Erkenntnisse aus dem menschenzentrierten Entwicklungsmodell. Vor diesem Hintergrund soll der Ansatz dem Anwender einen niederschweligen Einstieg in die interdisziplinäre Systemmodellierung ermöglichen, ohne Agilität und Flexibilität einzuschränken. Der Anwender soll sich statt auf Formalismen einer Modellierung, auf die Lösung seiner Entwicklungsaufgabe konzentrieren können.

Dazu werden zunächst bestehende Ansätze untersucht. Unter Berücksichtigung von Anwendungsstudien zur interdisziplinären Systemmodellierung wird daraus ein initiales Zielsystem erstellt. Auf Grundlage dieses Zielsystems wird ein Validierungsprototyp namens „SystemSketcher“ entwickelt. Dieser stellt eine allererste Werkzeugrealisierung für die Erprobung der grundsätzlichen Tragfähigkeit des Ansatzes dar und wird anhand eines mechatronischen Beispiels erläutert. Die Erprobung des Ansatzes findet durch ein Panel aus Teilnehmern der studentischen Lehrveranstaltung „IP – Integrierte Produktentwicklung“ statt, die ein Batteriesystem für Elektrofahrzeuge im Sinne der Produktgenerationsentwicklung mit SystemSketcher zu einem Energiespeichersystem für landwirtschaftliche Anwesen weiterentwickeln.

Mit der nachgewiesenen grundsätzlichen Tragfähigkeit des Ansatzes und einer ersten Werkzeugrealisierung ist eine Grundlage für weitere wissenschaftliche Untersuchungen zur interdisziplinären Produktentwicklung entstanden. Die interdisziplinäre Systemmodellierung könnte sich in diesem Bereich weiter als Schlüsseltechnologie etablieren.

abstract

The world is changing, and so is product engineering. Formerly shaped as discipline-specific, the competence profile of engineers changed to an interdisciplinary characteristic. The Karlsruhe approach of product engineering takes this into account by considering the engineer, his individual thinking and his personal acting as the central and vital part of product engineering.

The motivation of this thesis is to support the engineer in interdisciplinary product engineering with a new approach of interdisciplinary systems modeling. The basic idea of creating a systems model, which contains and provides all relevant information in an interdisciplinary comprehensible way, is not new. It has been implemented by several modeling approaches, the most prominent is modeling with Systems Modeling Language (SysML). While the potential of SysML has been broadly recognized, there is a lack of adoption in practical application. Therefore, the objective of the research presented in this thesis is to devise a new approach of interdisciplinary systems modeling. This approach will be based on insights from a human-centered explanation model. It should provide a low barrier to entry interdisciplinary systems modeling without compromising on agility or flexibility. In essence, the user should be able to focus on solving his development task instead of focusing on formalism of modeling.

Initially, existing modeling approaches are examined. Taking insights from application studies into account, an initial draft is generated which then is implemented in „SystemSketcher“. SystemSketcher is a first, initial tooling which is used to evaluate the basic practicability on the example of a mechatronic system.

Participants of „IP – Integrated Product Development“, a design lecture for students which provides high-level engineering education by simulating a real-world innovation and product development experience, are used as probands. Their original assignment is to seek innovation in the field of battery systems for electric vehicles, which is refined for stationary application in this case study. They execute conceptual design of a stationary energy storage system using SystemSketcher.

The proven viability of the approach as well as the prototypical tooling serve as a launch pad for further research and application in interdisciplinary product engineering. Therein, interdisciplinary systems modeling could potentially prove as a key technology.

Danksagung

Diese Arbeit entstand am IPEK – Institut für Produktentwicklung am Karlsruher Institut für Technologie (KIT) im Rahmen meiner Tätigkeit als Sachbearbeiter und Projektleiter in Innovationsprojekten der Elektromobilität.

Mein besonderer Dank gilt dabei meinem Doktorvater, Prof. Dr.-Ing. Dr. h. c. Albert Albers, der es mir ermöglicht hat, diese Tätigkeit sowohl durch theoretische Betrachtung als auch durch praktische Anwendung auszuüben. Er hat mir dabei umfassende Kenntnisse über die methodische Produktentwicklung und –Validierung zur zielstrebigem Lösung von Entwicklungsaufgaben nach der Karlsruher Schule vermittelt, sowie mich in meiner Herangehensweise an neue Herausforderungen nachhaltig persönlich geprägt.

Herrn Prof. Dr.-Ing. Marcus Geimer möchte ich für die Übernahme des Korreferates danken, sowie für den fachlichen Austausch und die Anregungen zu meinem Forschungsthema.

Weiterhin möchte ich mich bei Dr.-Ing. Matthias Behrendt bedanken, der mich in seiner Funktion als Oberingenieur in meiner Forschungsarbeit begleitet hat. Ich möchte ihm insbesondere für die fachlichen Anregungen zur systemischen Betrachtung von Elektromobilitätssystemen, die Unterstützung meiner experimentellen Untersuchungen derer auf dem Gesamtfahrzeugprüfstand, sowie für die Anregungen zur Untersuchung der Systemmodellierung danken.

Karlsruhe, im Herbst 2016

Florian Munker

„Simple can be harder than complex; you have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.“

Steve Jobs, 1998

Inhalt

1	Einleitung	1
2	Grundlagen und Stand der Forschung	6
2.1	Die Karlsruher Schule der Produktentwicklung – KaSPro	6
2.2	Begriffsklärung	7
2.3	Erweitertes ZHO-Modell	9
2.4	Die drei Säulen interdisziplinärer Systemmodellierung	11
2.5	Relevante Ansätze zur interdisziplinären Systemmodellierung	12
2.5.1	OMG SysML	13
2.5.2	IBM Telelogic Harmony-SE	14
2.5.3	IBM RUP SE	14
2.5.4	VITEC MBSE	15
2.5.5	JPL SA	15
2.5.6	Dori OPM	16
2.5.7	CONSENS	16
2.5.8	RFLP	16
2.6	Systemmodellierung in der Produktgenerationsentwicklung	17
2.7	Gegenwärtige Anwendersicht zur Modellierung mit SysML	21
2.8	Anwendungsstudien interdisziplinärer Systemmodellierung	23
2.9	Zwischenfazit	24
3	Motivation und Zielsetzung	26
3.1	Motivation	26
3.2	Zielsetzung	28
3.3	Forschungsfragen	30
4	Forschungsdesign	31
4.1	Forschungsmethode	31
4.2	Vorgehen zur Entwicklung des Ansatzes	33
4.3	Vorgehen zur Validierung des Ansatzes	34
5	Analyse der relevanten Ansätze	35
5.1	Analyse der Sprachen	35
5.2	Redundanz und Konsistenz in den Sprachräumen	40
5.3	Grafischer Vergleich der Sprachräume	42
5.4	Analyse der Methoden	46
5.1	Analyse der Werkzeuge	46
5.2	Zwischenfazit	50
6	Randbedingungen zur interdisziplinären System-modellierung	51
6.1	Formalisierung	51
6.2	Unmittelbar erforderliche Randbedingungen	52
6.2.1	Anwenderorientierte Objektorientierung	52
6.2.2	Anwenderorientierte grafische Modellierung	54

6.2.3	Anwenderorientierte Sichtenbildung	54
6.3	Ergänzende Randbedingungen	55
6.3.1	Verhaltensmodellierung mit Zuständen	56
6.3.2	Anwenderspezifische Erweiterung der Modellierung.....	56
6.3.3	Umgang mit der grafischen Modellierung	56
6.3.4	Darstellungsformen von Traceability	57
6.3.5	Auswertemechanismen am Modell.....	57
6.3.6	Vererbung und Weiterverwendung	57
7	Initiales Zielsystem des Ansatzes	58
7.1	Initiales Zielsystem der Sprache	58
7.1	Initiales Zielsystem der Methode	61
7.1	Initiales Zielsystem des Werkzeugs	63
7.2	Zwischenfazit initiales Zielsystem	65
8	Validierungsprototyp – SystemSketcher	67
8.1	Implementierung des Validierungsprototyps.....	67
8.2	Realisierung und Einschränkungen im Validierungsprototyp	70
8.3	Gesamtüberblick über Grafisches User Interface	72
8.4	Partialmodelle	72
8.5	Strukturbäume.....	74
8.6	Sichtenbildung.....	76
8.7	Komponenten Picker.....	78
8.8	Metadaten Manager	80
8.9	Grafische Modellierung	81
8.10	Relationen Manager.....	83
9	Validierung des Ansatzes	85
9.1	Studiendesign	86
9.2	Durchführung der Studie	90
9.3	Studienergebnisse	94
9.4	Zwischenfazit	98
10	Zusammenfassung und Ausblick.....	99
10.1	Zusammenfassung.....	99
10.2	Ausblick	100
11	Literaturverzeichnis.....	102
12	Anhang: Analyse der Sprachräume	107
13	Anhang: Grafischer Plot der Sprachräume.....	129
14	Anhang: Materialien zur Validierungsstudie.....	134

1 Einleitung

Unsere Welt verändert sich für den Menschen. Auch in der Produktentwicklung. INCOSE, der „International Council on Systems Engineering“, arbeitet dies in seiner Studie „A World In Motion – INCOSE Systems Engineering Vision 2025“² heraus. Als gemeinnützige Organisation zur Verbreitung und Weiterentwicklung des interdisziplinären, auf der Systemtheorie basierenden Forschungs- und Entwicklungsansatzes „Systems Engineering“, fokussiert sie sich darin auf Trends im Kontext der interdisziplinären Produktentwicklung. INCOSE sieht das gegenwärtig vornehmlich in der Luft- und Raumfahrt bewährte Systems Engineering als Antwort auf die steigende Komplexität und zunehmende Vernetzung von technischen Systemen in der Produktentwicklung.

ALBERS und GAUSEMEIER beschreiben in einem Kapitel der Studie „Smart Engineering“³ der deutschen Akademie der Technikwissenschaften (acatech) den Trend zur systemorientierten Produktentstehung. Sie erkennen, wie die INCOSE, den Trend zu intelligenten und vernetzten mechatronischen Systemen. Auch sie sehen Systems Engineering als Kern einer integrierten Produktentwicklung. Diese integrierte Produktentwicklung soll eine „gleichberechtigte, systemorientierte Integration von Methoden und Arbeitstechniken aller Fachdisziplinen“³ darstellen.

Das Karlsruher Lehrmodell für Produktentwicklung – KaLeP ist ein durchgängiges Ausbildungssystem in der universitären Lehre am IPEK, das den Anspruch hat, den Ingenieur von morgen gezielt auf diese Herausforderungen vorzubereiten. ALBERS, BURKARDT und MATTHIESEN propagieren darin den ausgebildeten Ingenieur als Generalisten mit starker Problemlösungskompetenz und zielgerichteter Arbeitsweise als einen heute und in Zukunft dringend benötigten „Problemlösungs-Manager“.⁴

Was dies für das Kompetenzprofil bedeutet, arbeiten ALBERS, DENKENA und MATTHIESEN in einer weiteren acatech-Studie „Faszination Konstruktion – Berufsbild und Tätigkeitsfeld im Wandel“⁵ am Beispiel des Konstrukteurs weiter heraus.

„Der Konstrukteur von morgen braucht neben klassischem Konstruktions-Know-how, wie Kenntnisse zu Maschinenelementen, Funktionsgruppen, Fertigungs- und Montagetechnik und räumlichem Vorstellungsvermögen zunehmend Kenntnisse in

² INCOSE SE VISION PROJECT TEAM 2014

³ ALBERS & GAUSEMEIER 2012

⁴ ALBERS ET AL. 2001

⁵ ALBERS ET AL. 2012a

Informatik, Simulationstechnik, Elektrotechnik und Mechatronik. Er muss aber auch im Projektmanagement firm sein sowie ganzheitliches Denken, Kreativität, Kommunikations- und Problemlösungsfähigkeit mitbringen.“⁵

Sie beschreiben damit den Konstrukteur als Menschen mit einem breiten Kompetenzprofil, dass in spezifischen Fachdomänen besonders tiefgründig ist. Um dies zu verdeutlichen, wird dieses Kompetenzprofil hier in Bild 1 als **Kammprofil** dargestellt. Ein horizontaler Balken stellt die übergreifende Kompetenz dar, während in der Vertikalen die tiefer gehenden Kompetenzen unterschiedlicher Fachdomänen visualisiert sind.

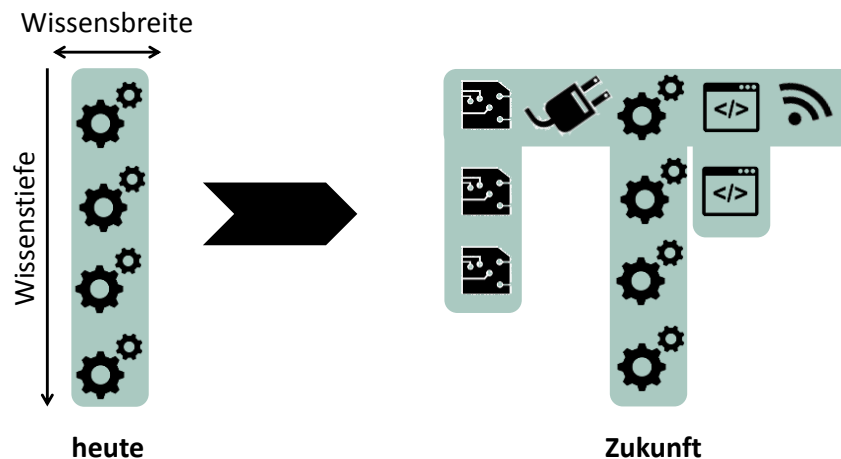


Bild 1: Kammförmiges Kompetenzprofil des „Konstrukteurs von morgen“⁶, propagiert im Karlsruher Lehrmodell für Produktentwicklung - KaLeP⁷ seit 2001

ALBERS ET AL. stellen ein menschenzentriertes Modell auf, das die mentalen Iterationen des denkenden und handelnden Menschen im Zentrum der Produktentstehung in Form einer „liegenden Acht“ beschreibt.⁸ Es basiert auf einem Systemtriple nach ROPOHL⁹, das von ALBERS aufgenommen und als ZHO-Modell in die Produktentstehung eingeführt wurde.¹⁰ Dieses ZHO-Modell beschreibt einen systemtechnischen Ansatz in der Produktentstehung, bei dem durch ein Handlungssystem kontinuierlich Elemente des Zielsystems (beispielsweise Anforderungen, Ziele, Randbedingungen) in das Objektsystem (beispielsweise virtuelle und physische Modelle, Zwischenergebnisse, Endprodukt) überführt werden.

⁶ ALBERS ET AL. 2012a

⁷ ALBERS ET AL. 2001

⁸ ALBERS ET AL. 2011

⁹ ROPOHL 1975

¹⁰ ALBERS 2010

In der „**liegenden Acht**“ wird der Mensch als zentrales Handlungssystem begriffen. Er verfügt dort über eine Wissensbasis und einen Lösungsraum. In einem Synthese- und Analyseprozess generiert er aus fallspezifischen Wissen über Explizierung im Zielsystem Lösungen in seinem mentalen Lösungsraum. In einem Synthese- und Analyseprozess generiert er aus dem Lösungsraum über Explizierung der Lösungen im Objektsystem neues Wissen in seiner mentalen Wissensbasis. Durch den kontinuierlichen Ablauf dieses Prozesses überführt der Entwickler eine anfangs vage Idee in Lösungen. Seine Wissensbasis und sein Lösungsraum werden sich dabei an der individuellen Ausprägung seines Kompetenzprofils orientieren.

ALBERS ET AL. greifen bei der ganzheitlichen Darlegung der Validierung in der Produktentwicklung¹¹ ein erweitertes Modell der „liegenden Acht“ auf, in welchem nach Bild 2 von ALBERS ET AL. die Aspekte der Kreation und der Validierung ergänzt wurden, um das Wechselspiel dieser als kontinuierlichen, geschlossenen Kreislauf zu beschreiben.¹²

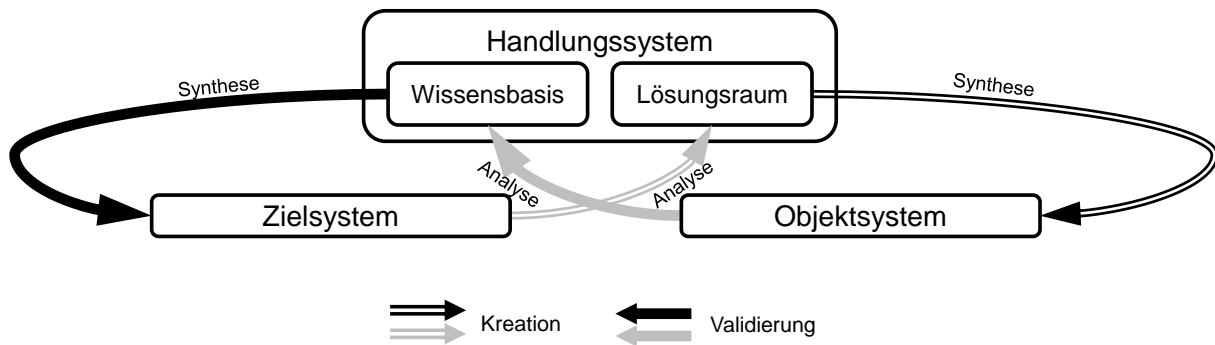


Bild 2: Modell der „liegenden Acht“, übersetzt nach ALBERS ET AL.¹²

Eine gemeinsame Sprache, die von Ingenieuren aller Fachdisziplinen gesprochen und gleichermaßen verstanden wird, könnte den Menschen im übergreifenden Ast des kammförmigen Kompetenzprofils und in der interdisziplinären Zusammenarbeit über individuell ausgeprägte Wissensbasen und Lösungsräume hinweg unterstützen. Diese Überlegung ist Teil des Model-Based Systems Engineering (MBSE). Gemäß der „INCOSE Systems Engineering Vision 2025“, wird MBSE eine wichtige Rolle in der Produktentstehung einnehmen. Die INCOSE skizziert darin die Vision eines flächendeckenden Einsatzes bis zum Jahr 2025, zudem besonders außerhalb der bisherig überwiegenden Applikation in den Domänen Luft- und Raumfahrt, sowie Militärtechnik.¹³

¹¹ ALBERS ET AL. 2016a

¹² ALBERS ET AL. 2013a

¹³ INCOSE SE VISION PROJECT TEAM 2014

Teil von MBSE ist die **interdisziplinäre Systemmodellierung** mit einer gemeinsamen Sprache, zum Beispiel mit der Beschreibungssprache SysML.¹⁴ Diese stellt eine solche disziplinübergreifende gemeinsame Sprache dar. Ziel dieser Modellierung ist, mit einer standardisierten, fachdisziplinübergreifenden Sprache ein interdisziplinäres Systemmodell aufzubauen, das übergreifend von allen beteiligten Fachkräften trotz individuell ausgeprägter Wissensbasen und Lösungsräume verstanden und genutzt werden kann. Mit ihr kann Wissen kompetenzübergreifend abgebildet und zugänglich gemacht werden. Gegenstand der Modellierung mit SysML sind Blockschaltbilder, in welchen die technischen Aspekte eines Systems, wie Systemkontext, Anforderungen, Anwendungsfälle und Abläufe, sowie funktionale und logische Systemarchitektur und Testfälle, abgebildet werden können. Diese sind im eigentlichen Sinne Repräsentationen der in einem Modell gespeicherten Inhalte. Diese Bilder werden in der Modellierung mit SysML als „Diagramme“ bezeichnet. Sie dienen sowohl zur Visualisierung, als auch zur Exploration von systemischen Zusammenhängen.¹⁵ Zur Verdeutlichung werden in Bild 3 exemplarisch einige dieser Diagramme anhand des Beispiels einer Telematik- und Ladeinheit für Elektrofahrzeuge¹⁶ dargestellt.

Diese Arbeit greift das Thema interdisziplinäre Systemmodellierung als Baustein von MBSE auf. Es wird ein neuer anwenderorientierter Ansatz zur interdisziplinären Systemmodellierung konzipiert.

¹⁴ OMG 2015

¹⁵ FRIEDENTHAL ET AL. 2012

¹⁶ Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

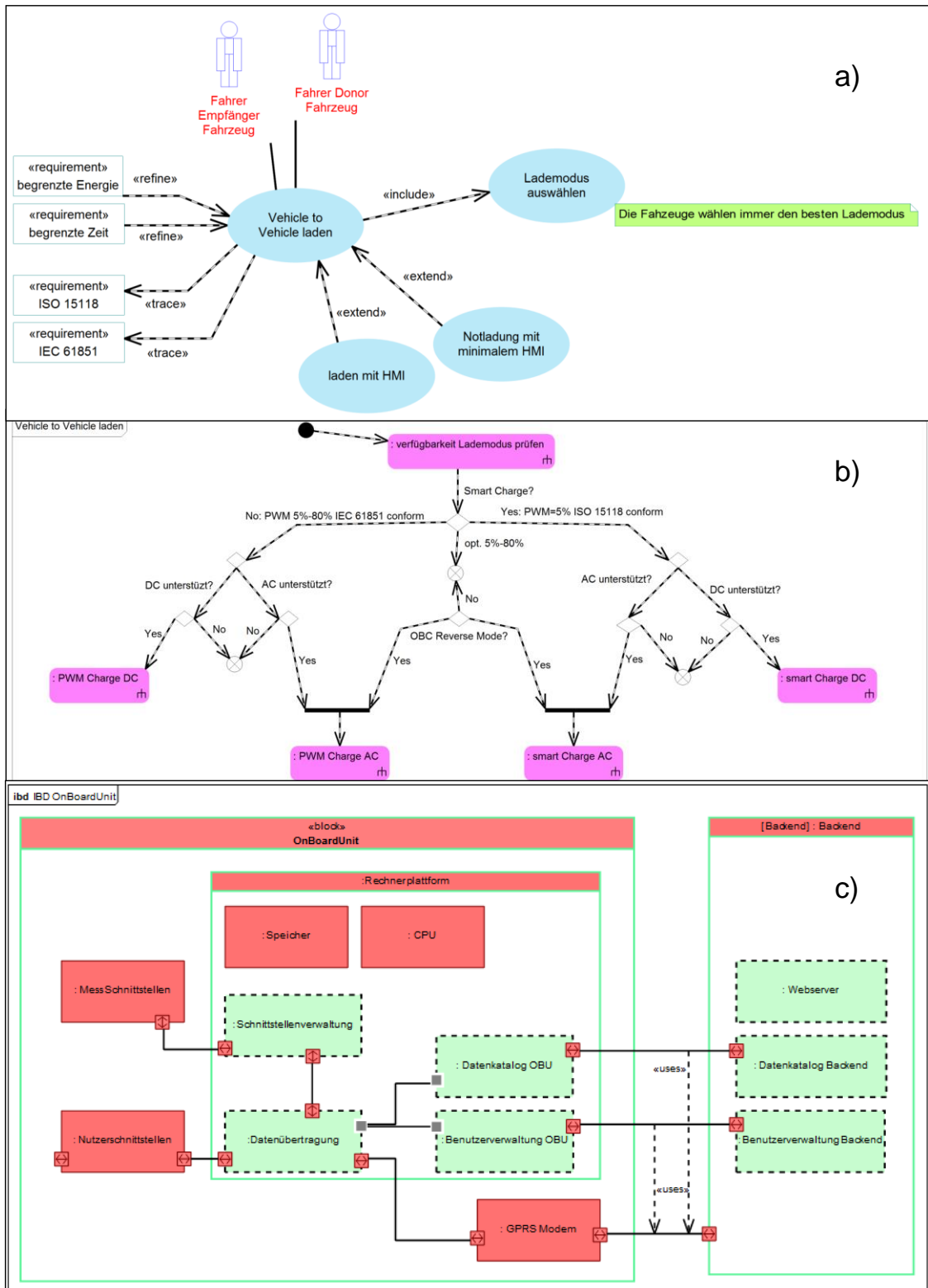


Bild 3: Auszug eines SysML-Modells: a) Anforderungen und Anwendungsfälle, b) Ablaufdiagramm, c) Systemarchitektur¹⁷ der Telematik- und Ladeinheit¹⁸ aus dem Abschlussbericht des Projekts ELISE¹⁹

¹⁷ ALBERS ET AL. 2013c

2 Grundlagen und Stand der Forschung

Nachdem in der Einleitung der Forschungsgegenstand im Kontext zu aktuellen Trends und dem Verständnis des Menschen gesetzt wurde, werden zunächst die Karlsruher Schule der Produktentwicklung – KaSPro erläutert und Begriffe erklärt, die in dieser Arbeit zum Verständnis des Stands der Technik und des Forschungsgegenstandes besondere Relevanz haben. Diese beinhalten die Begriffe Systems Engineering (SE), Model-Based Systems Engineering (MBSE) und interdisziplinäre Systemmodellierung.

Anschließend wird auf den Stand der Technik und Forschung der interdisziplinären Systemmodellierung eingegangen. Es werden hier bislang unveröffentlichte Teilergebnisse aus den Rohdaten einer im Stand der Forschung durchgeführten Studie vorgestellt, die die gegenwärtige Anwenderperspektive von interdisziplinärer Systemmodellierung mit SysML beleuchten.

2.1 Die Karlsruher Schule der Produktentwicklung – KaSPro

Die Karlsruher Schule der Produktentwicklung – KaSPro ist ein Ansatz mit neuen Methoden und Prozessen zur menschenzentrierten Produktentwicklung. Die KaSPro und ihre Bausteine gründen auf fünf zentralen Hypothesen von ALBERS nach Tabelle 1.

Tabelle 1: Die fünf Hypothesen der KaSPro nach ALBERS²⁰

H1	Jeder Produktentwicklungsprozess ist einzigartig und individuell
H2	Die Produktentwicklung kann verstanden werden, als Überführung eines (anfängs vagen) Zielsystems über ein Handlungssystem in ein Objektsystem
H3	Validierung ist die zentrale Aktivität der Produktentwicklung
H4	Die Transformation von Zielen in Objekte kann als Problemlösungsprozess betrachtet werden. Die erzeugten Objekte müssen dabei in Relation zu den Zielen beschrieben werden.
H5	Eine technische Funktion erfordert mindestens zwei Wirkflächenpaare, sowie deren verbindende Leitstützstruktur

¹⁸ Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

¹⁹ MUNKER ET AL. 2015

²⁰ ALBERS 2010

Insbesondere die Hypothese H1 und H2 machen deutlich, dass ein Produktentwicklungsprozess nicht als Ablauf definierter Schritte verstanden werden darf. ALBERS versteht die Produktentwicklung in der KaSPro daher als kontinuierlichen, iterativen Prozess in welchem sich das Ziel- und Objektsystem von einem anfangs vagen Reifegrad gemeinsam kontinuierlich entwickeln. Er fasst dieses in ein auf den fünf Hypothesen basierendes Meta-Modell, das integrierte Produktentstehungsmodell – iPeM²¹, welches mittlerweile um die Aspekte einer Produktgenerationsentwicklung (PGE) erweitert wurde.²² ALBERS begegnet mit dem iPeM als Baustein der KaSPro der in der Praxis erforderlichen Agilität und Flexibilität in der Produktentwicklung. Die Agilität und Flexibilität in der Produktentwicklung ist dadurch tief im Verständnis der KaSPro verankert.

2.2 Begriffsklärung

Der Begriff **Systems Engineering (SE)** ist von INCOSE geprägt. SE ist nach INCOSE definiert als interdisziplinärer Ansatz, um (technische) Systeme zu entwickeln.²³ Charakteristisch für SE sind das Anforderungsmanagement sowie die funktionale Systementwicklung, also die Entwicklung mit einer abstrakten Systembeschreibung auf funktionaler Ebene und das Denken in Abstraktionsebenen. SE wird dabei nicht nur als Vorgehen, sondern auch als Gesinnung oder auch als Rollenbeschreibung für einen Systemingenieur (englisch Systems Engineer) verstanden. SE berücksichtigt den kompletten Produktlebenszyklus bis hin zur Entsorgung.²³

SE und INCOSE haben ihre Wurzeln in der Luft- und Raumfahrt, sowie in den Verteidigungsprogrammen der USA. INCOSE wurde ursprünglich als NCOSE – National Council on Systems Engineering gegründet, um das Systems Engineering in den USA voranzutreiben. Doch auch außerhalb der USA wurden stets erfolgreiche technische Systeme entwickelt. WINZER leitet schon durch die Formulierung ihres Buchkapitels „Das Systems Engineering (SE) – altes Denken in neuem Gewand“²⁴ ein, dass die Grundphilosophie des SE generischen Charakter hat und im Wesentlichen mit Systemdenken und Systemtheorie, beispielsweise nach HABERFELLNER ET AL.²⁵ übereinstimmt. Wesentlichen Beitrag zur Systemtheorie liefert

²¹ ALBERS & BRAUN 2011

²² ALBERS ET AL. 2016e

²³ WALDEN ET AL. 2015

²⁴ WINZER 2013

²⁵ HABERFELLNER ET AL. 2012

beispielsweise ROPOHL mit seinen Arbeiten zur allgemeinen Systemtheorie²⁶ schon seit deren Erstauflage.²⁷

Eine zentrale Rolle im SE ist durch den Systemingenieur (englisch Systems Engineer) besetzt.²⁸ Ein äquivalentes Rollenbild ist im deutschen von ALBERS ET AL. im Rahmen einer acatech-Studie in Form des „Systemkonstruktors“ geprägt worden.²⁹ Obwohl begrifflich in den USA geprägt, scheint SE also auf einer international verbreiteten Grundphilosophie zu basieren. Der verstärkte, konsequente und flächendeckende Einsatz des SE ist ein Aspekt, den INCOSE in der „SE Vision 2025“³⁰, sowie ALBERS und GAUSEMEIER³¹ in den im Kapitel „Einleitung“ beschriebenen Zukunftsstudien, herausgearbeitet haben.

Das übergeordnete Ziel von „Model-Based“ Ansätzen wird vom INCOSE als Übergang von Dokumenten hin zu Computer-interpretierbaren Modellen formuliert. **Model-Based Systems Engineering (MBSE)** wird definiert als Anwendung von Modellierung zur Unterstützung von Aktivitäten zum Anforderungsmanagement, Systemgestaltung, Analyse, sowie Verifizierung und Validierung über den gesamten Produktlebenszyklus.²⁸ Es ist also die Nutzung von Modellen zur Unterstützung im oben beschriebenen Vorgehen des Systems Engineering. Hier sollte begrifflich zwischen Model-Based Systems Engineering (MBSE) und Model-Based Engineering (MBE) unterschieden werden.

Model-Based Engineering (MBE) umfasst alle Modelle im gesamten Produktlebenszyklus. MBSE kann dabei als Teildisziplin des MBE verstanden werden, die diejenigen Modelle umfasst, die Aspekte des Systems Engineering unterstützen.³² Dieses umfasst beispielsweise Modelle wie CAD, FEM oder CFD für MBE und überwiegend Modelle von Anforderungsmanagement, funktionaler Systembeschreibungen oder interdisziplinärer Systemmodellierung im MBSE.

Bild 4 stellt die Zuordnung dieser und der folgenden Begrifflichkeiten grafisch dar.

²⁶ ROPOHL 2009

²⁷ ROPOHL 1975

²⁸ WALDEN ET AL. 2015

²⁹ ALBERS ET AL. 2012a

³⁰ INCOSE SE VISION PROJECT TEAM 2014

³¹ ALBERS & GAUSEMEIER 2012

³² BERGENTHAL 2011

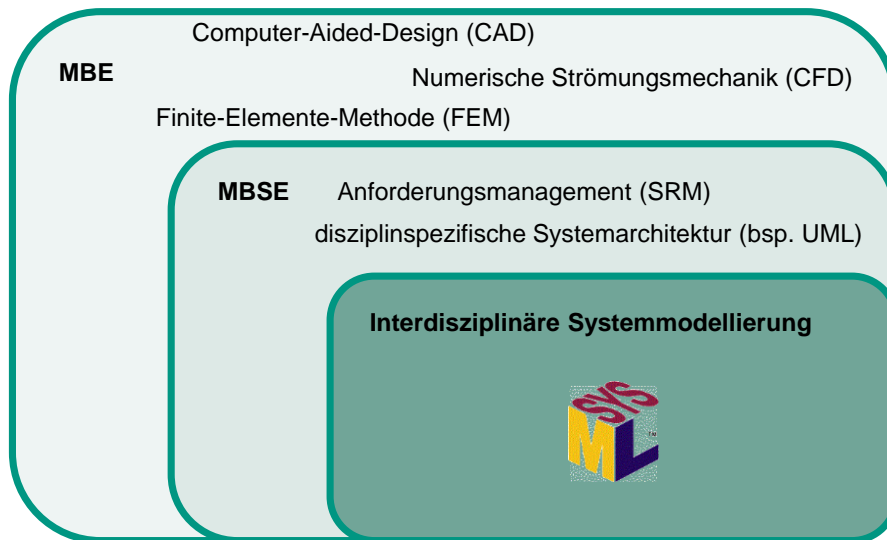


Bild 4: Interdisziplinäre Systemmodellierung als Teilmenge von MBSE als Teilmenge von MBE mit Beispielen

Die **interdisziplinäre Systemmodellierung** wird vom INCOSE als Baustein von MBSE verstanden.²⁸ Mit Hilfe der interdisziplinären Systemmodellierung können technische Systeme auf einer abstrakten Ebene allgemein verständlich für alle beteiligten Fachdisziplinen beschrieben werden. Dabei können alle technischen Aspekte von Anwendungsfällen und Anforderungen, über die Systemarchitektur bis hin zu Testfällen für Verifizierung und Validierung abgebildet werden. Besonders zu erwähnen ist in diesem Zusammenhang die von der Object Management Group (OMG – einem Konsortium für die Entwicklung von Standards), und der INCOSE in einem Gemeinschaftsvorhaben entwickelte und spezifizierte **Systems Modeling Language (SysML)**³³. SysML stellt eine standardisierte Sprache für die interdisziplinäre Systemmodellierung dar. Die interdisziplinäre Systemmodellierung wird in dieser Arbeit die zentrale Rolle einnehmen. Neben der Modellierung mit SysML existieren weitere relevante Ansätze zur interdisziplinären Systemmodellierung, auf die im Rahmen der wissenschaftlichen Untersuchungen dieser Arbeit in den folgenden Kapiteln im Detail eingegangen wird.

2.3 Erweitertes ZHO-Modell

ALBERS UND LOHMEYER betrachtet den Menschen im Zentrum der Produktentstehung ausgehend von der „liegenden Acht“³⁴. LOHMEYER entwickelt das Modell zum „erweiterte[n] ZHO-Modell“ weiter.³⁵ Gegenstand seiner Untersuchungen ist auch die Interaktion der Menschen in interdisziplinären Teams wie sie in Bild 5 dargestellt ist.

³³ OMG 2015

³⁴ ALBERS ET AL. 2011

³⁵ LOHMEYER 2013

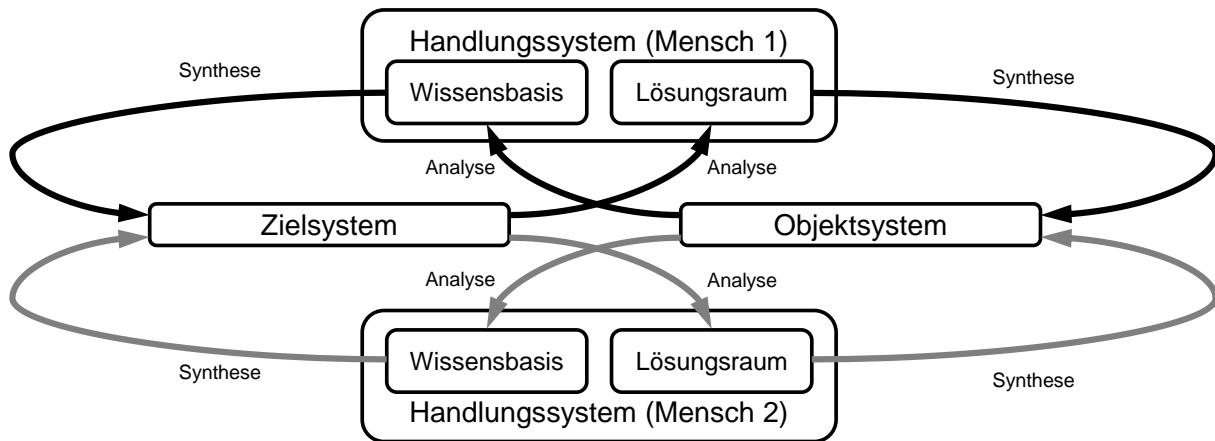


Bild 5: Zusammenarbeit zweier Menschen im Modell der „liegenden Acht“. Vereinfachte Darstellung nach Lohmeyer³⁵

Er versteht dabei die Menschen als Individuen mit individuellen Wissensbasen und individuellen Lösungsräumen, welche an einem gemeinsamen Ziel- und Objektsystem arbeiten. Dieses veranschaulicht die Gründe für Synergieeffekte und Missverständnisse in der interdisziplinären Zusammenarbeit. Synergieeffekte können in der Synthese und Analyse entstehen, wenn sich die individuellen Wissensbasen und Lösungsräume ergänzen. Missverständnisse können durch die fehlerhafte Interpretation in der Analyse bei überlappenden Anteilen der individuellen Wissensbasen und Lösungsräumen und bei redundanter Synthese entstehen, wenn sich die Individuen nicht ausreichend über die Ausprägung der Lösungsräume und Wissensbasen der beteiligten Menschen bewusst sind.

LOHMEYER³⁶ formuliert durch dieses menschenzentrierte Modell die zentralen Aspekte des Denkens und Handelns in der Entwicklungstätigkeit:

- **individuelle Denk- und Handlungsvorgänge des Menschen**

Das Vorgehen in der Entwicklungstätigkeit läuft nicht streng systematisch ab, sondern folgt einem opportunistischem Vorgehen. Die individuelle Wissensbasis und der individuelle Lösungsraum bilden die zentralen Elemente des Systems Mensch.

- **epistemische Unsicherheit**

Die epistemische Unsicherheit ist die Unsicherheit, die durch Erkenntnisgewinn verringert werden kann. Dahinter steht das Verständnis, dass eine vollständige Klärung der Aufgabenstellung zu Beginn des Entstehungsprozesses nicht möglich ist. Daran kann die co-evolutionäre Entwicklung von Ziel- und Objektsystem erklärt werden, bei der die

³⁶ LOHMEYER 2013

Unsicherheit im Laufe des Produktentstehungsprozesses kontinuierlich verringert wird.

- **iterative Entwicklungsprozesse**

Der iterative Charakter von Entwicklungsprozessen findet sich in der verschleiften Ausprägung des erweiterten ZHO-Modells in Form der „liegenden Acht“ wieder. Es erklärt dabei eine Iteration nicht als Rücksprung in eine vorhergehende Phase, sondern als Fortschritt im Wissen um Problem und Lösung und betont somit die essenzielle Notwendigkeit von Iterationen.

2.4 Die drei Säulen interdisziplinärer Systemmodellierung

Die interdisziplinäre Systemmodellierung stellt ein Mittel zur interdisziplinär verständlichen, abstrakten Darstellung der technischen Aspekte eines Systems dar. Dabei sollten die in Bild 6 dargestellten drei Säulen der interdisziplinären Systemmodellierung nach DELLIGATTI ET AL.³⁷ differenziert betrachtet werden.

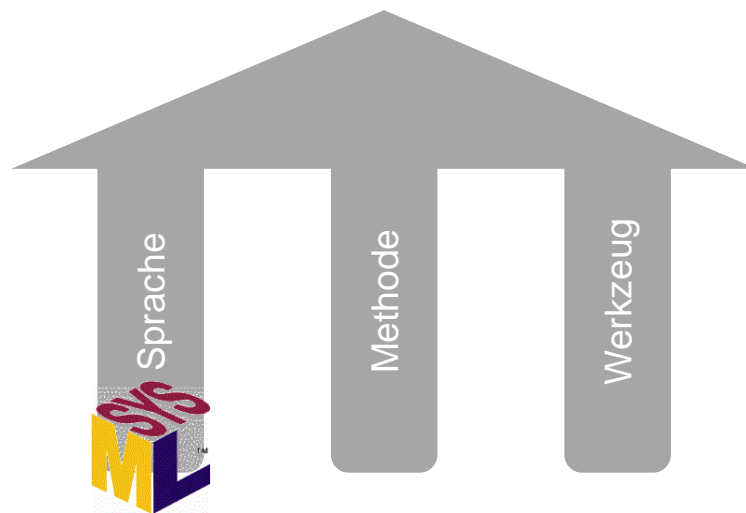


Bild 6: SysML im Kontext der drei Säulen der interdisziplinären Systemmodellierung. Eigene Darstellung nach DELLIGATTI ET AL.³⁸

Diese Säulen (Sprache, Methode und Werkzeug) lassen sich wie folgt differenziert beschreiben:

- **Sprache:** Die Sprache (beispielsweise SysML) stellt die Notation dar. Sie spezifiziert die Sprachelemente, die zur interdisziplinären Systemmodellierung zur Verfügung stehen.

³⁷ DELLIGATTI ET AL. 2014

³⁸ DELLIGATTI ET AL. 2014

- **Methode:** Die Methode gibt das Vorgehen vor. Sie spezifiziert, wann welche Aspekte mit welchen Sprachelementen modelliert werden und wie das Ergebnis der Modellierung aussehen soll. Siehe zur Modellierung beispielsweise auch FRIEDENTHAL ET AL.³⁹
- **Werkzeug:** Mit dem Werkzeug kann die Modellierung durchgeführt werden. Im Sinne des MBSE soll es sich hier um ein Software-Werkzeug handeln, welches die Modellinhalte modellbasiert abbildet. Es erlaubt die Handhabung der Sprachelemente. Mit Hilfe einer Methode generiert der Anwender daraus ein interdisziplinäres, vernetztes und computerinterpretierbares Modell.

Ansätze der interdisziplinären Systemmodellierung, die nur eine der Säulen spezifizieren und die anderen Säulen anerkennen aber losgelöst von deren Spezifikation betrachten, werden im Folgenden als **nicht integrierte Ansätze** bezeichnet.

Ansätze der interdisziplinären Systemmodellierung, die mindestens zwei oder alle drei Säulen integrativ spezifizieren, werden im Folgenden als **integrierte Ansätze** bezeichnet.

Die interdisziplinäre Systemmodellierung lässt sich auch ohne Software-Unterstützung, beispielsweise mit Metaplankarten auf einer Metaplanwand anwenden, wie es DÄNZER ET AL. von der Applikation mit der FAS-Methode aus der Praxis berichten.⁴⁰ Dieses würde dann als ein Artefakt des SE verstanden werden, jedoch durch Modellierung mit einem Software-Werkzeug als Computerinterpretierbares Modell in ein Artefakt des MBSE überführt werden können.

Die SysML stellt eine der bedeutendsten Sprachen zur interdisziplinären Systemmodellierung dar. Sie nimmt aber das weitere Vorgehen zur Modellierung oder die Umsetzung im Werkzeug nicht vorweg. SysML soll dabei ausschließlich die Säule der Sprache repräsentieren.⁴¹ Entschließt sich der Anwender, mit SysML zu modellieren, muss dazu eine Methode und ein Werkzeug ausgewählt werden.

2.5 Relevante Ansätze zur interdisziplinären Systemmodellierung

Es existiert eine Vielzahl von Ansätzen und Techniken zur interdisziplinären Systemmodellierung. ZINGEL⁴² gibt einen umfassenden Überblick über Sprachen, Methoden, Technologien und Ansätze der interdisziplinären Systemmodellierung. Für weitere Untersuchungen sind hier die repräsentativen Ansätze der interdisziplinären

³⁹ FRIEDENTHAL ET AL. 2012

⁴⁰ DÄNZER ET AL. 2014

⁴¹ OMG 2012

⁴² ZINGEL 2013

Systemmodellierung ausgewählt worden. Dies ist anhand der Ergebnisse einer INCOSE-Umfrage⁴³ erfolgt. Diese Umfrage hatte das Ziel, repräsentative Prozesse, Methoden und Werkzeuge des MBSE als strategische Informationsgrundlage für die INCOSE zu identifizieren. Ansätze aus dieser Umfrage wurden daher als besonders relevant betrachtet. Unter Berücksichtigung der 3-Säulen Perspektive lassen sich die Ansätze nach ihren Aspekten Sprache, Methode und Werkzeug extrahieren. Weiterhin wurden CONSENS⁴⁴, ein Ansatz mit akademischen Hintergrund, und RFLP, ein im CAD Werkzeug „Catia V6“ implementierter und an VDI 2206 angelehnter Ansatz⁴⁵, in den Untersuchungsraum aufgenommen.

Diese Auflistung stellt eine Auswahl auf Basis der INCOSE Studie dar und erhebt keinen Vollständigkeitsanspruch. Sie bildet die Grundlage für die weiteren Untersuchungen. Weitere, durch kontinuierliche Forschung entstehende neue Ansätze stellen beispielsweise die funktionale Produktbeschreibung (FPD Modell) im Rahmen der SE-VPE Methode⁴⁶ oder die Modellierung von Zielsystemen⁴⁷ dar und können mit geeigneten Erweiterungsmechanismen⁴⁸ auch in dem in dieser Arbeit erarbeiteten Ansatz abgebildet werden.

Nach gegenwärtigem Kenntnisstand konnte sich neben SysML keiner dieser Ansätze in der praktischen Anwendung in der Breite durchsetzen. Die Ansätze werden im Folgenden umrissen und es wird eine erste Einschätzung zu deren Stärken und Schwächen gegeben.

2.5.1 OMG SysML

Auf die von der Object Management Group (OMG) spezifizierte Systems Modeling Language (SysML) wurde bereits in den obigen Kapiteln vorgestellt. OMG SysML existiert seit September 2015 in der Version 1.4⁴⁹, ist aber aufgrund der zeitlichen Ausprägung des Untersuchungszeitraumes in der damals verfügbaren Version 1.3⁵⁰ untersucht worden. Der Sprung zur Version 1.4 weist jedoch keine grundlegenden Änderungen im Ansatz und im Sprachraum der SysML auf, so dass auch weiterhin von einer Validität der Untersuchung auf Basis von Version 1.3 ausgegangen werden kann. Passende Methoden sind beispielsweise die INCOSE Object-Oriented

⁴³ ESTEFAN 2007

⁴⁴ FRANK 2006

⁴⁵ KLEINER & KRAMER 2013

⁴⁶ GILZ 2014

⁴⁷ EBEL 2015

⁴⁸ Erweiterbarkeit des Ansatzes vgl. Kapitel 3.2 Zielsetzung

⁴⁹ OMG 2015

⁵⁰ OMG 2012

Systems Engineering Method (OOSEM)⁵¹, die Systems Modeling Toolbox (SYSMOD)⁵², sowie die Modellierung nach dem Karlsruher Lehransatz SysKIT⁵³. Als Werkzeuge zur Modellierung kommen beispielsweise Atego Artisan Studio (nun PTC Integrity Modeler)⁵⁴, Cameo Systems Modeler⁵⁵, Sparx Systems Enterprise Architect⁵⁶ in Betracht.

Stärken: Standardisierte Sprache der Systems Engineering Gemeinschaft.

Schwächen: Nicht integrierter Ansatz. Aus Anwenderperspektive wird das Potenzial nicht ausgeschöpft.⁵⁷

2.5.2 IBM Telelogic Harmony-SE

Obwohl Harmony-SE Werkzeugunabhängig gestaltet sein soll, besteht eine enge Verbindung zu dem IBM Werkzeug Rhapsody.⁵⁸ Harmony-SE ist im Kern eine Methodenspezifikation zur Modellierung von Anforderungen, Verhalten, sowie funktionaler und logischer Systemarchitektur. Als Sprache wird eine Teilmenge der SysML verwendet.⁵⁸ Detaillierte Informationen zu Sprache und Methode zur Analyse des Ansatzes sind dem „Harmony-SE Deskbook“⁵⁹ entnommen worden.

Stärken: Durchgängig beschriebene Methode zur Modellierung verfügbar.

Schwächen: Streng sequenzielle Methode, dies ist mit dem Prozessverständnis der Karlsruher Schule⁶⁰ nicht vereinbar.

2.5.3 IBM RUP SE

IBM Rational Unified Process for Systems Engineering (RUP SE) ist ein Handlungssystem für die Produktentstehung mit Fokus auf die Aktivitäten des Systems Engineering.⁶¹ Für die Aktivität der interdisziplinären Systemmodellierung wird die Sprache SysML verwendet. RUP-SE ist eng verzahnt mit den Werkzeugen der IBM Rational Familie.⁶¹

⁵¹ FRIEDENTHAL ET AL. 2012

⁵² WEILKIENS 2008

⁵³ MATTHIESEN ET AL. 2014

⁵⁴ PTC 2014

⁵⁵ NO MAGIC 2012

⁵⁶ SPARX 2015

⁵⁷ Vgl. Kapitel 2.7 Gegenwärtige Anwendersicht zur Modellierung mit SysML

⁵⁸ ESTEFAN 2007

⁵⁹ HOFFMANN 2014

⁶⁰ ALBERS 2010

⁶¹ ESTEFAN 2007

Stärken: Parallelisierte Aktivitäten nach dem Prozessverständnis der Karlsruher Schule.⁶²

Schwächen: Fokussiert sich als Handlungssystem global auf den Lebenszyklus der Produktentwicklung. Interdisziplinäre Systemmodellierung stellt darin eine nicht näher beschriebene Teilaktivität dar.

2.5.4 VITEC MBSE

Als unabhängige Methode bildet VITEC MBSE mit der engen Bindung an das VITEC CORE Werkzeug unter Verwendung der Sprache VITEC System Definition Language (SDL) dann doch einen integrierten Ansatz der Firma VITEC Corporation.⁶¹ Detaillierte Beschreibung zu Sprache, Methode und Werkzeug lässt sich den Umdrucken der Firma VITEC entnehmen.⁶³

Stärken: Integrierter Ansatz. Ein Zwiebelschalen-Modell („Onion Model“) beschreibt das Vorgehen in untereinander konsistenten Ebenen entsprechend eines Stage-Gate Prozesses. Während der Phasen wird ein ganzheitliches Systemmodell erstellt, welches sich über die Stages schrittweise im Reifegrad erhöht. Mögliche phasenübergreifende Iterationen werden nicht angegeben. Das Vorgehen entspricht somit nicht vollständig dem Verständnis einer agilen Produktentwicklung nach der KaSPro⁶⁴, liegt jedoch besonders im Vergleich zu den streng sequenziell ausgeprägten Ansätzen nahe an der KaSPro. Dieses Verständnis einer stufenweisen Reifegradentwicklung könnte eine Grundlage für ein agiles Vorgehen bilden und kann daher als Stärke gewertet werden.

Schwächen: Sehr umfangreich und formalisiert spezifizierter Sprachraum.

2.5.5 JPL SA

Die State Analysis des NASA Jet Propulsion Laboratory (JPL) stellt einen integrierten Ansatz mit Werkzeugunterstützung zur Verhaltensbeschreibung dar. Mit starkem Fokus auf die Zustandsmodellierung und einem spezifischen Sprachraum scheint JPL SA auf die Modellierung von Command&Control Strukturen, wie Satellit und Bodenstation, in der Raumfahrt zugeschnitten zu sein.⁶¹

Stärken: Durchdachte Zustandsmodellierung.

Schwächen: Spezifische Ausrichtung auf Anwendungsfälle der Raumfahrt.

⁶² ALBERS 2010

⁶³ VITECH 2011; 2013a; 2013b

⁶⁴ Vgl. Kapitel 2.1 Die Karlsruher Schule der Produktentwicklung – KaSPro

2.5.6 Dori OPM

Die Object-Process Methodology (OPM) von DORI ist mit inkludiertem „System Development Process“ ein integrierter Ansatz.⁶⁵ Detaillierte Informationen zur Analyse des Ansatzes sind dem Buch von DORI zu OPM⁶⁶ entnommen worden.

Stärken: Integrierter Ansatz. Verzicht auf Redundanz durch Verzicht auf Diagrammspezifikation. Schlanker, reduzierter Sprachraum. Kompakte Verhaltensmodellierung durch Fokussierung auf Zustandsmodellierung.

Schwächen: Akademische Perspektive, Anwendungsbeispiele aus der Praxis konnten nicht gefunden werden. Sprachraum ist eingeschränkt auf Architekturmodellierung.

2.5.7 CONSENS

CONSENS ist eine Spezifikationstechnik zur Beschreibung der Prinziplösung und ist mit Fokus auf selbstoptimierende Systeme entwickelt worden.⁶⁷ Obwohl für CONSENS ein exemplarisches Werkzeug erstellt wurde, liegt der Fokus auf der Methode. CONSENS definiert aber auch Sprachelemente. Mit dem Profil SysML4CONSENS kann nach CONSENS in SysML modelliert werden.⁶⁸

Stärken: Systematischer Aufbau und schlanker Umfang des Sprachraums.

Schwächen: Vernetzung kann nur sternförmig zentral zur Systemarchitektur erfolgen. Vernetzung erfolgt auf Partialmodelle, also auf eine Gruppierung, statt auf einzelne Sprachelemente.

2.5.8 RFLP

Unter RFLP sei hier die Umsetzung des RFLP Ansatzes im CAD Werkzeug Dassault Systèmes Catia V6 verstanden. RFLP ist ein Akronym für Requirement, Functional, Logical, Physical und basiert auf dem an das V-Modell der VDI 2206 angelehnten AFLP (Anforderung, Funktion, logische Lösungselemente, physikalische Lösungselemente) von EIGNER UND GILZ⁶⁹. Details zu RFLP in Catia V6 können der Veröffentlichung von KLEINER UND KRAMER⁷⁰ entnommen werden. Für die Analyse des Ansatzes sind weiterhin Beobachtungen des Autors aus einer eintägigen Anwenderschulung inkludiert worden.

⁶⁵ ESTEFAN 2007

⁶⁶ DORI 2011

⁶⁷ FRANK 2006

⁶⁸ IWANEK ET AL. 2013

⁶⁹ EIGNER & GILZ 2012

⁷⁰ KLEINER & KRAMER 2013

Stärken: integrierter Ansatz, schlanker Sprachraum, Verzicht auf Redundanz durch Verzicht auf Diagrammspezifikation.

Schwächen: Sprachraum eingeschränkt (Aspekte Anwendungsfälle und Validierung fehlen).

2.6 Systemmodellierung in der Produktgenerationsentwicklung

Untersuchungen von ALBERS ET AL. zeigen, dass die Produktentwicklung in der Praxis in den wenigsten Fällen auf der „grünen Wiese“ startet. Sie bilden ein Beschreibungsmodell, das die Perspektive einer Produktgenerationsentwicklung (PGE) mit Anteilen aus Gestaltvariation, Prinzipvariation und Übernahmevariation beschreibt.⁷¹ Neue Produktgenerationen basieren nach diesem Beschreibungsmodell auf Referenzprodukten mit einer grundsätzlichen Struktur, welche von Vorgänger- oder auch Wettbewerbsprodukten stammt. Das PGE Konzept versteht eine grundsätzliche Neuentwicklung nur als ein Extrem des PGE-Ansatzes, so dass auch diese Fälle beschrieben sind. Die PGE lässt sich beispielhaft am Zweimassenschwungrad, einem Schwingungsdämpfer im Antriebsstrang von Verbrennungskraftfahrzeugen, nachvollziehen.⁷² Sind die Variationsanteile bekannt und beschrieben, lassen sich daraus Risikobewertungen und damit Entscheidungsgrundlagen zur Unterstützung im Produktentwicklungsprozess ableiten.⁷³ Als wesentlicher Baustein der KaSPro, findet sich die PGE im Meta-Modell iPeM wieder.⁷⁴

Die interdisziplinäre Systemmodellierung kann hier genutzt werden, indem auf Modelle aus vergangenen Produktgenerationen zurückgegriffen wird. ALBERS ET AL. erkennen, dass dadurch Wissen und wesentliche Informationen einfließen und auch der Modellierungsaufwand durch den Übertrag reduziert werden kann.⁷⁵ Sie schaffen dazu ein Framework, in das die Systemmodelle eingeordnet werden können. Dieses Framework nach Bild 7 ordnet die Abstraktionsgrade der Systemmodellierung in einer Matrixdarstellung mit den Dimensionen Generisch-zu-Systemspezifisch, sowie Metamodellierung-zu-Systemmodellierung ein.

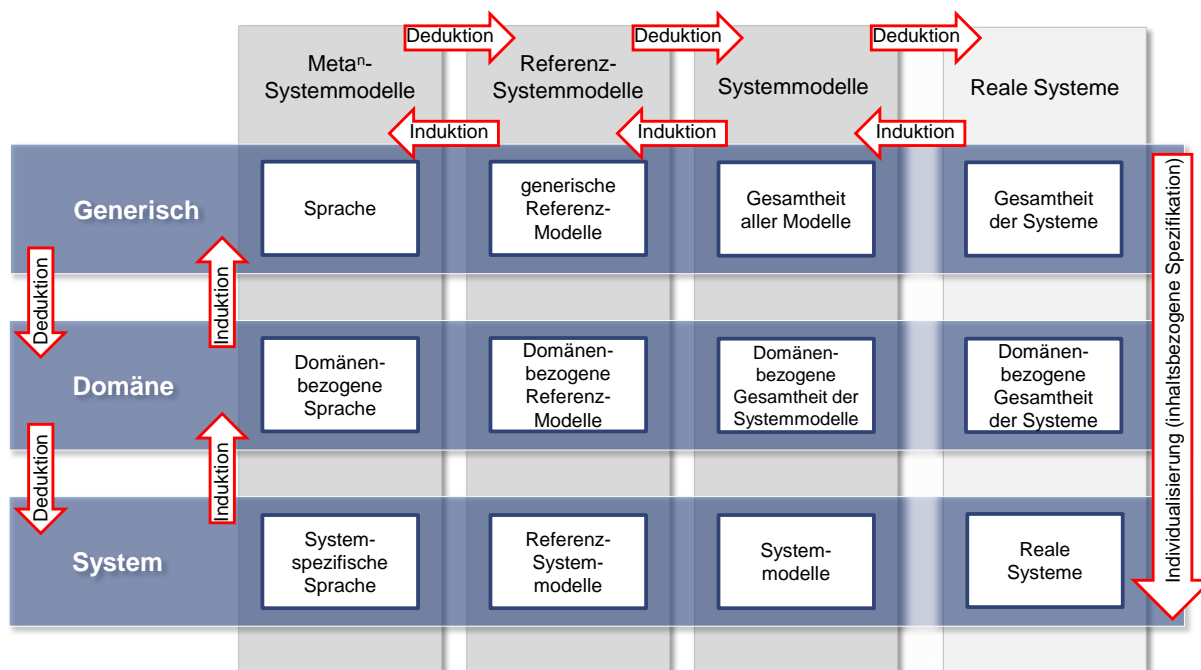
⁷¹ ALBERS ET AL. 2015a

⁷² ALBERS ET AL. 2016b

⁷³ ALBERS ET AL. 2016f

⁷⁴ ALBERS ET AL. 2016e

⁷⁵ ALBERS ET AL. 2014

Bild 7: Framework: Abstraktionsgrade der Systemmodellierung⁷⁵

Die interdisziplinäre Systemmodellierung kann nach Bild 8 entlang einer Diagonale im Framework angeordnet werden, deren Abstraktionsgrade durch ALBERS ET AL.⁷⁵ wie folgt beschrieben werden: Während in das Feld der „**Sprache**“ generische Systemmodellierung, wie z. B. die Modellierung mit SysML, eingeordnet werden, bildet die generalisierte Struktur eines spezifischen Systemmodells ein **domänenbezogenes Referenzmodell**. Dies kann im Sinne der PGE als Referenz in nachfolgende Systemmodelle überführt werden. Die **Gesamtheit der domänenbezogenen Systemmodelle** bildet eine Bibliothek von Referenzmodellen für die PGE in dem jeweils individuellen Kontext des entwickelnden Unternehmens. Aus den domänenbezogenen Referenzmodellen können die **Referenz-Systemmodelle** abgeleitet werden, welche denjenigen Anteil der strukturellen Inhalte der domänenbezogenen Referenzmodelle enthalten, welcher spezifisch relevant für das zu entwickelnde System ist. Im Bereich der **domänenbezogenen Sprachen** kann die generische Beschreibungssprache durch Erweiterungsmechanismen an den konkreten Anwendungsfall angepasst werden. Des Weiteren werden hier spezifische domänenbezogene Sprachen eingeordnet. In das Feld der **generischen Referenz-Modelle** fallen die Modellierungsmethoden zur Anwendung der generischen Sprachen, die auch im Rahmen dieser Arbeit gesondert betrachtet werden.⁷⁶

⁷⁶ Vgl. Kapitel 2.4 Die drei Säulen interdisziplinärer Systemmodellierung

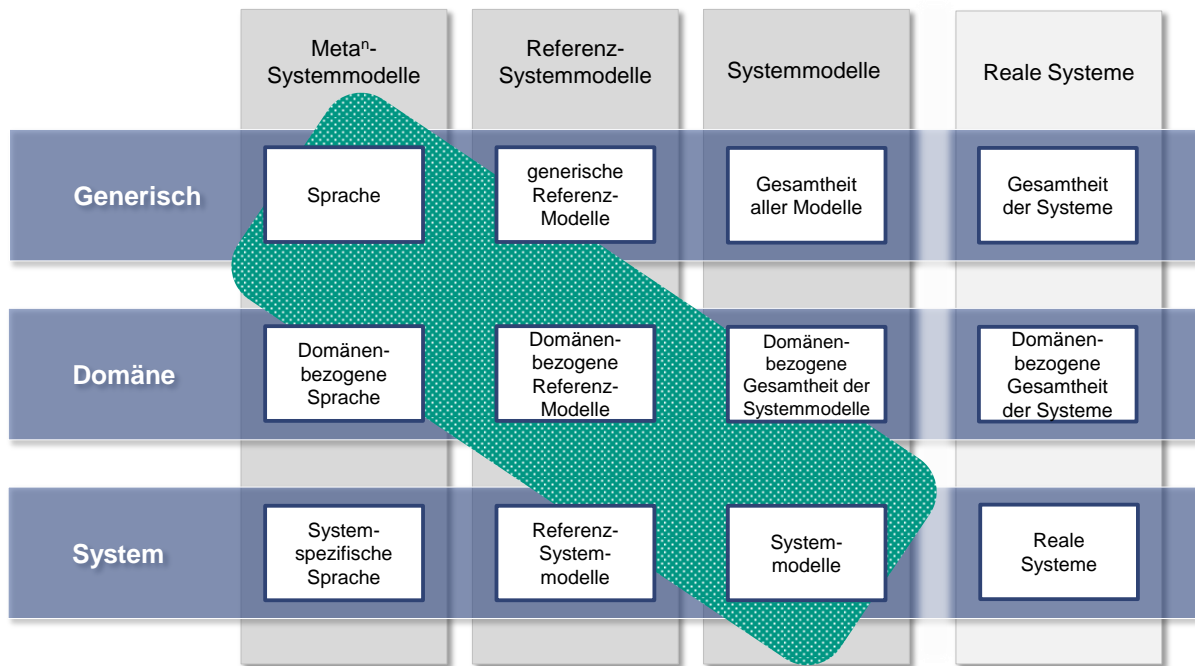


Bild 8: Interdisziplinäre Systemmodellierung als Diagonale im Framework der Abstraktionsgrade der Systemmodellierung, vereinfachte Darstellung des Frameworks⁷⁷

Entlang dieser Diagonale können Forschungsfelder der interdisziplinären Systemmodellierung aufgeschlüsselt werden. ALBERS ET AL. geben damit einen Überblick über die Forschungsaktivitäten in der Karlsruher Schule.⁷⁸ Im Folgenden werden diese und weitere Forschungsaktivitäten außerhalb der Karlsruher Schule dargestellt. Diese weiteren Forschungsfelder lassen sich hier kompakt umreißen. Einen vertieften Einblick in das Feld der interdisziplinären Systemmodellierung leistet beispielsweise ZINGEL⁷⁹.

Im Bereich der **generischen Meta-Modellierung** werden im akademischen und industriellen Umfeld Ansätze zur Systemmodellierung durch Anpassung der Sprache weiter oder neu entwickelt. Mit der Verbreitung und Standardisierung sind vornehmlich Arbeiten basierend auf der SysML zu erkennen, beispielsweise durch Entwicklung eines Sprachprofils „SysML4CONSENS“⁸⁰ für die Verwendung des CONSENS Ansatzes oder Ansätze zur funktionsorientierten Modellierung von Produkten.⁸¹ Zu den nicht auf SysML basierenden ist die Modellierung mit

⁷⁷ ALBERS ET AL. 2014

⁷⁸ ALBERS ET AL. 2016d

⁷⁹ ZINGEL 2013

⁸⁰ IWANEK ET AL. 2013

⁸¹ DINGER ET AL. 2012

fundamentalen Elementtypen nach dem Ansatz „Functional Modeling Concept“ (FMC)⁸² zu zählen.

Durch Ableitung **generischer Referenzmodellierung** werden Methoden der generischen Systemmodellierung erforscht. Auf SysML basierende Arbeiten sind die Entwicklung von Modellierungsmethoden, ergänzt mit einem SysML Sprachprofil, beispielsweise die FAS Methode⁸³ von LAMM UND WEILKIENS, oder die von ZINGEL entwickelte Sprache zur Modellbildung technischer Systeme.⁷⁹

In den Bereich der **Referenz-Modelle** fallen Arbeiten des IPEK im Bereich der Anwendung für Baukastenmodellierung⁸⁴ sowie die Forschungsarbeiten im Rahmen der Anwendungsstudien.⁸⁵ SCHERER befragt 30 Entwickler eines deutschen Automobilherstellers zur Baukastenentwicklung. Er berichtet im Rahmen seines Forschungsberichtes von einem Bedarf an methodischer Unterstützung im Anforderungsmanagement zum Umgang mit Zielen, Anforderungen und Randbedingungen, was die Arbeit in der Baukastenentwicklung erleichtern könnte. Er begegnet diesem Bedarf mit dem Einsatz von interdisziplinärer Systemmodellierung.⁸⁶ ALBERS, KURRLE und KLINGLER befragen 45 Experten der deutschen Automobilindustrie verschiedener Disziplinen zu Herausforderungen in der Entwicklung von System-of-Systems am Beispiel der Vernetzung von Fahrzeugen, begrifflich auch unter „Connected Car“ bekannt. Auch sie erkennen den Bedarf an methodischer Unterstützung im Umgang mit Anforderungen, Zielen und Randbedingungen.⁸⁷

Zur Anknüpfung der interdisziplinären Systemmodellierung an **domänenbezogene Sprachen** sind die Arbeiten zur Kopplung von SysML Modellen an das Product Lifecycle Management (PLM) zu nennen, die mit einem Sprachprofil zur Systemmodellierung SE-VPE⁸⁸ einhergehen. Das Forschungsprojekt mecPro² widmet sich der globalen und ganzheitlichen Einbindung in den Produktentstehungsprozess.⁸⁹ Arbeiten am IPEK greifen sich den Aspekt der Kopplung zu domänenspezifischen Partialmodellen des Maschinenbaus heraus.⁹⁰

⁸² VON DUNGERN

⁸³ LAMM & WEILKIENS 2010

⁸⁴ ALBERS ET AL. 2015b

⁸⁵ Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

⁸⁶ SCHERER 2016

⁸⁷ ALBERS ET AL. 2016c

⁸⁸ GILZ 2014

⁸⁹ EIGNER ET AL. 2015

⁹⁰ MOESER ET AL. 2015

2.7 Gegenwärtige Anwendersicht zur Modellierung mit SysML

Von besonderer Relevanz für die OMG ist die Anwendersicht zu SysML, da die OMG als Standardisierungskonsortium die Weiterentwicklung des SysML Sprachstandards vorantreibt. Während in Reaktionen auf eine Umfrage der OMG⁹¹ („Request for Information“) die Modellierung mit SysML im Allgemeinen als gut wahrgenommen bewertet wird, lässt sich aus den abschließenden Empfehlungen noch Verbesserungspotenzial ableiten. Wesentliche Erkenntnis daraus ist, dass die Lernkurve von SysML als steil wahrgenommen wird, da sowohl die Sprache SysML, eine Methode und ein Werkzeug erlernt werden müssen. Die OMG leitet daraus die Empfehlung ab, Sprache, Methode und Werkzeug in Zukunft noch strikter getrennt zu vermitteln, damit der Lernfortschritt losgelöst stattfinden und für den Anwender vereinzelt werden kann.⁹¹

Eine Umfrage des IPEK zeigt, dass bei den Befragten das Potenzial der interdisziplinären Systemmodellierung mit SysML anerkannt, deren Leistungsfähigkeit in der Praxis aber noch als gering eingeschätzt wird.⁹² Befragt wurden hierzu 10 Ingenieure aus der industriellen Praxis, die in unterschiedlichen Branchen tätig sind. Wobei nur diejenigen eine Einschätzung zur Modellierung mit SysML abgeben konnten, die SysML bereits aus dem Einsatz in der Praxis kannten.

Eine weitere Umfrage des IPEK hat das Verständnis des Funktionsbegriffes im Fokus.⁹³ In einer abschließenden offenen Frage wird der Anwender zu Verbesserungsmöglichkeiten der SysML befragt. Das Panel dieser Umfrage besteht aus Systemingenieuren aus Deutschland und Österreich aus Industrie und Wissenschaft mit der Zugehörigkeit zu unterschiedlichen Fachdisziplinen. Von den insgesamt 50 Rückmeldenden dieser Umfrage hat eine Teilmenge von 24 diese Frage beantwortet, welche alle mindestens eine Grundlagenkompetenz in Form von Lesekompetenz von SysML-Diagrammen angegeben haben. Der Großteil der Rückmeldenden hat mindestens eine Grundlagenkompetenz in Form von Modellierungskompetenz angegeben, die granulare Aufschlüsselung kann Bild 9 entnommen werden. Aus diesen Antworten lassen sich 30 bisher unveröffentlichte auswertbare Einzelnennungen wie in Bild 9 dargestellt, extrahieren.

⁹¹ BONE & CLOUTIER 2010

⁹² ALBERS ET AL. 2012b

⁹³ ALBERS & ZINGEL 2013

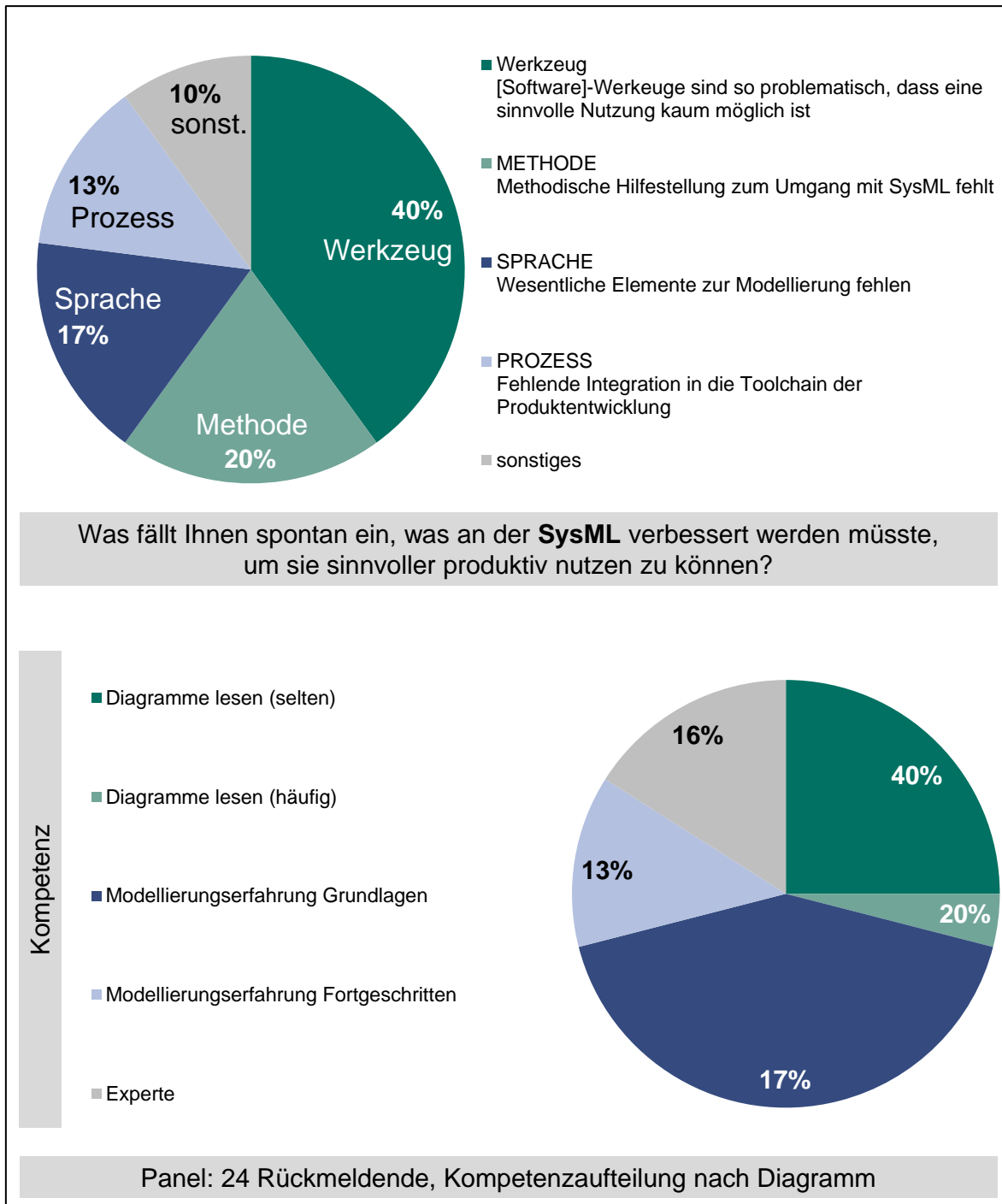


Bild 9: Anwenderperspektive SysML.

Eigene Darstellung bisher unveröffentlichter Teilergebnisse einer Anwenderbefragung.⁹³

Obwohl explizit nach SysML gefragt wurde, enthalten die Rückmeldungen Aspekte zu allen Säulen der interdisziplinären Systemmodellierung (Sprache, Methode und Werkzeug). Dies lässt darauf schließen, dass die Trennung der drei Säulen in der Praxis für den Anwender keine Relevanz besitzt oder von ihm nicht wahrgenommen wird.

2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

Im Rahmen dieser Arbeit sind Anwendungsstudien durchgeführt worden, um einen Eindruck der Praxistauglichkeit interdisziplinärer Systemmodellierung zu erhalten. Hierfür ist die interdisziplinäre Systemmodellierung mit SysML unter Verwendung aktueller Modellierungswerkzeuge in zwei Forschungsprojekten eingesetzt worden. Die Anwendungsstudien umfassten die projektbegleitende Modellierung eines innovativen Hochvoltspeichers für Elektrofahrzeuge im Projekt NEXHOS⁹⁴ und eines verteilten Telematik- und Datensystems mit autonomer Ladeinheit für Elektrofahrzeuge im Projekt ELISE⁹⁵. An beiden Studien konnte beobachtet werden, dass zur Erreichung eines praktischen Nutzens der interdisziplinären Systemmodellierung mit SysML noch eine umfassende Methodenentwicklung zur Umgehung von Modellierungsproblemen erforderlich ist:

Gegenstand des Projekt NEXHOS war die Konzipierung eines Hochvoltspeichers der nächsten Generation für Plug-In Hybride im Verbund mit Vertretern von Automobilzulieferern unter Federführung eines OEM. Die interdisziplinäre Systemmodellierung mit SysML war hier als begleitende Unterstützung im Entwicklungsprozess vorgesehen worden. Dies umfasst die Sammlung der Anforderungen von OEM und Zulieferer und die Integration derer ins Modell, sowie die modellbasierte Abbildung der Systemarchitektur anhand von Informationen aus dem fachlichen Austausch mit den Entwicklern. Bereits frühzeitig musste auf den Austausch modellbasierter Informationen in Form von Diagrammen zugunsten der Akzeptanz verzichtet werden: Es wurden vereinfachte Steckbriefe ohne formale Notation dem Austausch von Diagrammen mit SysML-Notation vorgezogen, woraus sich jedoch keine Aussagen zur Praxistauglichkeit der Modellierung mit SysML ableiten lassen, da auf eine vorherige Schulung der Projektpartner verzichtet wurde. Die Entwickler im Projekt waren nach funktionalen Einheiten statt nach Baugruppen organisiert, womit die Abbildung einer funktional gruppierten Systemarchitektur, im Projekt als „virtuelle Komponenten“ bezeichnet, neben einer konventionell gegliederten Systemarchitektur nach physikalischer Allokation in Baugruppen erforderlich war. Dies hat die Entwicklung einer Modellierungsmethode zur konsistenten Abbildung mehrerer Systemstrukturen im Modell erfordert.⁹⁶ Wesentliche Erkenntnis aus dieser Anwendungsstudie ist, dass im Falle des Projekts NEXHOS eine interdisziplinäre Systemmodellierung mit SysML nur durch maßgebliche Entwicklungsarbeit an Modellierungsmethode und –Sprache geleistet

⁹⁴ WAGNER ET AL. 2013

⁹⁵ MUNKER ET AL. 2014b

⁹⁶ MUNKER ET AL. 2014a

werden konnte, wobei trotz Sprachenkonformität die universelle Anwendbarkeit in verschiedenen Entwicklungswerkzeugen nicht garantiert werden konnte.⁹⁷

Gegenstand des Projekts ELISE war die Konzipierung eines systemintegrierten Daten-Gateways für Elektrofahrzeuge, bestehend aus einer Telematikeinheit im Fahrzeug, welche Daten sammelt, und einem Backend bestehend aus Servern des Projektpartners, welche die Daten entgegennehmen und weiterverarbeiten. Die interdisziplinäre Systemmodellierung mit SysML war hier als begleitende Unterstützung zur Anforderungsanalyse und zum Systementwurf vorgesehen worden. Im Rahmen von Recherchen zu Verbundprojekten und einer Expertenbefragung ist versucht worden zu erfassen, an welcher Art von Fahrzeugdaten in einem multimodalen Mobilitätssystem Bedarf besteht. Diese sind als Anforderungen modelliert worden und konnten im Systemmodell durch Erfüllungsbeziehungen in Relation zur Systemarchitektur gesetzt werden. Eine Erfüllungsprüfung der Anforderungen war somit modellbasiert möglich. Zur Modellierung der Systemarchitektur war es erforderlich, eine Modellierungsmethode zur Abbildung von Hard- und Softwareanteilen im Modell zu entwickeln.⁹⁸ Wesentliche Erkenntnis aus dieser Anwendungsstudie ist, dass im Projekt ELISE die interdisziplinäre Systemmodellierung mit SysML auf die jeweilige Modellierungsaufgabe angepasst werden muss, um eine interdisziplinäre Systemarchitektur aus Hard- und Softwarekomponenten wie gewünscht abbilden zu können.

2.9 Zwischenfazit

Die Integration der drei Säulen in Ansätzen der interdisziplinären Systemmodellierung kann als Differenzierungsmerkmal angesehen werden. Ansätze der interdisziplinären Systemmodellierung, die nur eine der Säulen spezifizieren und die anderen Säulen anerkennen aber losgelöst von der deren Spezifikation betrachten, werden im Folgenden als **nicht integrierte Ansätze** bezeichnet.

Ansätze der interdisziplinären Systemmodellierung, die alle drei Säulen integrativ spezifizieren, werden im Folgenden als **integrierte Ansätze** bezeichnet.

Es existiert eine Reihe an integrierten und nicht integrierten Ansätzen, die nach einer ersten Sichtung individuelle Stärken und Schwächen aufweisen. Jedoch konnte sich bis zum gegenwärtigen Zeitpunkt keiner dieser Ansätze in der Praxis etablieren.

⁹⁷ MUNKER ET AL. 2014a

⁹⁸ ALBERS ET AL. 2013c

Die Relevanz der interdisziplinären Systemmodellierung im Produktentstehungsprozess (PEP) zeigt sich vor allem in dem Potenzial im Rahmen des Einsatzes in der Produktgenerationsentwicklung (PGE).

Eine Reihe an Forschungsvorhaben beschäftigen sich mit Ansätzen zur Systemmodellierung, deren Anwendung, sowie der Einbindung in den PEP. Ein Großteil dieser Vorhaben basiert auf SysML.

Im modernen Verständnis steht der Mensch im Mittelpunkt der Produktentstehung. Die interdisziplinäre Systemmodellierung könnte den Menschen in der interdisziplinären Produktentwicklung unterstützen, indem sie ihm als Werkzeug zur kompetenzübergreifenden Abbildung und Zugänglichmachung von systemrelevantem Wissen dient. Der Mensch nimmt in diesem Falle die Rolle des Anwenders von Lösungen zur interdisziplinären Systemmodellierung ein.

Für Anwender in der Praxis scheint vordergründlich ein Gesamtkonzept, bestehend aus Sprache, Methode und Tool, relevant zu sein. Befragte Anwender, welche bereits Erfahrung in der Modellierung mit SysML besitzen, erkennen das Potenzial der solchen interdisziplinären Systemmodellierung, stehen dem praktischen Nutzen jedoch kritisch gegenüber. Anwendungsstudien im Rahmen dieser Arbeit bestätigen dies exemplarisch.

Durch die Separierung von Sprache, Methode und Werkzeug nehmen die Anwender die Lernkurve als steil wahr. Die OMG leitet daraus die Empfehlung ab, Sprache, Methode und Werkzeug noch strikter getrennt zu vermitteln. Diese Schlussfolgerung der OMG scheint jedoch den Bedürfnissen der Anwender zu widersprechen.⁹⁹

⁹⁹ Vgl. Kapitel 2.7 Gegenwärtige Anwendersicht zur Modellierung mit SysML

3 Motivation und Zielsetzung

In diesem Kapitel wird auf Basis des erweiterten ZHO-Modells die Motivation dieser Arbeit abgeleitet. Daran schließt sich die Zielsetzung an, für deren Bearbeitung geeignete Forschungsfragen formuliert werden.

3.1 Motivation

Die übergeordnete Zielsetzung dieser Arbeit ist, den Ingenieur in den Herausforderungen von morgen zu unterstützen. Die zunehmende Interdisziplinarität der Produkte wirkt sich mit einhergehender Steigerung der Komplexität und Vernetzung¹⁰⁰ auf die Ziel- und Objektsysteme aus. Das Kompetenzprofil der Entwickler¹⁰⁰ wird dabei in seiner Ausprägung einem Kammprofil ähneln. Der horizontale, fachdisziplinübergreifende Ast des Kammprofils würde den individuellen Entwicklern die Chance bieten, ein gemeinsames mentales Modell von Ziel- und Objektsystem zu entwickeln.

Die interdisziplinäre Systemmodellierung könnte als Hilfsmittel eines solchen übergreifenden Verständnisses dienen. Durch abstrakte Modellierung fachdisziplinübergreifender Systemaspekte aus Ziel- und Objektsystem¹⁰¹ würde eine für jeden individuellen Entwickler gleichermaßen verständliche Systembeschreibung entstehen. Wenn dies gelingt, könnten Fachkräfte die interdisziplinäre Systemmodellierung als ein Werkzeug benutzen, von dem sie wissen, dass es in den individuellen mentalen Wissensbasen und Lösungsräumen der beteiligten Fachkräfte gleich interpretiert wird. Dieses könnte nach dem Verständnis des erweiterten ZHO-Modells einen wichtigen Beitrag für die Erschließung von Synergieeffekten und Vermeidung von Missverständnissen leisten. Bild 10 stellt die interdisziplinäre Systemmodellierung im Kontext der kammförmig ausgeprägten Kompetenzen in Wissensbasis und Lösungsraum vereinfacht dar.

¹⁰⁰ Vgl. Kapitel 1 Einleitung

¹⁰¹ Das interdisziplinäre Systemmodell beschreibt Zielvorgaben (z.B. Ziele und Randbedingungen), die auch Systembeschreibungen (beispielsweise die Systemarchitektur) beinhalten können. Nach Albers sind im ZHO-Modell die Informationen (beispielsweise Ziele, Randbedingungen oder auch die Systemarchitektur) Bestandteil des Zielsystems, während deren Artefakte (beispielsweise Dokumente oder Modelle von Zielen und Randbedingungen oder Modell der Systemarchitektur) Bestandteile des Objektsystems darstellen. Im Falle eines interdisziplinären Systemmodells wären alle im Modell beinhalteten Informationen Bestandteil des Zielsystems, während das Modell als solches ein Artefakt des Objektsystems darstellt. Zur vereinfachten Darstellung nach Bild 10 wird hier das interdisziplinäre Systemmodell nebenstehend zu Ziel- und Objektsystem visualisiert.

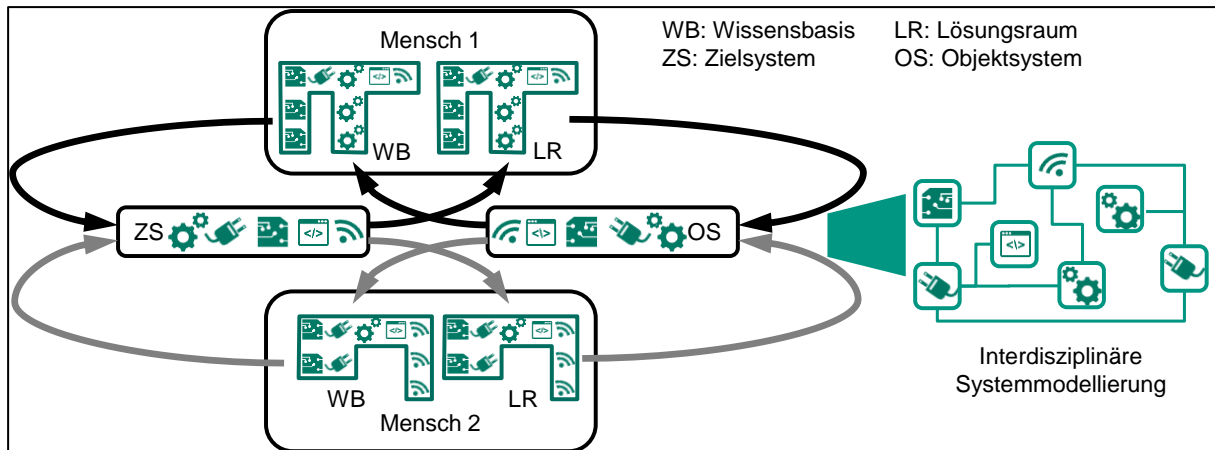


Bild 10: Kammförmige Kompetenzprofile, interdisziplinäre Produkte und interdisziplinäre Systemmodellierung am erweiterten ZHO-Modell.

Die Idee interdisziplinärer Systemmodellierung ist nicht neu, sie wird in bestehenden Modellierungsansätzen und in Forschungsfeldern aufgegriffen.¹⁰² Nach den Grundlagen und des Stands der Forschung, insbesondere nach der gegenwärtigen Anwendersicht,¹⁰³ scheint sich die interdisziplinäre Systemmodellierung in der breiten Anwendung domänenübergreifend noch nicht durchgesetzt zu haben, worauf auch Anwendungsstudien¹⁰⁴ und Rückmeldungen vom Befragten, die bereits mit SysML Modellierung in Berührung gekommen sind¹⁰⁵, hindeuten.

Die dieser Arbeit zugrunde liegende Motivation ist daher, die interdisziplinäre Systemmodellierung für den Anwender nach menschenzentrierten Bedürfnissen nutzbarer zu gestalten. Hierzu wird der Mensch als denkendes und handelndes Zentrum nach den dieser Arbeit zugrunde liegenden zwei zentralen Hypothesen berücksichtigt:

H1: Ein Anwender der interdisziplinären Systemmodellierung ist wie der Mensch als denkendes und handelndes Zentrum der Produktentstehung nach dem erweiterten ZHO-Modell¹⁰⁶ zu verstehen.

H2: Das erweiterte ZHO-Modell¹⁰⁶ bildet die Grundlage zur Erfassung der Bedürfnisse und damit zur Gestaltung eines anwenderorientierten Ansatzes der Systemmodellierung.

¹⁰² Vgl. Kapitel 2.5 Relevante Ansätze zur interdisziplinären Systemmodellierung und Kapitel 2.6 Systemmodellierung in der Produktgenerationsentwicklung

¹⁰³ Vgl. Kapitel 2.7 Gegenwärtige Anwendersicht zur Modellierung mit SysML

¹⁰⁴ Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

¹⁰⁵ Vgl. Kapitel 2.7 Gegenwärtige Anwendersicht zur Modellierung mit SysML

¹⁰⁶ Vgl. Kapitel 2.3 Erweitertes ZHO-Modell

3.2 Zielsetzung

Ziel dieser Arbeit ist, einen integrierten Ansatz zur interdisziplinären Systemmodellierung zu konzipieren. Dieser soll die Stärken der bestehenden relevanten Ansätze aufgreifen und neu kombinieren. Es soll dabei ein niederschwelliger Einstieg in die interdisziplinäre Systemmodellierung ermöglicht werden, ohne Agilität und Flexibilität einzuschränken.

Übergeordnetes Ziel ist, den Ansatz so zu gestalten, dass der Anwender die interdisziplinäre Systemmodellierung nutzen kann und sich dabei statt auf Formalismen der Modellierung, auf die Lösung seiner Entwicklungsaufgabe konzentrieren kann.

Anhand dieser zentralen Aspekte des Mensch-zentrierten erweiterten ZHO-Modells zum Denken und Handeln in der Entwicklungstätigkeit¹⁰⁷ lassen sich konkrete Zielsetzungen für den Ansatz ableiten. Diese werden in Tabelle 2 gegenübergestellt.

Tabelle 2: Zielsetzungen an den Ansatz nach Aspekten des erweiterten ZHO-Modells

Aspekte erweitertes ZHO-Modells	abgeleitete Zielsetzung
Individuelle Denk- und Handlungsvorgänge <ul style="list-style-type: none"> • opportunistisches Vorgehen in der Entwicklung • individuelle Wissensbasis • individueller Lösungsraum 	Individuelle Gestaltbarkeit der Modellierung <ul style="list-style-type: none"> • Schlanke, einfach zu erlernende Basis der interdisziplinären Systemmodellierung zur initialen Modellierung • Erweiterbarkeit und Anpassbarkeit der Systemmodellierung
Epistemische Unsicherheit <ul style="list-style-type: none"> • co-evolutionäre Entwicklung von Zielsystem und Objektsystem 	Agilität <ul style="list-style-type: none"> • Erweiterung und Anpassung muss effektiv und agil vom Anwender selbst bedarfsgerecht durchgeführt werden können
Iterative Entwicklungsprozesse <ul style="list-style-type: none"> • Wechsel von Analyse- und Syntheseschritten 	Agilität in Analyse- und Syntheseschritten <ul style="list-style-type: none"> • Analysemöglichkeiten am Modell • Analysemöglichkeiten müssen effektiv und agil vom Anwender selbst bedarfsgerecht konfiguriert werden können

¹⁰⁷ Vgl. Kapitel 2.3 Erweitertes ZHO-Modell

Abgrenzend zu den Forschungsfeldern die sich mit der Methode zur Modellierung von Systemen beschäftigen¹⁰⁸, wird hier nicht nur eine Methode, sondern ein ganzheitlicher integrierter Ansatz zur Modellierung beliebiger technischer Systeme konzipiert. Dieser wird unter Berücksichtigung, aber unabhängig und lösungsneutral, von bestehenden Ansätzen entworfen und vereint die drei Säulen der Systemmodellierung¹⁰⁹ (Sprache, Methode und Werkzeug) in einer Gesamtlösung.

¹⁰⁸ Vgl. Kapitel 2.6 Systemmodellierung in der Produktgenerationsentwicklung

¹⁰⁹ Vgl. Kapitel 2.4 Die drei Säulen interdisziplinärer Systemmodellierung

3.3 Forschungsfragen

Um den aus den Grundlagen und dem Stand der Forschung hergeleiteten und in der Motivation und Zielsetzung formulierten Forschungsgegenstand zu bearbeiten, werden wie folgt Forschungsfragen formuliert.

Die Forschungsfragen F1 und F2 greifen den Teil der Zielsetzung auf, nach dem bestehende Ansätze zur interdisziplinären Systemmodellierung Berücksichtigung finden sollen.

Der Forschungsfrage F1 liegt dabei die Hypothese zugrunde dass sich in bestehenden Ansätzen Gemeinsamkeiten finden lassen, die auf eine besondere Relevanz für die interdisziplinäre Systemmodellierung hindeuten.

F1: Welche gemeinsame Basis von Sprachelementen, methodischen Einzelschritten und Funktionalitäten des Werkzeugs lässt sich an den bestehenden Ansätzen erkennen?

Der Forschungsfrage F2 liegt die Hypothese zugrunde, dass sich aus der gemeinsamen Basis, sowie aus weiteren Analysen und Beobachtungen ein neuer anwenderorientierter Ansatz zur interdisziplinären Systemmodellierung synthetisieren lässt.

F2: Wie lässt sich aus der gemeinsamen Basis und aus weiteren Randbedingungen ein neuer integrativer Ansatz synthetisieren?

Der Forschungsfrage F3 liegt der Anspruch zugrunde, mit der Erprobung des Ansatzes eine Grundlage für weitere Forschung oder Innovation zu bieten.

F3: Wie gut dient der Ansatz dem Entwickler als Hilfsmittel für das übergreifende Systemverständnis?

Diese Forschungsfragen sind aufeinander aufbauend formuliert. Anhand der Fragen wird das Forschungsvorhaben vom Entwurf bis zur Validierung geleitet. Die Beantwortung aller Fragen stellt somit den wesentlichen wissenschaftlichen Beitrag dieser Arbeit dar. Zur zielgerichteten Bearbeitung der Fragen wird eine Forschungsmethode verwendet. Die Eingliederung der Fragen in diese Forschungsmethode und deren zugrunde liegenden Zielsetzungen werden im nachfolgenden Kapitel 4 detailliert beschrieben.

4 Forschungsdesign

In diesem Kapitel wird die zur Sicherstellung der wissenschaftlichen Qualität angewandte Forschungsmethode dargestellt. Weiterhin wird das Vorgehen für die wesentlichen Teile der Arbeit, der Entwicklung und der Validierung des Ansatzes, dargelegt.

4.1 Forschungsmethode

Zur Beantwortung der Forschungsfragen und zur Sicherstellung der wissenschaftlichen Qualität orientiert sich diese Arbeit an dem Modell einer Forschungsmethode von MARXEN UND ALBERS¹¹⁰, welche von MARXEN¹¹¹ nach Bild 11 weiterentwickelt wurde. MARXEN UND ALBERS vereinen mit diesem Modell einen generischen Ansatz von BLESSING UND CHAKRABARTI¹¹² mit aus empirischen Beobachtungen abgeleiteten Kategorien von CANTAMESSA.¹¹³

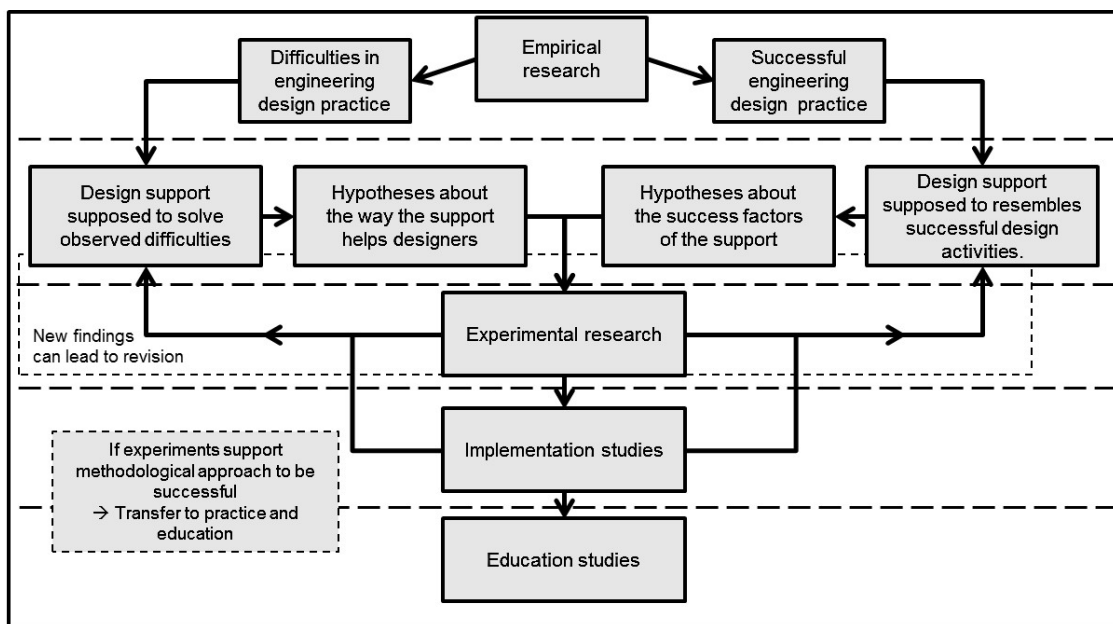


Bild 11: Forschungsmethode nach MARXEN¹¹¹ mit den Phasen I-V

Die in dieser Arbeit dargestellte Forschung umfasst die Phasen I-III der Forschungsmethode.

¹¹⁰ MARXEN & ALBERS 2012

¹¹¹ MARXEN 2014

¹¹² BLESSING & CHAKRABARTI 2009

¹¹³ CANTAMESSA 2003

Das Modell beschreibt eine Forschungsmethode für Themen, die in der Produktentwicklung unterstützen. Phase I beinhaltet empirische Forschung zur Ermittlung von Forschungslücken. Im linken Ast dieser Phase werden durch Beobachtung Defizite aktueller Praktiken und damit Unterstützungsbedarf ermittelt. Im rechten Ast werden durch Beobachtungen erfolgreiche Praktiken identifiziert und daraus Unterstützung abgeleitet. Diese Phase ist Bestandteil des Kapitels 2 „Grundlagen und Stand der Forschung“ dieser Arbeit. Das im Rahmen dieser Forschungsaktivitäten angewandte Vorgehen stellt eine Kombination aus den beiden Ästen dar, indem bestehende Ansätze mit ihren Schwächen (linker Ast) und Stärken (rechter Ast) untersucht werden. Wesentliche Erkenntnis aus Phase I ist die Notwendigkeit eines anwenderorientierten Ansatzes der interdisziplinären Systemmodellierung zur Erschließung des Potenzials in der Praxis mit Vereinigung der Stärken bestehender Ansätze.

Wesentliches Ergebnis der Phase I sind die Motivation, Zielsetzung und Forschungsfragen aus Kapitel 3 „Motivation und Zielsetzung“.

Phase II beinhaltet die Ableitung eines Unterstützungskonzepts aus den Erkenntnissen von Phase I. Das entsprechende Vorgehen zur Erstellung eines initialen Zielsystems für den Ansatz wird in Kapitel 7 „Initiales Zielsystem des Ansatzes“ beschrieben. Wesentlicher Bestandteil dieses Prozesses sind die Analyse und Vergleich der Charakteristika der relevanten Ansätze hinsichtlich Sprache, Methode und Werkzeug aus Kapitel 5 „Analyse der relevanten Ansätze“, sowie die Ermittlung von Randbedingungen zur Systemmodellierung aus den Anwendungsstudien der Modellierung mit SysML in Kapitel 6 „Randbedingungen zur interdisziplinären Systemmodellierung“. Das initiale Zielsystem des Ansatzes wird dann in Kapitel 7 „Initiales Zielsystem des Ansatzes“ dargestellt.

Wesentliches Ergebnis der Phase II ist das initiale Zielsystem des neuen Ansatzes aus Kapitel 7 „Initiales Zielsystem des Ansatzes“ und damit die Beantwortung der Forschungsfragen F1 und F2.

Phase III beinhaltet die Erstellung eines Prototypen namens „SystemSketcher“ in Form eines Software-Werkzeugs für die experimentelle Erforschung mittels einer Laborstudie unter Laborbedingungen. Das entsprechende Vorgehen zur experimentellen Validierung wird im Kapitel 9 „Validierung des Ansatzes“ dargestellt. Wesentlicher Bestandteil dieses Prozesses ist das in Kapitel 8 „Validierungsprototyp – SystemSketcher“ vorgestellte Prototyp, sowie die Validierung nach Kapitel 9 „Validierung des Ansatzes“.

Wesentliches Ergebnis der Phase III sind die Erkenntnisse aus der Laborstudie und damit die Beantwortung der Forschungsfrage F3.

Phase IV beinhaltet experimentelle Erforschung mittels Studien zur Validierung im realen Einsatz in Feldstudien. Die in dieser Arbeit dargestellte Forschung umfasst die Phasen I-III. Die Validierung des Ansatzes wird durch Studien zur grundsätzlichen Tragfähigkeit im Rahmen der Phase II abgebildet.

Phase V beinhaltet den Transfer der Forschungsergebnisse in die Lehre. Sie ist im Rahmen der Arbeit nicht betrachtet worden. Ansatzpunkte zu einer möglichen Phase V werden im Rahmen von Kapitel 10 „Zusammenfassung und Ausblick“ dargestellt.

4.2 Vorgehen zur Entwicklung des Ansatzes

Die Entwicklung des neuen Ansatzes basiert auf einem gemischt analytisch und empirischem Vorgehen. Dieses Vorgehen und das Ergebnis wurden im Rahmen eines Konferenzbeitrages auf dem Tag des Systems Engineering (TdSE) 2015 der deutschen Systems Engineering Fachcommunity, Gesellschaft für Systems Engineering (GfSE) vorgestellt.¹¹⁴ Bild 12 fasst das vorgestellte Vorgehen zusammen. Im Rahmen dieser Arbeit wird auf die Einzelschritte des Vorgehens im Detail wie folgt beschrieben eingegangen werden.

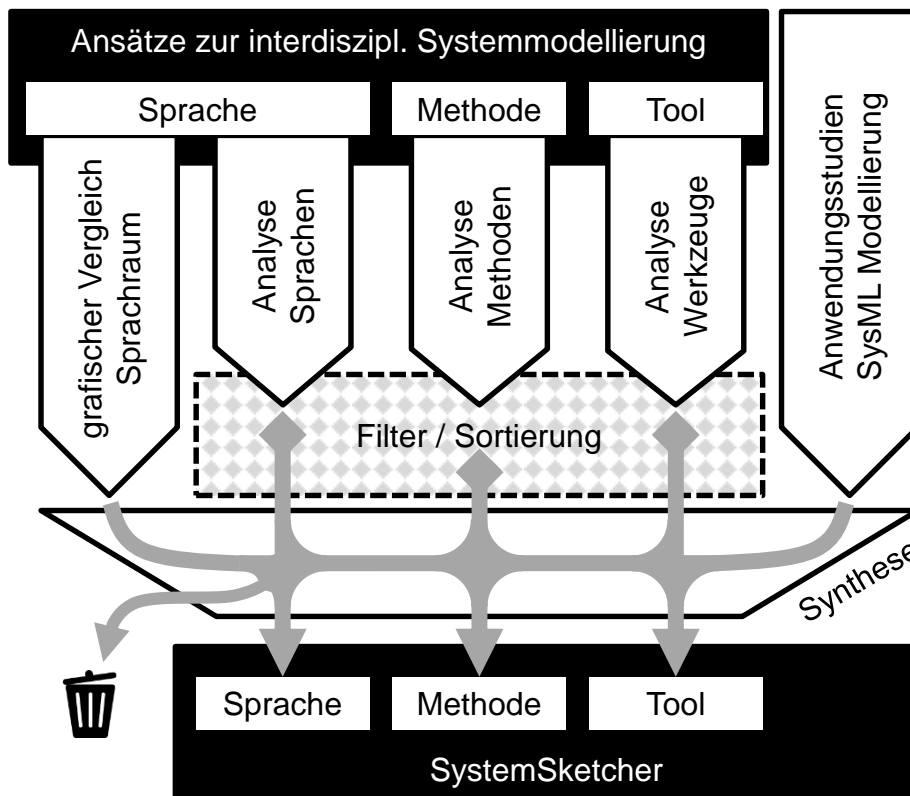


Bild 12: Vorgehen zur Entwicklung des Ansatzes. Beschriftung modifiziert in Anlehnung an Munker & Albers 2015¹¹⁴

¹¹⁴ MUNKER & ALBERS 2015

Den **analytisch geprägten Teil** in Kapitel 5 stellt die Analyse der als relevant identifizierten Ansätze¹¹⁵ aus Kapitel 2.5 hinsichtlich ihrer Ausprägung von Sprache, Methode und Modellierungswerkzeug dar.¹¹⁶ Ziel der Analyse ist, eine gemeinsame ansatzübergreifende Basis der interdisziplinären Systemmodellierung zu finden. Deren Inhalte werden als relevant interpretiert und fließen in das initiale Zielsystem des Ansatzes ein. Neben einer tabellarischen Analyse wird in Kapitel 5, um der netzartigen Ausprägung der Sprachräume gerecht zu werden, eine grafische Analyse vorgenommen. Anhand eines grafischen Plots werden die Sprachräume visualisiert und daran Muster beobachtet.

Im **empirisch geprägten Teil** in Kapitel 6 fließen Erkenntnisse aus den Anwendungsstudien, der Modellierung des Telematik-Datensystems, sowie des Hochvoltspeichers, ein.¹¹⁷ Daraus werden unmittelbar erforderliche sowie ergänzende Randbedingungen zur interdisziplinären Systemmodellierung abgeleitet.

Auf dieser Basis wird das **initiale Zielsystem des Ansatzes** nach Kapitel 7 gestaltet.¹¹⁸ Dabei werden die Ergebnisse der Analysen aufgegriffen und integriert. Der Ansatz stellt somit eine Gesamtlösung aus einer Zusammenstellung von Aspekten bestehender Ansätze und Erkenntnissen aus den Anwendungsstudien dar.

4.3 Vorgehen zur Validierung des Ansatzes

Die Validierung in Kapitel 9 stellt einen wesentlichen Bestandteil dieser Arbeit dar.¹¹⁹ Mit der Validierung soll die grundsätzliche Tragfähigkeit des Ansatzes nachgewiesen werden. Hierzu sind die wesentlichen Aspekte des initialen Zielsystems in einem ersten Prototypen namens „SystemSketcher“ in Form eines Software-Werkzeugs implementiert worden.¹²⁰ Für die Validierung ist ein Studiendesign gewählt worden, bei dem Probanden ein technisches System begleitet durch den Studienleiter mit Hilfe eines Validierungsprototypen modellieren. Der Nachweis der Tragfähigkeit erfolgt auf Grundlage von Bewertungen des Ansatzes und Rückmeldungen der Probanden. Eine Bewertung und Einordnung der allgemeinen Gebrauchstauglichkeit erfolgt nach der etablierten Methode der „Systems Usability Scale“ (SUS),¹²¹ sowie mit einem Fragenkatalog anhand des erweiterten ZHO Modells.¹²²

¹¹⁵ Vgl. Kapitel 2.5 Relevante Ansätze zur interdisziplinären Systemmodellierung

¹¹⁶ Vgl. Kapitel 5 Analyse der relevanten Ansätze

¹¹⁷ Vgl. Kapitel 6 Randbedingungen zur interdisziplinären System-modellierung

¹¹⁸ Vgl. Kapitel 7 Initiales Zielsystem des Ansatzes

¹¹⁹ Vgl. Kapitel 9 Validierung

¹²⁰ Vgl. Kapitel 8 Validierungsprototyp – SystemSketcher

¹²¹ BANGOR ET AL. 2009

¹²² Vgl. Kapitel 2.3 Erweitertes ZHO-Modell

5 Analyse der relevanten Ansätze

In diesem Kapitel wird die vom Autor und ALBERS auf dem TdSE 2015¹²³ vorgestellte Analyse aufgegriffen und inhaltlich vertieft dargestellt. Tabelle 3 gibt eine Übersicht der in Kapitel 2.3 erfassten und umrissenen Ansätze mit ihrer Aufgliederung nach Sprache, Methode und Werkzeug.

Tabelle 3: Untersuchte Ansätze mit den Aspekten Sprache-Methode-Werkzeug. Beschriftung modifiziert in Anlehnung an Munker & Albers 2015¹²³

Ansatz	Sprache	Methode	Werkzeug	Integrierter Ansatz
interdisz. Systemmodellierung mit SysML	SysML, SysML LITE	OOSEM, SYSMOD, SysKIT	SysML fähige Tools AS, CSM, EA, etc.	NEIN
IBM Telelogic Harmony SE	SysML subset	Harmony-SE	IBM Telelogic	NEIN
IBM RUP SE	UML / SysML	RUP-SE	IBM Rational Suite	NEIN
VITEC MBSE	VITEC SDL	VITEC MBSE Onion Model	VITEC CORE Suite	JA
JPL SA	JPL SA	JPL SA	State Database RDBMS	JA
Dori OPM	OPD and OPL	OPM system process	OPCAT	JA
CONSENS	CONSENS	CONSENS „Anwendungsszenarien“	Mechatronic Modeller	NEIN
Dassault RFLP	Dassault RFLP	RFLP	Catia V6	JA

5.1 Analyse der Sprachen

Grundlage dieser Analyse stellen die Sprachen der relevanten Ansätze dar. Ziel dieser Analyse ist, thematisch verwandte Sprachelemente ansatzübergreifend zu identifizieren, die einen Hinweis auf mögliche relevante Sprachelemente zur interdisziplinären Systemmodellierung geben.

Nicht explizit berücksichtigt worden ist dabei der Ansatz IBM RUP SE, da dessen Modellierungssprachen SysML und UML bereits in dem Ansatz „interdisziplinäre Systemmodellierung mit SysML“ enthalten und die Sprache UML als domänenspezifische Sprache für die Modellierung von vorrangig Software Systemen nicht Bestandteil dieser Sprachenanalyse ist.

¹²³ MUNKER & ALBERS 2015

Aus den in Kapitel 2.3 beschriebenen verfügbaren Quellen, wie Spezifikationen oder Handbüchern, sind die Sprachelemente differenziert nach Entitäten (z. B. Systemelement oder Anforderung) und Relationen (z. B. Zuordnungsbeziehungen) extrahiert worden.

In Summe sind aus 8 Sprachen 208 Entitäten¹²⁴ und 525 Relationen erfasst worden. Diese Entitäten und Relationen sind nach thematisch verwandten Elementen in 8 Gruppierungen gegliedert worden.

Durch Abstraktion einer Bedeutung zu jedem Sprachelement sind ansatzübergreifend die Elemente gemeinsamer Bedeutung, aber unterschiedlicher Bezeichnung identifiziert worden. Im Falle dass keine Bedeutung abstrahiert werden konnte, ist die originale Bezeichnung des Sprachelements für die Bedeutung übernommen worden.

Die Abstraktion der Bedeutung ist unter Berücksichtigung der vorliegenden Materialien und Spezifikationen der Ansätze¹²⁵ nach Interpretation des Autors erfolgt. Die Abstraktion wird daher weder dem Anspruch der Objektivität, noch der Vollständigkeit unterliegen können.

Jedoch besteht der Anspruch an der Nachvollziehbarkeit der wissenschaftlichen Arbeiten. Der Entwurfsprozess wird im Folgenden transparent mit für das Verständnis relevanten exemplarischen Ausschnitten aus der Sprachenanalyse dargestellt. Die erfassten Sprachelemente und deren abstrahierte Bedeutungen finden sich als Referenz im Anhang dieser Arbeit wieder.

Mit diesem Vorgehen können nun ansatzübergreifend mehrfach vorkommende Sprachelemente unterschiedlicher Bezeichnung aber gleicher Bedeutung identifiziert werden. Dazu sind zunächst sind diejenigen Sprachelemente ausgeleitet worden, die im Untersuchungsraum mit ihrer Bedeutung mindestens doppelt vorkommen, im Folgenden als „Vorkommenshäufigkeit von ≥ 2 “ bezeichnet. Das mehrfache Vorkommen innerhalb eines einzelnen Sprachraumes eines einzelnen Ansatzes ist dabei nicht gezählt worden, so dass nur ansatzübergreifend mehrfach vorkommende Sprachelemente identifiziert wurden.

Gruppierungsübergreifende Relationen sind dabei in diejenigen Gruppierungen einsortiert worden, aus denen sie entspringen. Mehrfach vorkommende Sprachelemente, die sich mehreren Gruppierungen zuordnen lassen, sind während

¹²⁴ Aktualisierte Auflistung von 208 Entitäten in Vergleich zur Veröffentlichung des TdSE mit 210 (ohne Auswirkung auf die weitere Berechnung zur Vorkommenshäufigkeit)

¹²⁵ Vgl. Kapitel 2.5 Relevante Ansätze zur interdisziplinären Systemmodellierung

der Ausleitung als „übergreifend zuordenbar“ gruppiert worden, dies betrifft im Ergebnis 4 Relationen.

Nach diesem Vorgehen sind in Summe 23 ansatzübergreifende Entitäten und 18 ansatzübergreifende Relationen identifiziert worden. Diese werden in Tabelle 4 gelistet.

Diese bilden eine gemeinsame Basis von Sprachelementen und damit einen Hinweis auf mögliche relevante Sprachelemente zur interdisziplinären Systemmodellierung.

Tabelle 4: Anzahl der Entitäten und Relationen nach Gruppierungen und ansatzübergreifend mehrfach vorkommende mit Vorkommenshäufigkeit ≥ 2

Gruppierung	Entitäten	Relationen	≥ 2	≥ 2
			Entitäten	Relationen
Anwendungsfälle	9	17	2	0
Anforderungen	16	58	1	4
Validierung	8	48	1	1
Verhalten / funktionale Architektur	105	72	10	5
Systemstruktur / logische Architektur	43	52	4	2
Parametrische Modellierung	8	3	2	1
Zusatzinformationen	9	72	2	1
Modell Infrastruktur	10	203	1	0
übergreifend zuordenbar	X	X	0	4
Summe	208	525	23	18

Tabelle 5 schlüsselt, gegliedert nach den gefundenen Gruppierungen, die Entitäten mit Vorkommenshäufigkeit ≥ 2 über ihr Vorkommen in den Ansätzen auf. Die originale Bezeichnung der Entitäten variiert dabei ansatzspezifisch bei gleicher Bedeutung. Die originalen Bezeichnungen sind für die Abbildung des ansatzübergreifenden Vorkommens nicht weiter relevant. Sie werden aber bei der Analyse nach Redundanz und Konsistenz der Sprachräume¹²⁶ aufgegriffen.

¹²⁶ Vgl. Kapitel 5.2 Redundanz und Konsistenz in den Sprachräumen

Bei den mehrfach vorkommenden Relationen ist während der Ausleitung zwischen gruppierungsübergreifenden Relationen nach Tabelle 7, und Relationen innerhalb von Gruppierungen nach Tabelle 6 differenziert worden, um sprachraumübergreifende Relationen zu finden und besser darstellen zu können. Die gruppierungsübergreifenden Relationen bilden einen Hinweis auf besonders relevante Relationen, da sie den Sprachraum überspannen und dadurch systemübergreifende Beziehungen abbilden können. Sie sind zur Darstellung in diejenigen Gruppierungen einsortiert worden, aus denen sie entspringen.

Tabelle 6: Auf Gruppierungen begrenzte, ansatzübergreifend mehrfach, mindestens doppelt, vorkommende Relationen

Auf Gruppierung begrenzte, Ansatzübergreifend mehrfach vorkommende Relationen mit gleicher Bedeutung	SysML	SysML Lite	IBM Telelogic Harmony SE	Vitech SDL	JPL State Analysis	Dori OPM	RFLP in Catia V6	CONSENS
Anwendungsfälle								
Anforderungen								
Ableitung	X		X					
Trace	X		X	X				
Verfeinerung	X			X				X
Validierung								
Verhalten / funktionale Architektur								
Verbindung von Verhaltenselementen	X	X	X				X	
Zusammenführung	X		X	X				
Transition	X		X		X			
Message	X		X					
Systemstruktur / logische Architektur								
Verbindung von Schnittstellen	X	X	X				X	
Parametrische Modellierung								
Zusatzinformationen								
Relation zu weiterführender Beschreibung	X			X				
Modell Infrastruktur								
übergreifend zuordenbar								
Subsumtion	X	X	X	X		X		X
Vererbung	X		X			X		X
Referenz	X		X			X		

Tabelle 7: Die Gruppierungen überspannende, ansatzübergreifend mehrfach, mindestens doppelt, vorkommende Relationen. Zuordnung nach Ursprungs-Gruppierung

Die Gruppierungen überspannende, Ansatzübergreifend mehrfach vorkommende Relationen mit gleicher Bedeutung Zuordnung nach Ursprungs-Gruppierung	SysML	SysML Lite	IBM Telelogic Harmony SE	Vitech SDL	JPL State Analysis	Dori OPM	RFLP in Catia V6	CONSENS
Anwendungsfälle								
Anforderungen								
Erfüllung	X	X	X	X			X	
Validierung								
Verifizierung	X		X	X				
Verhalten								
Relation zu Zuständen				X		X		
Systemstruktur / logische Architektur								
Relation zw. Funktion und Systemarchitektur	X	X		X			X	
Parametrische Modellierung								
Relation zu parametrischen Elementen	X	X	X					
Zusatzinformationen								
Modell Infrastruktur								
übergreifend zuordenbar								
allg Zusammenhänge / Relationen	X		X					

5.2 Redundanz und Konsistenz in den Sprachräumen

Werden die übergreifend vorkommenden Entitäten und Relationen entlang der Bedeutung einzeln nach ihren originären Bezeichnungen aufgeschlüsselt, lassen sich Redundanzen in den Sprachräumen erkennen. Unter „Redundanz“ wird hier redundante Verwendung im Sinne von mehreren, für die gleiche Bedeutung verwendeten, Sprachelemente innerhalb eines Sprachraums verstanden.

Dies wird hier beispielhaft anhand der Subsumtion nach Tabelle 8 dargestellt. Die Subsumtion (deutsch „Unterordnung“) wird hier als Relation zur Strukturmodellierung bzw. Dekomposition verstanden. Es ist beispielsweise bei SysML ersichtlich, dass mit den Relationen „containment“, „«extend»“, „composition“ und „aggregation“ eine 4-fach Redundanz im Sprachraum zur Subsumtion besteht. Redundanzfreiheit würde bedeuten, dass für die Subsumtion eine einzige Relation im Sprachraum zur Verfügung stehen würde.

Tabelle 8: Redundante, mehrfache Verwendung der Relationen mit Bedeutung „Subsumtion“ innerhalb der einzelnen Sprachräume

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Subsumtion	X	X	X	X		X		X
originale Bezeichnung	containment «extend» composition aggregation	containment composition aggregation	containment «extend» composition	extends decomposes includes built in		Aggregation		ist Teil von logische Gruppe
Redundanz	4x	3x	3x	4x		-		2x

Ergebnisse zur Konsistenz werden weiterhin am Beispiel des SysML Sprachraums gezeigt. Unter „Konsistenz“ wird hier die konsistente Verwendung von Sprachelementen im Sprachraum innerhalb eines Ansatzes verstanden.

Werden die Entitäten und Relationen entlang ihrer Bedeutung mit ihren originären Bezeichnungen einzeln nach Gruppierungen aufgeschlüsselt, lassen sich neben Redundanzen auch Inkonsistenzen in den Sprachräumen erkennen. Hier wird beispielhaft die Betrachtung der Subsumtionsrelationen beibehalten, aber auf SysML eingeschränkt werden. Die 4-fach redundanten Relationen werden nicht konsistent, sondern nach Tabelle 9 in den Gruppierungen differenziert angewandt. Es ist ersichtlich, dass in einigen Gruppierungen nach Spezifikation keine Subsumtionsrelationen vorgesehen sind, während in anderen eine redundante Verwendung stattfindet. Eine konsistente Ausprägung würde bedeuten, dass alle für die Subsumtion vorgesehenen Relationen gleichermaßen in allen Gruppierungen des Sprachraums Anwendung finden würden.

Tabelle 9: Inkonsistente Verwendung der Subsumtionsrelationen im SysML Sprachraum

SysML Sprachraum	Anwendungsfälle	Anforderungen	Validierung	Verhalten	Systemstruktur / logische Architektur	Parametrische Modellierung	Zusatzinformationen	Modell/Infrastruktur	übergreifend zuordbar
Subsumtion	X	X		X	X			X	
originale SysML Bezeichnung	«extend»	containment		composition	composition aggregation containment			containment	

Es wird die Hypothese aufgestellt, dass Redundanzen und Inkonsistenzen im Sprachraum den Umgang mit der Modellierung für den Anwender erschweren. Die Erkenntnisse aus der Betrachtung von Redundanz und Konsistenz lassen sich wie folgt zusammenfassen:

Ein anwenderorientierter Ansatz zur interdisziplinären Systemmodellierung sollte in seinem Sprachraum redundanzfrei und konsistent sein.

5.3 Grafischer Vergleich der Sprachräume

Die Sprachelemente (Entitäten und Relationen) sind in Form von Netzgraphen grafisch dargestellt worden. Entitäten sind als Knoten, und Relationen als Kanten repräsentiert. Die Entitäten sind dabei nach denen in der tabellarischen Analyse gefunden Gruppierungen¹²⁷ gruppiert worden. Die Fläche der Gruppierungen korreliert mit der Anzahl der beinhalteten Elemente. Für die Darstellung und zur Ableitung von Mustern wurden jedoch alle Sprachelemente innerhalb von Gruppierungen optisch ausgeblendet. Dadurch konnten anhand der Fläche der Gruppierungen und der Ausprägung der gruppierungsübergreifenden Relationen Charakteristika und Muster in den Sprachräumen beobachtet werden. Die Positionierung der Gruppierungen erfolgt für alle untersuchten Sprachräume nach identischem Schema. Als besonders charakteristische Grafen werden hier die Sprachräume von SysML, VITEC und RFLP in Bild 13 dargestellt, da sie in ihrer

¹²⁷ Vgl. Kapitel 5.1 Analyse der Sprachen

Charakteristik die für die Nachvollziehbarkeit der folgenden Beobachtungen wesentliche Aspekte abbilden. Eine vollständige Abbildung aller analysierten Sprachräume kann dem Anhang dieser Arbeit entnommen werden.

Es ist ersichtlich, dass sich diese Sprachräume in ihrer Übersichtlichkeit und Erfassbarkeit stark unterscheiden. Bei weiterer Betrachtung ist diese „Übersichtlichkeit“ auf die Aspekte Umfang und Systematik eingegrenzt worden.

Unter „**Umfang**“ wird hier die visuell wahrnehmbare Menge an Entitäten und Relationen verstanden, wobei die Anzahl der Entitäten durch die Fläche der Gruppierungen repräsentiert wird. Es ist ersichtlich, dass SysML und VITEC einen umfangreichen Sprachraum im Vergleich zum schlankeren RFLP besitzen. Einige der in RFLP nicht mit Elementen besetzten Gruppierungen („Zusatzinformationen“, „Modell Infrastruktur“ und „Parametrische Modellierung“) sind in diesem Ansatz als Mechanismus des Werkzeuges umgesetzt, während andere („Anwendungsfälle“ und „Validierung“) in diesem Ansatz fehlen.

Unter „**Systematisch**“ werden hier Muster verstanden, die beim Aufbau des Sprachraums zu beobachten sind. Diese lassen sich beispielsweise am Vergleich von SysML zu CONSENS darstellen. Relationen in SysML lassen sich in formalisierte, deren Bedeutung genau spezifiziert ist, und generalisierte, deren Bedeutung sich aus der Verwendung im Kontext ergibt, klassifizieren. Durch Visualisierung von formalisierten und generischen Relationen wird deren gemischte Verwendung im SysML Sprachraum ersichtlich. Weiterhin kann nach solchen mit spezifischer Ausprägung, und solchen mit systematischer Ausprägung klassifiziert werden.

Spezifische Relationen dürfen nur zwischen spezifischen Entitäten verwendet werden. Systematische Relationen dürfen zwischen einer Entität und allen, oder zwischen allen Entitäten beliebig verwendet werden.

Bild 14 stellt diese Beobachtungen zusammenfassend dar. Dort ist die systematische Ausprägung des Sprachraums von CONSENS mit wenigen formalisierten, aber systematisch zur Gruppierung der Systemstruktur hin angeordneten Relationen im Vergleich zur SysML ersichtlich. Der Sprachraum von CONSENS wirkt in dieser Darstellung übersichtlicher, als die gemischt formalisiert-generalisierte und spezifisch-systematische Ausprägung in SysML.

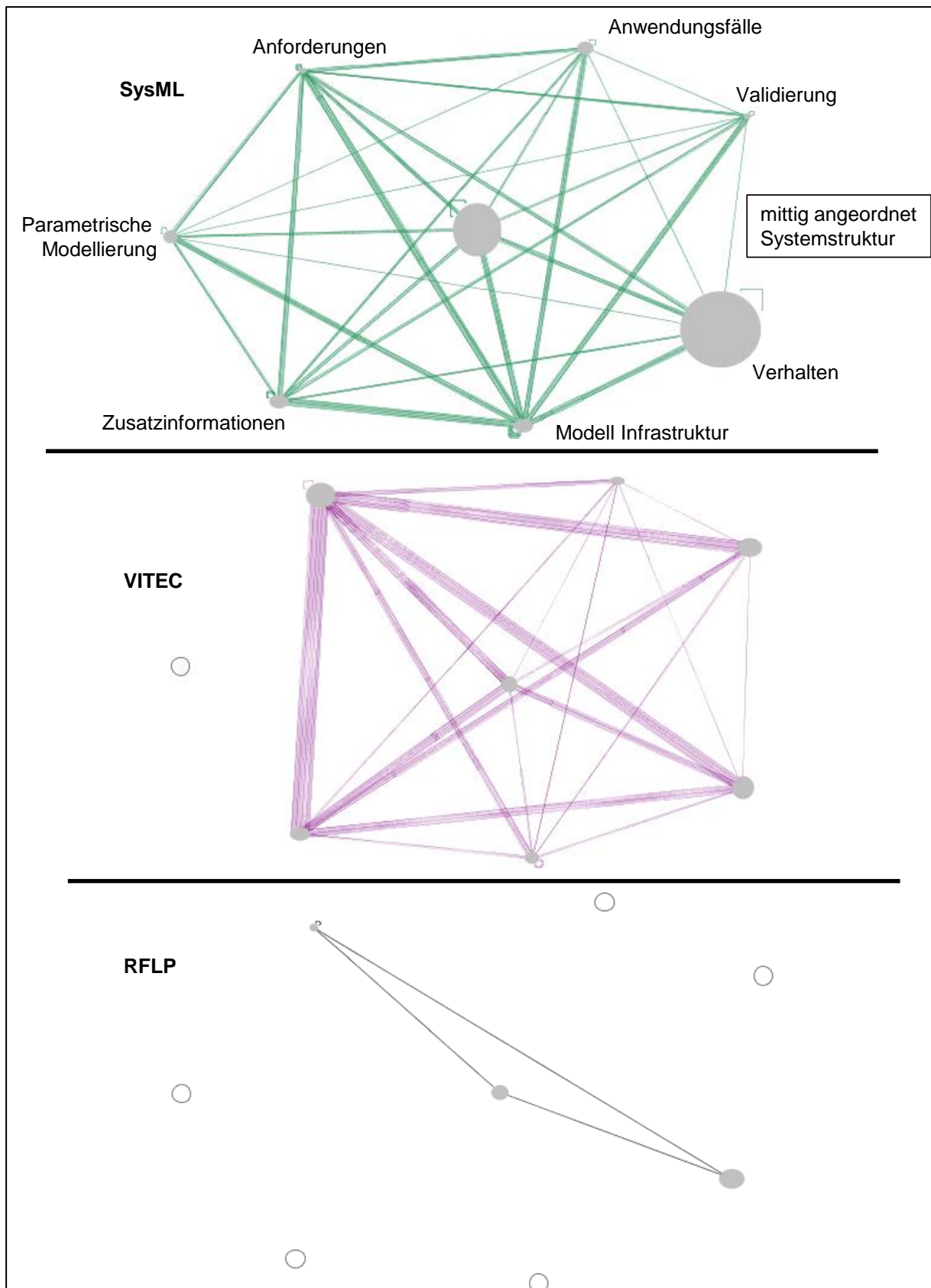


Bild 13: Sprachräume SysML, VITEC und RFLP im grafischen Vergleich

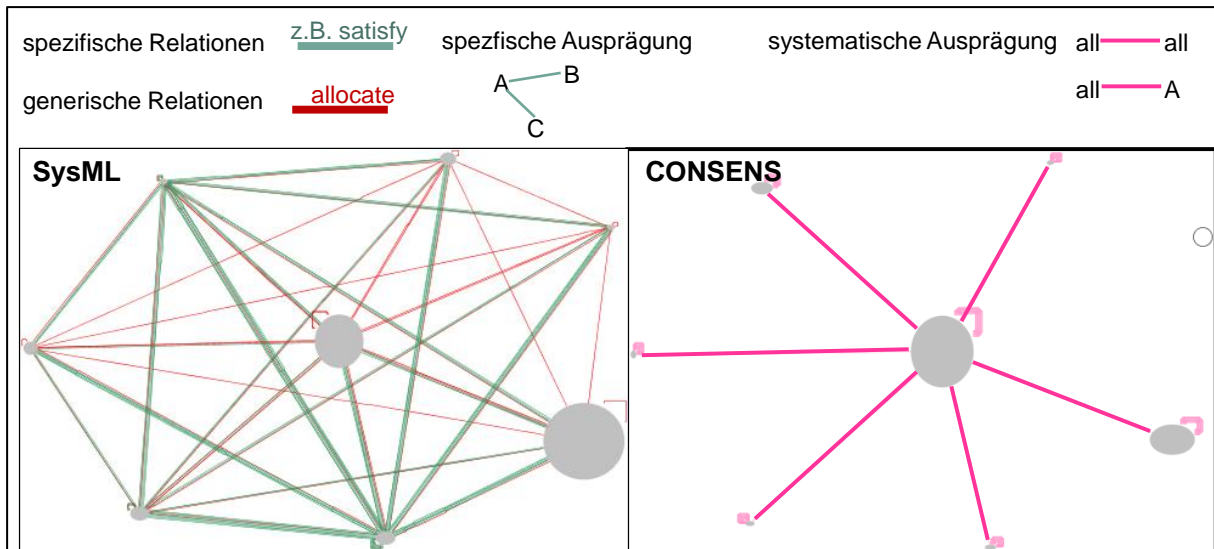


Bild 14: Systematischer Aufbau der Sprachräume von SysML und CONSENS im grafischen Vergleich

Es wird die Hypothese aufgestellt, dass die Erfassbarkeit und Zugänglichkeit eines Modellierungsansatzes für den Anwender mit der Übersichtlichkeit des Sprachraumes dieses Ansatzes korreliert. Die Erkenntnisse aus dem grafischen Vergleich der Sprachräume lassen sich damit wie folgt zusammenfassen:

Für einen anwenderorientierten Ansatz zur interdisziplinären Systemmodellierung sollten folgende Aspekte hinterfragt werden:

- **der Umfang des Sprachraums,**
- **welche Gruppierungen als Mechanismus des Werkzeugs umgesetzt und damit aus dem Sprachraum ausgegliedert werden können,**
- **welche Gruppierungen im Sprachraum zwingend notwendig sind,**
- **die gemischt formalisiert-generalisierte Verwendung von Relationen zugunsten der nicht gemischten,**
- **die spezifische Ausprägung von Relationen zugunsten der systematischen.**

5.4 Analyse der Methoden

Aus den Methoden der Ansätze sind deren methodische Einzelschritte extrahiert worden. Es sind 12 methodische Einzelschritte identifiziert worden, die auf 6 Gruppierungen verteilt werden konnten. Die Gruppierungen der methodischen Schritte orientieren sich an denen der Sprachelemente. Damit lassen sich in der Synthese die Sprachelemente den einzelnen methodischen Schritten zuordnen. Auf eine Reduzierung nach ansatzübergreifend gemeinsam vorkommenden Schritten ähnlich dem Vorgehen zur Analyse der Sprachräume ist hier verzichtet worden, da in Summe 12 Einzelschritte identifiziert werden konnten, die sich ohne Informationsverdichtung tabellarisch übersichtlich darstellen lassen. Es ist auch erfasst worden, ob die Methode streng sequenziell, iterativ oder flexibel abläuft und ob eine modellübergreifende Vernetzung vorgesehen ist. Ein streng sequenzielles Vorgehen ist mit dem Prozessverständnis der Karlsruher Schule der Produktentwicklung nicht vereinbar.¹²⁸ Unter iterativ wird hier ein Vorgehen verstanden, bei dem nach Iterationen innerhalb einer Gruppierung in die nächste gesprungen wird. Auch Rücksprünge sind möglich. Beim flexiblen Vorgehen geschieht die Bearbeitung der methodischen Schritte zeitlich unabhängig und parallelisiert. Tabelle 10 schlüsselt, gegliedert in die Gruppierungen, die methodischen Einzelschritte in ihrem Vorkommen über die Ansätze auf.

Diese bilden eine Basis von methodischen Einzelschritten und damit einen Hinweis auf mögliche relevante methodische Schritte zur interdisziplinären Systemmodellierung.

5.1 Analyse der Werkzeuge

Aus den Beobachtungen der Anwendungsstudien im Rahmen dieser Arbeit¹²⁹ kann abgeleitet werden, dass durch eine Auflistung und Vergleich von einzelnen Funktionalitäten der Werkzeuge keine zielführenden Aussagen zur Werkzeugperformance im praktischen Einsatz getätigt werden kann. Es wird hier deshalb auf eine tabellarische Ableitung analog der Sprachen- und Methodenanalyse verzichtet.

¹²⁸ ALBERS 2010

¹²⁹ Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

Tabelle 10: Ansatzübergreifende Verwendung methodischer Einzelschritte

Methodische Einzelschritte und ihre Verwendung in den Ansätzen.	OOSEM with SysML	IBM Telelogic Harmony SE	IBM RUP SE	Vitech	JPL State Analysis	Dori OPM	RFLP in Catia V6	CONSENS	Sysmod incl. FAS	SysKIT
Systemkontext										
modelliere Systemkontext	X							X	X	X
Anwendungsfälle										
modelliere Akteure/Stakeholder	X	X							X	X
modelliere Anwendungsfälle	X	X	X						X	X
Anforderungen										
modelliere Ziele									X	
modelliere Anforderungen	X	X	X	X	X		X	X	X	X
Verhalten										
modelliere Verhalten / Systemprozesse	X	X		X		X	*	X	X	X
modelliere Zustände	X	X			X	X			X	X
modelliere Systemfunktionen (funktionale Struktur)	X	X			X	X	X	X	X	X
Systemstruktur										
modelliere Fachwissen									X	
modelliere Konzeptstruktur	X								X	
modelliere Systemstruktur (physische Struktur)	X	X		X	X	X	X	X	X	X
Validierung										
modelliere Testfälle	X	**		X						X
Modellübergreifend										
vernetze übergreifend (traceability)	X	X		X			X	X		X
Vorgehen										
modelliere streng sequentiell		X				X				
modelliere iterativ	X			X			X			
modelliere flexibel			X					X	X	X
* Verhalten / Systemprozesse werden in RFLP Catia V6 über Dymola abgebildet										
** Integration von Rational Rhapsody TestConductor										

Der Beobachtungsraum der Werkzeuge umfasst diejenigen, mit denen im Rahmen der Anwendungsstudien gearbeitet werden konnte. Diese sind im Einzelnen die in den Anwendungsstudien eingesetzten SysML Werkzeuge „Cameo Systems Modeler“ (CSM)¹³⁰ und „Atego Artisan Studio“ (AS), (nun unter dem Label „PTC Integrity Modeler“),¹³¹ und der im Rahmen einer eintägigen Anwenderschulung studierte

¹³⁰ No MAGIC 2012

¹³¹ PTC 2014

RFLP Ansatz in Catia V6.¹³² Kernpunkte des VITEC CORE Werkzeugs sind anhand der ausführlichen Produktdokumentation¹³³ untersucht worden.

Die Analyse der Werkzeuge ist nach den folgenden drei Aspekte gegliedert:

- Ausrichtung auf die interdisziplinäre Systemmodellierung

Bei RFLP und CORE ist das Werkzeug auf die interdisziplinäre Systemmodellierung ausgerichtet. Bei AS wird die interdisziplinäre Systemmodellierung mit SysML durch Erweiterung des Werkzeuges gelöst. Die ursprüngliche Ausrichtung auf die Modellierung von Software-Systemen mit UML bleibt dabei für den Anwender dominant. Einen Mittelweg stellt CSM als spezifisches Werkzeug für die interdisziplinäre Systemmodellierung mit SysML dar. Dennoch basiert auch dieses auf einem UML-Modellierungswerkzeug namens „Magic Draw“.¹³⁴ Die UML-spezifische Modellierung ist dabei für den Anwender weitestgehend ausgeblendet, die Mechanismen, Strukturierung und Bedienung nach dem Konzept der UML Modellierung sind für den Anwender jedoch präsent.

Bei der Implementierung interdisziplinärer Systemmodellierung als Erweiterung (add-on) eines bestehenden Modellierungswerkzeuges spürt der Anwender in der Handhabung, dass das Werkzeug ursprünglich nicht für diesen Einsatzzweck konzipiert wurde. Eine add-on Umsetzung auf bestehenden Werkzeugen ist bezüglich der Anwenderorientierung zu hinterfragen.

- Potenzial durch Auswertemöglichkeiten

RFLP weist eine Kopplung an die hauseigene Lösung des Werkzeuganbieters zur numerischen Simulation (Dymola)¹³⁵ auf, die für Berechnungen mit dem interdisziplinären Systemmodell verwendet werden kann. Die SysML sieht durch die Spezifikation der parametrischen Modellelemente eine vom Werkzeuganbieter unabhängige Kopplungsmöglichkeit zur numerischen Berechnung vor. Beobachtungen im Rahmen der Anwendungsstudien zeigen, dass die Einbindung der Berechnungsmöglichkeiten in der praktischen Anwendung aus Anwenderperspektive im Vergleich zu RFLP durch die Anknüpfung externer Berechnungslösungen dabei von den Anbietern nicht optimal umgesetzt wird.

¹³² KLEINER & KRAMER 2013

¹³³ VITECH 2011; 2013a; 2013b

¹³⁴ NO MAGIC 2012

¹³⁵ DASSAULT 2016

Sowohl AS als auch CSM weisen weitere Auswertemöglichkeiten, wie Traceability-Matrizen auf. Im direkten Vergleich lässt sich beobachten dass, deren Nutzen im praktischen Einsatz maßgeblich von der Konfigurierbarkeit durch den Anwender abhängig ist. CSM weist dabei im direkten Vergleich zu AS effizientere und effektivere Mechanismen auf.

Der „Nutzen“ der Systemmodellierung konnte in der praktischen Anwendung mit den im Werkzeug verfügbaren, möglichen Auswertemechanismen korreliert werden. Der Nutzen steigt, wenn diese vom Anwender effizient und effektiv nach seinen Bedürfnissen einsetzbar und konfigurierbar sind.

- Integration der Aspekte Sprache und Methode

Während bei CORE das Werkzeug auf die mitgelieferte „System Definition Language“ (SDL)¹³⁶ ausgerichtet ist, integriert das RFLP Werkzeug Catia V6 die Sprache und Methode des an VDI2206 angelehnten AFLP¹³⁷ Ansatzes.

Durch das der Modellierung mit SysML zugrunde liegende Konzept der Trennung der drei Säulen¹³⁸ weisen beide SysML Lösungen eine Trennung von Sprache, Methode und Werkzeug auf. Der dieser Trennung zugrunde liegende Gedanke einer Unabhängigkeit von Sprache, Methode und Werkzeug scheint in der praktischen Anwendung jedoch nicht zu greifen. Die Anwendungsstudien zeigen beispielsweise, dass trotz Konformität einer Maßnahme zur Sprachspezifikation die Anwendbarkeit im Werkzeug dennoch nicht zwingend sichergestellt ist.¹³⁹

Bei Ansätzen Konzept einer Trennung von Sprache, Methode und Werkzeug konnte kein Mehrwert in der praktischen Anwendung erfahren werden. Das Potenzial einer universell, werkzeugunabhängig anwendbaren Systemmodellierung durch eine unabhängige Sprachspezifikation konnte in der praktischen Anwendung nicht gehoben werden.

¹³⁶ VITECH 2011

¹³⁷ EIGNER & GILZ 2012

¹³⁸ Vgl. Kapitel 2.4 Die drei Säulen interdisziplinärer Systemmodellierung

¹³⁹ MUNKER ET AL. 2014a

5.2 Zwischenfazit

Ergebnis der Analyse der relevanten Ansätze stellt mit den gefundenen Gemeinsamkeiten in Sprachraum, methodischen Schritten und Aspekten der Werkzeuge eine mögliche relevante Basis für die interdisziplinäre Systemmodellierung dar. Sie beantwortet damit die Forschungsfrage F1.¹⁴⁰

Weiteres Ergebnis sind die abgeleiteten Erkenntnisse aus den beobachteten Mustern des grafischen Vergleichs mit der Forderung nach schlankem Umfang und systematischem Aufbau, sowie die aus dem tabellarischen Sprachvergleich abgeleitete Forderung nach Redundanzfreiheit und Konsistenz.

Die Ergebnisse deuten darauf hin, dass die integrative Berücksichtigung von Sprache, Methode und Werkzeug einen zentralen Anteil an der Synthese des anwenderorientierten Ansatzes einnehmen sollte.

¹⁴⁰ Vgl. Kapitel 3.3 Forschungsfragen

6 Randbedingungen zur interdisziplinären System-modellierung

In diesem Kapitel werden die Randbedingungen zur Systemmodellierung vorgestellt. Diese sind aus der Zielsetzung nach Aspekten des erweiterten ZHO-Modells¹⁴¹ und aus Beobachtungen im Rahmen der Anwendungsstudien¹⁴² und der Zielsetzung abgeleitet worden. Sie sind im Rahmen der Vorstellung auf dem TdSE 2015¹⁴³ erwähnt worden und werden hier inhaltlich vertieft dargestellt. Unmittelbar erforderliche Randbedingungen werden als für den Umgang mit der Modellierung zwingend erforderlich verstanden. Die ergänzenden Randbedingungen sind ebenfalls erforderlich. Sie fließen zwar in die Synthese, jedoch nicht in den Prototypen zur Validierung der Tragfähigkeit des Ansatzes ein.

6.1 Formalisierung

Die INCOSE führt im Rahmen ihrer Publikation zum Systems Engineering in die Thematik der Formalisierung von Modellen ein.¹⁴⁴ Unter „formal“ werden hier strikt spezifizierte Kriterien verstanden, nach denen modelliert werden muss, damit das Modell formalen Ansprüchen genügt. Die Interpretation formaler Modelle sei eindeutig durch die formale Spezifikation möglich. Der Zweck der interdisziplinären Systemmodellierung kann darin gesehen werden, durch formalisierte Modellierung ein formales Modell anzustreben, welches inhaltlich vollständig und berechenbar sei. Das Gegenstück dazu stellt die informale Modellierung mit beispielsweise Freitext-Formulierungen oder Skizzen dar. Unter „informal“ wird hier eine freie Modellierung verstanden, der keine Spezifikation zu Grunde liegt. Die Interpretation informaler Modelle sei unscharf. Erst Absprachen oder Vereinbarungen können die Interpretation informaler Modelle fixieren.

In der Beschreibung einer notwendigen Ausprägung von Formalisierung bleibt INCOSE vage. Sie beschreibt, dass im Kontext des Systems Engineering die Modellierung einen gewissen Grad an Formalisierung enthalten muss, jedoch nicht welchen.¹⁴⁴

¹⁴¹ Vgl. Kapitel 3.2 Zielsetzung

¹⁴² Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

¹⁴³ MUNKER & ALBERS 2015

¹⁴⁴ WALDEN ET AL. 2015

Vor dem Hintergrund des opportunistischen Vorgehens des Menschen und der co-evolutionären Entwicklung von Ziel- und Objektsystem¹⁴⁵ ist auch in den Anwendungsstudien beobachtet worden, dass sich die Modellbildung inhaltlich evolutionär und spezifisch für die jeweiligen Problemstellungen der Entwicklungstätigkeit ausgebildet hat. Eine im Vorfeld getroffene vollständige, allgemeingültige Formalisierung wird daher als unrealistisch eingeschätzt.

Aus diesen Beobachtungen lässt sich ableiten dass eine Modellierung, die dem denkenden und handelnden Menschen gerecht werden soll, semi-formal gestaltet sein muss. Sie sollte aus einer einfachen Basis-Modellierung bestehen und sich durch Erweiterungen vom Anwender entsprechend seiner Bedürfnisse anpassen lassen.¹⁴⁶ Die Bedeutung der Erweiterung sollte in einer Vereinbarung von den Anwendern niedergeschrieben werden.

6.2 Unmittelbar erforderliche Randbedingungen

Als unmittelbar erforderliche Randbedingungen sind hier solche identifiziert worden, welche für eine grundsätzliche Modellierung mit dem integrierten Ansatz zwingend notwendig sind, da sie für die eigentliche Modellierung, also für das Ermöglichen des Abbildens von Informationen im Modell, als wesentlich eingeschätzt wurden. Sie sind zwingend zu berücksichtigen und fließen daher in das initiale Zielsystem des Ansatzes¹⁴⁷ ein.

6.2.1 Anwenderorientierte Objektorientierung

Unter „Objektorientierung“ wird hier das Konzept von Klassen und Instanzen verstanden. Aus den Beobachtungen der Anwendungsstudien lässt sich ableiten, dass der Anwender im Modell mehrfache Instanzen einer Klasse bilden können muss, beispielsweise bei der Modellierung eines Energiespeichers mit zwei ähnlich ausgeprägten Sattelhälften.¹⁴⁸

Aus diesen Beobachtungen lässt sich ableiten, dass Objektorientierung in der Modellierung für den Umgang mit der Komplexität im Modell erforderlich ist.

Durch geeignete Nutzerführung und Darstellung könnte das Konzept von Klassen und Instanzen vor dem Anwender versteckt werden, ohne dass dabei auf die

¹⁴⁵ Vgl. Kapitel 2.3 Erweitertes ZHO-Modell

¹⁴⁶ Zwar lässt sich auch in der SysML eine Teilmenge von Sprachelementen als Basis-Sprache verwenden und erweitern. Dies würde jedoch einem integrierten Ansatz (Sprache-Methode-Werkzeug) und damit der zentralen Motivation und Zielsetzung dieser Arbeit zur Anwenderorientierung nach Kapitel 2.7 widersprechen.

¹⁴⁷ Vgl. Kapitel 7 Initiales Zielsystem des Ansatzes

¹⁴⁸ WAGNER 2015

Mechanismen der Objektorientierung verzichtet werden müsste. Dies könnte geschehen durch:

- konsequente Modellierung mit Instanzen und die Ausblendung der Klassen

Nach Beobachtungen im Rahmen der Anwendungsstudien kann der Umgang mit der Modellierung erleichtert werden, indem auf eine gleichzeitige Darstellung von Klassen (in SysML äquivalent „Blöcke“) und Instanzen (in SysML äquivalent „Part Properties“ / „Rollen“) verzichtet wird, um zu vermeiden dass der Anwender sowohl die durch Instanzen aufgebaute Systemstruktur (in SysML äquivalent „Containment Tree“), als auch mit der Verwaltung von Klassen in Klassenbäumen (in SysML äquivalent „Paketstruktur“) beaufschlagt wird.

Modellelementbäume sollten sich auf die Darstellung der Systemstruktur durch Instanzen beschränken, während Klassen intern vom Werkzeug verwaltet werden könnten.

- Nutzerführung bei der Instanziierung bestehender Klassen

Durch geeignete Gestaltung des Werkzeuges sollte der Anwender in der Lage sein, bereits existierende Elemente zu finden und erneut zu instanziierten, anstatt versehentlich identische Modellelemente mehrfach neu anzulegen. Dies könnte eine konsistente Modellierung und den Umgang mit der Komplexität im Modell erleichtern.

- Merkmale und Eigenschaften in Klassen abbilden

In mechatronischen Systemen sind, nach der zweiten Grundhypothese des von ALBERS UND MATTHIESEN entwickelten Contact-and-Channel-Ansatzes¹⁴⁹ (C&C²-A, weiterer Beitrag durch MATTHIESEN,¹⁵⁰ sowie ALBERS UND SADOWSKI¹⁵¹), Merkmale und Eigenschaften funktionsbestimmend. Eine Trennung von Klassen als Gerüst und Instanzen mit konkreten Werten für Merkmale, die während der Instanziierung eingesetzt werden (das entspricht dem Vorgehen in der Informatik), wäre hier aufgrund der vielfältigen Wechselwirkungen mit funktionsrelevanten Auswirkungen in mechatronischen Systemen nicht zielführend. Eine Möglichkeit wäre, sämtliche Merkmale und Eigenschaften mit ihren qualitativen und quantitativen Werten in den Klassen abzubilden.

¹⁴⁹ ALBERS & MATTHIESEN 2002

¹⁵⁰ MATTHIESEN 2002

¹⁵¹ ALBERS & SADOWSKI 2014

Notwendig für die Instanzen wäre das Attribut der Bezeichnung (Name der Instanz) und die Verbindung der Schnittstellen im Systemkontext. Dies wird anhand eines Beispiels am Validierungsprototyp näher dargestellt.¹⁵²

Dieses Konzept entspricht weitestgehend dem Prinzip der Rollenbildung von Klassen („Part Properties“) in SysML. Es würde aber den weiterführenden Mechanismus der eigentlichen Instanziierung in SysML (separate SysML „Instance Specification“ mit Belegung der quantitativen Merkmale) verwerfen.

6.2.2 Anwenderorientierte grafische Modellierung

Aus den Beobachtungen der Anwendungsstudien lässt sich ableiten, dass Relationen und Verbindungen von Schnittstellen schon bei einfachen Systemen eine netzartige Ausprägung aufweisen. Es konnte beobachtet werden, dass sich eine tabellarische Darstellung nur begrenzt eignet, netzartige Ausprägungen für den Anwender zugänglich abzubilden. Bei einer grafischen Modellierung der Systemarchitektur könnten zudem die Modellelemente analog zum physikalischen Aufbau positioniert werden.

Grafische Modellierung wird hier als notwendig für die interdisziplinäre Systemmodellierung verstanden.

Auf der grafischen Modellierungsfläche sollte der Anwender beliebige Modellelemente platzieren und in Relation setzen können. Im Gegensatz zur SysML sollten jedoch keine spezifischen Diagrammtypen¹⁵³ existieren, auf denen ausschließlich spezifische Modellelemente platziert werden dürften. Aus den Beobachtungen der Anwendungsstudien lässt sich ableiten, dass die Bedeutung der auf einer Diagrammfläche visualisierten Elemente sich hinreichend aus den verwendeten Elementen ergibt. Eine Einschränkung der Verwendbarkeit auf spezifische Diagramme kann vielmehr dazu führen, dass die komplette Nachvollziehbarkeit (Traceability) im Modell nicht mehr auf einem Diagramm dargestellt werden kann.

6.2.3 Anwenderorientierte Sichtenbildung

Aus den Beobachtungen der Anwendungsstudien kann abgeleitet werden, dass es ab moderat komplexen Systemen sinnvoll sein kann, für spezifische Anwendergruppen den Umfang des Modells auf spezifische Sichten aufzuteilen. Die Sichten könnten einen spezifischen Blickwinkel auf die gleichen Modellelemente zeigen. Diagramme einer grafischen Modellierung lassen sich auch unter dem Begriff

¹⁵² Vgl. Kapitel 8 Validierungsprototyp – SystemSketcher

¹⁵³ OMG 2015

„Sichten“ eingliedern, da sie für den Anwender eine Teilmenge der Modellinhalte repräsentieren. ALBERS und LOHMEYER erkennen die Sichtenbildung als generellen wesentlichen Bestandteil im Systems Engineering in Form von anwenderspezifischen „System-of-Interest“. Sie greifen dies in Ihrer Idee einer menschenzentrierten, nutzenstiftenden fortgeschritteneren Methode auf, die sie als „Advanced Systems Engineering“ bezeichnen.¹⁵⁴

Aus Beobachtungen der Anwendungsstudien lässt sich aber ableiten, dass zur Sichtenbildung neben der zweckgebundenen Filterung von Elementen zwingend auch die zweckgebundene hierarchische Gliederung der Modellelemente Berücksichtigung finden muss. Unter Sichten werden hier deshalb spezifische Blickwinkel auf die hierarchische Gliederung der Modellelemente verstanden. Nach diesen Beobachtungen sind spezifische hierarchische Gliederungen für spezifische Sichten erforderlich.¹⁵⁵

Es sollte dem Anwender eine Möglichkeit gegeben werden, spezifische Sichten nach Bedarf zu erstellen und zu verwalten.

6.3 Ergänzende Randbedingungen

Als ergänzende Randbedingungen sind hier solche identifiziert worden, welche die Produktivität einer Systemmodellierung sicherstellen können. Im Gegensatz zu den unmittelbar erforderlichen Randbedingungen, welche eine Modellierung ermöglichen, generieren die ergänzenden einen Nutzen aus dem Modell. Nach dieser Einschätzung kann ein Modell alleinig mit den unmittelbar erforderlichen Randbedingungen erstellt werden, jedoch sind die ergänzenden Randbedingungen für den operativen, nutzenstiftenden und praxistauglichen Einsatz einer Modellierung erforderlich. Sie fließen in das initiale Zielsystem¹⁵⁶ des Ansatzes ein. Zur Bearbeitbarkeit der Umsetzung eines Prototyps im Rahmen einer Dissertation, muss aus Aufwandsgründen bei der Implementierung¹⁵⁷ und Validierung¹⁵⁸ abgewogen werden, ob auch ohne realisierte ergänzenden Randbedingungen eine Erprobung der wichtigsten Annahmen durchgeführt werden kann.

¹⁵⁴ ALBERS & LOHMEYER 2012

¹⁵⁵ MUNKER ET AL. 2014a

¹⁵⁶ Vgl. Kapitel 7 Initiales Zielsystem des Ansatzes

¹⁵⁷ Vgl. Kapitel 8 Validierungsprototyp – SystemSketcher

¹⁵⁸ Vgl. Kapitel 9 Validierung des Ansatzes

6.3.1 Verhaltensmodellierung mit Zuständen

Für die systemübergreifende Verhaltensmodellierung im interdisziplinären Systemmodell könnte die Verwendung von diskreten Zuständen ausreichend sein. Unter „Zustand“ wird hier der Status eines Systems zu diskreten Zeitpunkten verstanden. Analog zu dem Verständnis des Contact-and-Channel-Ansatzes nach ALBERS UND SADOWSKI, nach dem die zustandsabhängige Funktionserfüllung mit der zustandsabhängigen Ausbildung von Wirknetzen aus der Wirkstruktur eines Systems einhergeht,¹⁵⁹ wird hier die Verbindung von Schnittstellen der Entitäten der Systemarchitektur als zustandsabhängig verstanden. Die Zustände sollten als Mechanismus im Werkzeug umgesetzt werden. Weiterführende Verhaltensmodellierung, wie Abläufe oder Sequenzen, könnten in einer fachspezifischen Modellierungstechnik erfolgen und darin auch vom Werkzeug ausführbar gestaltet werden.

6.3.2 Anwenderspezifische Erweiterung der Modellierung

Vor dem Hintergrund der Forderung nach anwenderorientierter, bedarfsgerechter Erweiterung,¹⁶⁰ sollte diese als Mechanismus im Werkzeug umgesetzt werden. Es sollte dem Anwender die Möglichkeit gegeben werden, Erweiterungen nach Bedarf zu erstellen und zu verwalten und dann auch in der Benutzeroberfläche des Werkzeugs nutzen zu können.

Weiterführend sollte die Konfiguration von Einschränkungen der Modellierung bzw. deren Erweiterungen vorgesehen werden. Der Anwender könnte beispielsweise Regeln erstellen, die Dekompatibilität von Schnittstellen prüfen. Dies sollte als Mechanismus im Werkzeug ausgeführt werden. So könnte beispielsweise die Konformität der Modellierung zu einer semi-formalen Vereinbarung geprüft und sichergestellt werden.

6.3.3 Umgang mit der grafischen Modellierung

Aus den Beobachtungen der Anwendungsstudien lässt sich ableiten, dass sich die Handhabbarkeit der grafischen Modellierung signifikant auf die Anwendbarkeit und Nutzbarkeit des gesamten Ansatzes zur interdisziplinären Systemmodellierung auswirkt. Es wären Mechanismen im Werkzeug vorzusehen, die dem Anwender das Explorieren des Modellinhaltes auf der grafischen Modellierungsfläche erlauben. Hierzu würden auch Indikatoren für ausgeblendete Elemente und Relationen, sowie Mechanismen zur effektiven automatischen Anordnung der Modellelemente zählen.

¹⁵⁹ ALBERS & SADOWSKI 2014

¹⁶⁰ Vgl Kapitel 3.2 Zielsetzung

6.3.4 Darstellungsformen von Traceability

Unter Traceability wird hier die Vernetzung von Modellelementen und deren Nachvollziehbarkeit verstanden. Sie stellt einen wesentlichen Baustein der interdisziplinären Systemmodellierung dar. Sie ist im initialen Zielsystem des Ansatzes zwingend vorzusehen und wurde im Rahmen der Analyse¹⁶¹ bereits als relevant identifiziert.

Aus den Beobachtungen der Anwendungsstudien lässt sich ableiten, dass der Nutzen von Traceability wesentlich durch die Darstellungsformen gesteigert werden kann. Hierzu wären ergänzend Mechanismen zur anwenderkonfigurierbaren Anzeige der Vernetzung in tabellarischer, oder in matrizenartiger Darstellung vorzusehen, um mehr Nutzen aus der Traceability zu generieren.

6.3.5 Auswertemechanismen am Modell

Dieser Punkt greift die Betrachtung der Semi-Formalisierung auf. Das semi-formale Modell ist nicht mehr berechenbar, aber rechnerinterpretierbar. Rechnerische Auswertungen des Modells wären ebenso anwenderspezifisch zu konfigurieren wie spezifischen Erweiterungen der Modellierung. Es müsste also vom Anwender im jeweiligen Kontext entschieden werden, welche Arten von Berechnungen anhand des Modells durchgeführt werden sollen. Es sollte angedacht werden, dem Anwender Mechanismen im Werkzeug bereitzustellen, mit denen er Auswertungen nach seinen Bedürfnissen selbst konfigurieren kann.

6.3.6 Vererbung und Weiterverwendung

Die Ableitung von Elementen wird in objektorientierten Konzepten als Vererbung bezeichnet. In den Anwendungsstudien konnte beobachtet werden, dass im Umgang mit Vererbung durch Mehrfachvererbung und bedingte Vererbung eine Komplexität im Umgang mit der Modellierung erzeugt wird. Die Vererbung könnte durch ein separates Modul des Werkzeugs, beispielsweise einer separaten „Komponentenbibliothek“ mit Varianten- oder Konfigurationsmanagement, abgebildet werden.

¹⁶¹ Vgl. Kapitel 5 Analyse der relevanten Ansätze

7 Initiales Zielsystem des Ansatzes

In diesem Kapitel wird die Synthese des Ansatzes in Form einer Spezifikation dargestellt. Diese basiert auf der Analyse relevanter Ansätze¹⁶² und den ermittelten Randbedingungen aus Anwendungsstudien¹⁶³ zur interdisziplinären Systemmodellierung. Diese Synthese ist vom Autor und ALBERS auf dem TdSE 2015 vorgestellt worden.¹⁶⁴ Sie wird hier aufgegriffen und inhaltlich vertieft dargestellt.

7.1 Initiales Zielsystem der Sprache

Maßgeblichen Einfluss auf die Synthese der Sprachelemente hat das Ergebnis der Analyse der Sprachen.¹⁶⁵ Die gefundenen Gruppierungen sind in **Partialmodelle** überführt worden.

Während der Synthese des Zielsystems der Sprache ist im Wesentlichen eine **Reduktion**, sowie eine **Neupartitionierung** der im Rahmen der Sprachenanalyse identifizierten Elemente durchgeführt worden.

Durch die Reduktion ergibt sich mit der Verschlinkung des Sprachraumes die Chance einer besseren Erfassbarkeit durch den Anwender. Durch die Spezifizierung von Erweiterungsmechanismen im Werkzeug¹⁶⁶ kann das Risiko, dass der Sprachraum für die Modellierung spezifischer Systeme zu knapp ausfallen könnte, ausgeschlossen werden.

Unter Neupartitionierung wird hier die Implementierung von ehemals im Sprachraum als relevant identifizierten Elementen oder ganzer ehemaliger Gruppierungen als Funktionalität im Werkzeug oder in der Methode verstanden. Dabei kann der Anwender durch Hilfestellungen in der Benutzeroberfläche des Werkzeugs oder im methodischen Leitfaden bei der Anwendung geführt werden. Beispielsweise muss der Anwender keine Sprachelemente, wie Parametrik-Blöcke für die parametrische Modellierung, erlernen, wenn diese als Berechnungs-Funktionalität in das Werkzeug integriert sind. Indem der Ansatz als integrative Gesamtlösung konzipiert wird, können alle Funktionalitäten des Werkzeugs im initialen Zielsystem spezifiziert werden, wodurch die Umsetzung sichergestellt werden kann.

¹⁶² Vgl. Kapitel 5 Analyse der relevanten Ansätze

¹⁶³ Vgl. Kapitel 2.8 Anwendungsstudien interdisziplinärer Systemmodellierung

¹⁶⁴ MUNKER & ALBERS 2015

¹⁶⁵ Vgl. Kapitel 5.1 Analyse der Sprachen

¹⁶⁶ Vgl. Kapitel 7.1 Initiales Zielsystem des Werkzeugs

Die Reduktion und Neupartitionierung des Sprachraums wird nachfolgend beschrieben und zusammenfassend in Tabelle 11 kompakt dargestellt.

Die **Verhaltensmodellierung** ist auf die Modellierung der Funktionsstruktur (funktionale Architektur) reduziert worden. Dieses beinhaltet Funktionen und Zustände, wobei die Zustandsmodellierung nicht Gegenstand des Sprachraumes, sondern einen Mechanismus im Werkzeug darstellt.¹⁶⁷

Die **parametrische Modellierung**, **Zusatzinformationen** und **Modell-Infrastruktur** sind aus dem Sprachraum ausgegliedert worden. Grundlage dieser Entscheidung sind die Hinterfragungen des Umfangs der Sprachräume sowie die Überlegungen zur rechnerischen Auswertbarkeit¹⁶⁸. Die ausgegliederten Inhalte sollen nicht Gegenstand des Sprachraumes sein, aber als Mechanismus des Modellierungswerkzeuges abgebildet werden.

Schnittstellen sind bei Funktionen und Komponenten zur Modellierung der Funktions- und Systemstruktur vorgesehen worden.

Maßgeblichen Einfluss auf die Ausbildung der **Relationen** hat die Forderung nach Redundanzfreiheit und Konsistenz,¹⁶⁹ sowie nach systematischem Aufbau.¹⁷⁰ Als Ergebnis sind in Summe zwei Relationen für allgemeine Beziehungen und Zuordnungsbeziehungen, sowie für die „Verbindung“ von Schnittstellen zur Verwendung vorgesehen worden. Vor dem Hintergrund des schlanken und systematischen Aufbaus des Sprachraums sind die Relationen bidirektional ausgeprägt worden. Die genaue Bedeutung der Relation soll sich aus dem Kontext der in Beziehung gesetzten Modellelemente ergeben.

Für **Verbindung** ist als einziges Sprachelement eine instanzenspezifische Ausprägung vorgesehen worden: Verbindungen sollen spezifisch zu den Instanzen der Systemkomponenten existieren, dessen Schnittstellen sie verbinden. Dadurch ist die Modellierung von Wirknetzen nach dem Contact-and-Channel-Ansatz C&C²-A¹⁷¹ vorbereitet, indem die Schnittstellen von Systemkomponenten erst in Abhängigkeit des instanzenspezifischen Einsatzes verbunden bzw. gepaart werden und so, analog zur Modellierung mit Wirkflächenpaaren im C&C²-A, eine Art Wirk-Netz ausbilden.

Dies wird am Beispiel der vier Räder eines PKW erläutert und in Bild 15 schematisch dargestellt: Alle Instanzen der Klasse „Rad“ besitzen die gleichen Schnittstellen.

¹⁶⁷ Vgl. Kapitel 6.3.1 Verhaltensmodellierung mit Zuständen

¹⁶⁸ Vgl. Kapitel 6.3.5 Auswertemechanismen am Modell

¹⁶⁹ Vgl. Kapitel 5.2 Redundanz und Konsistenz in den Sprachräumen

¹⁷⁰ Vgl. Kapitel 5.3 Grafischer Vergleich der Sprachräume

¹⁷¹ ALBERS & MATTHIESEN 2002

Diese seien aber wie folgt instanzenspezifisch verbunden: Während die Vorderräder mit der vorderen Achse verbunden sind, werden die Hinterräder mit der hinteren Achse verbunden. Die instanzenspezifische Ausprägung dieser Verbindungen hat funktionsrelevante Auswirkungen auf die beteiligten Systemkomponenten (beispielsweise Vorderräder + vordere Achse: Antriebskräfte übertragen; Hinterräder + hintere Achse: Spur führen; im Falle von Frontantrieb).

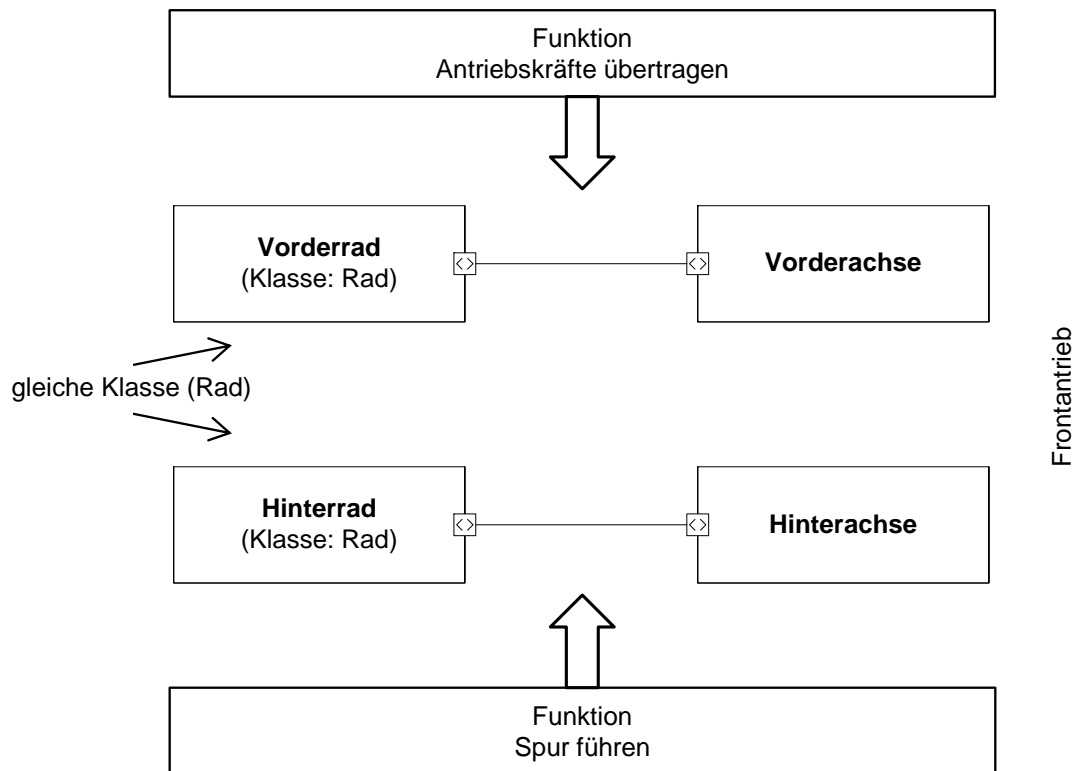


Bild 15: Instanzenspezifische Ausprägung der Verbindungen der Klasse „Rad“ mit ihren Instanzen „Vorderrad“ und „Hinterrad“ mit funktionsrelevanten Auswirkungen

Im Ansatz sollen durch die instanzenspezifische Ausprägung der Verbindung diese funktionsrelevanten Auswirkungen, die sich aus dem Aufbau eines Wirknetzes ergeben, abgebildet werden können. Die „Verbindung“ soll jedoch keine weiteren Informationen mit Ausnahme der Zugehörigkeit zu Zuständen enthalten. Alle relevanten Informationen zu Flüssen sollen als Merkmale und Eigenschaften der Schnittstellen modelliert werden.

Die **Subsumtionsrelation** dient zur Strukturierung der Modellelemente. Sie ist aus dem Sprachraum ausgegliedert worden. Eine Abbildung von Strukturbäumen im Werkzeug bei gleichzeitiger Definition mit Subsumtionsrelationen im Sprachraum würde eine Redundanz darstellen. Die Strukturierung soll deshalb in Form von Strukturbäumen durch einen Mechanismus des Werkzeugs abgebildet werden. Die **Vererbungsrelation** ist vor dem Hintergrund der Forderung nach

anwenderorientierter Umsetzung der Objektorientierung¹⁷² aus dem Sprachraum ausgegliedert worden. Sie soll Gegenstand eines Moduls des Werkzeugs zur Verwaltung von Modellelementen sein.

Zur Modellierung der **Systemstruktur** ist die Entität „Komponente“ vorgesehen worden. Unter „Komponente“ werden hier die Elemente in den Detaillierungsebenen der Systemstruktur (Sub- und Teilsysteme) verstanden. In den Anwendungsstudien wurde beobachtet, dass der Begriff „Komponente“ von Entwicklern in der Automobilindustrie sowohl für Sub- als auch für Teilsysteme auf verschiedenen Detaillierungsebenen verwendet und verstanden wird. Beispielsweise wird die (virtuelle) „Komponente“ Thermomanagement als ein Teilsystem verstanden, obwohl als „Komponente“ bezeichnet.¹⁷³ Vor dem Hintergrund der übergeordneten anwenderorientierten Ausrichtung ist deshalb der Begriff „Komponente“ zur Modellierung der Systemstruktur gewählt worden, da er sich in der Sprache der Anwender für die Beschreibung von Sub-, Teilsystemen und Einzelkomponenten gleichermaßen eignet.

7.1 Initiales Zielsystem der Methode

Maßgeblichen Einfluss auf die Synthese der Methode hat das Ergebnis der Analyse der Methoden.¹⁷⁴ Die gefundenen Gruppierungen und methodischen Einzelschritte stehen im Einklang mit den Partialmodellen der Sprachelemente¹⁷⁵ und konnten nach diesen gegliedert werden.

Während der Synthese des Zielsystems der Methode ist im Wesentlichen eine **Reduktion** auf die als wesentlich eingeschätzten Einzelschritte einer interdisziplinären Systemmodellierung durchgeführt worden.

Durch die Reduktion ergibt sich mit der Verschlankung der zu erlernenden methodischen Einzelschritte die Chance einer besseren Erfassbarkeit durch den Anwender. Durch die Spezifizierung von Erweiterungsmechanismen im Werkzeug¹⁷⁶ kann das Risiko, dass die Methodik für die Modellierung spezifischer Systeme nicht mehr applizierbar ist, minimiert werden. Der Anwender kann durch die semi-formale Ausprägung des Ansatzes mit Hilfe von Vereinbarungen eigene methodische Schritte definieren und diese anwenden, indem durch die Erweiterbarkeit des Sprachraumes dafür zweckgerichtete eigene Typen von Sprachelementen erstellt werden.

¹⁷² Vgl. Kapitel 6.2.1 Anwenderorientierte Objektorientierung

¹⁷³ MUNKER ET AL. 2014a

¹⁷⁴ Vgl. Kapitel 5.4 Analyse der Methoden

¹⁷⁵ Vgl. Kapitel 7.1 Initiales Zielsystem der Sprache

¹⁷⁶ Vgl. Kapitel 7.1 Initiales Zielsystem des Werkzeugs

Tabelle 12 stellt diese im Folgenden näher beschriebene Synthese kompakt dar.

Der **Systemkontext** soll im Partialmodell der Systemstruktur auf oberster Detaillierungsebene mit dessen Sprachelementen modelliert werden.

Die Modellierung von **Zielen, Fachwissen und Konzeptstruktur** ist nicht als Basis-Bestandteil der Methode und der Sprache vorgesehen worden: Der Ansatz soll eine Basismethode und Basissprache beinhalten, die vom Anwender bei Bedarf, beispielsweise um Ziele nach dem Ansatz zur Modellierung von Zielsystemen nach ALBERS ET AL.¹⁷⁷ und EBEL¹⁷⁸ erweitert werden kann.

Die **Verhaltensmodellierung** ist auf die Modellierung einer funktionalen Struktur und von Zuständen reduziert worden.

Die **Vorgehensweise** zur Modellierung soll nach dem Prozessverständnis der Karlsruher Schule der Produktentwicklung¹⁷⁹ flexibel erfolgen.

7.1 Initiales Zielsystem des Werkzeugs

Maßgeblichen Einfluss auf die Synthese des Werkzeugs hat die Forderung nach integrativer Betrachtung von Sprache und Methode¹⁸⁰ sowie die unmittelbaren und ergänzenden Randbedingungen.¹⁸¹

So sind durch die integrative Betrachtung der Methode im Werkzeug beispielsweise dedizierte Bereiche durch Schaltflächen vorzusehen, die den Anwender durch die methodischen Schritte leiten.

Zusatzinformationen, wie beispielsweise Kommentare, sind als Mechanismus im Werkzeug zu implementieren. Im modalen Dialog soll jedem Modellelement ein Freitext Kommentar hinzugefügt werden können.

Merkmale und Eigenschaften sind Werkzeuggestützt abzubilden, indem jeder Entität im modalen Dialog beliebige Merkmale und Eigenschaften zugewiesen werden können.

Sprachelemente zur **Modell-Infrastruktur**, beispielsweise die Entität „Paket“ zur Strukturierung des Modells, werden durch die Werkzeugunterstützung nicht mehr benötigt. Die Strukturierung des Modells soll anhand von Strukturbäumen aufgebaut werden.

¹⁷⁷ ALBERS ET AL. 2013b

¹⁷⁸ EBEL 2015

¹⁷⁹ ALBERS 2010

¹⁸⁰ Vgl. Kapitel 3.2 Zielsetzung

¹⁸¹ Vgl. Kapitel 6 Randbedingungen zur interdisziplinären System-modellierung

Tabelle 12: Initiales Zielsystem der Methode aus den in den Ansätzen identifizierten methodischen Einzelschritten. Entfall durch Entscheidung zur Reduktion (gekennzeichnet durch gestrichelte Linien).

Initiales Zielsystem der Methode als Syntheseergebnis aus den identifizierten methodischen Einzelschritten der Ansätze	OOSEM with SysML	IBM Telelogic Harmony SE	IBM RUP SE	Vitech	JPL State Analysis	Dori OPM	RFLP in Catia V6	CONSENS	Sysmod incl. FA5	SysKIT	Synthese initiales Zielsystem Methode
Systemkontext											
modelliere Systemkontext	X							X	X	X	X
Anwendungsfälle											
modelliere Akteure/Stakeholder	X	X							X	X	X
modelliere Anwendungsfälle	X	X	X						X	X	X
Anforderungen											
modelliere Ziele									X		X
modelliere Anforderungen	X	X	X	X	X		X	X	X	X	X
Verhalten											
modelliere Verhalten / Systemprozesse	X	X		X		X	*	X	X	X	X
modelliere Zustände	X	X			X	X			X	X	X
modelliere Systemfunktionen (funktionale Struktur)	X	X			X	X	X	X	X	X	X
Systemstruktur											
modelliere Fachwissen									X		X
modelliere Konzeptstruktur	X								X		X
modelliere Systemstruktur (physische Struktur)	X	X		X	X	X	X	X	X	X	X
Validierung											
modelliere Testfälle	X	**		X						X	X
Vernetzung der Partialmodelle											
vernetze übergreifend (traceability)	X	X		X			X	X		X	X
Vorgehen											
modelliere streng sequentiell		X				X					X
modelliere iterativ	X			X			X				X
modelliere flexibel			X					X	X	X	X
* Verhalten / Systemprozesse werden in RFLP Catia V6 über Dymola abgebildet											
** Integration von Rational Rhapsody TestConductor											

Weiterhin sind alle weiteren Forderungen der **unmittelbaren und ergänzenden Randbedingungen** zu übernehmen. Als ein zentraler Aspekt wird hier das anwenderorientierte Konzept der Objektorientierung und der grafischen Modellierung erwähnt.

Die Modellierung der Entitäten hat in **Strukturbaumen** zu erfolgen. Dem liegt die Ausgliederung der Subsumtionsrelation aus dem Sprachraum zugrunde. Jeder Strukturbaum soll durch ausschließliche Visualisierung von Instanzen der Forderung nach **anwenderorientierter Objektorientierung** folgen. Klassen sollen vom Werkzeug intern verwaltet werden. Bearbeitet der Anwender eine Instanz im Strukturbaum, sollen diese Änderungen werkzeugintern auf die Klasse und auf alle weiteren Instanzen dieser Klasse rückgespiegelt werden.

In der **grafischen Modellierung** soll der Anwender beliebige Modellelemente auf der Modellierungsfläche platzieren und in Relation setzen können. Zusätzlich zu den starken Mechanismen zur Exploration hat das Werkzeug die Möglichkeit zu bieten, Diagramme bedarfsgerecht als Favoriten abzuspeichern. Im Gegensatz zur SysML soll der Anwender dabei nicht an spezifische Diagrammtypen gebunden sein, sondern bedarfsgerecht Diagramme beliebig benennen und verwalten können.

Dieses Konzept der grafischen Modellierung lässt sich mit dem prägnanten Satz zusammenfassen: **Der Ansatz kennt keine spezifischen Diagrammtypen, aber kann dem Anwender so viele Diagramme wie gewünscht ermöglichen.**

7.2 Zwischenfazit initiales Zielsystem

Unter Berücksichtigung der Analyseergebnisse¹⁸² und ermittelten Randbedingungen¹⁸³ ist das initiale Zielsystem des Ansatzes entstanden. Es beinhaltet integrativ die Aspekte von Sprache, Methode und Werkzeug und spezifiziert damit den Ansatz als anwenderorientierten und integrierten Ansatz. Sie bildet einen schlanken Ansatz ab, der dem Anwender einen niederschweligen Einstieg in die interdisziplinäre Systemmodellierung eröffnen könnte. Diese Gesamtspezifikation beantwortet die Forschungsfrage F2.¹⁸⁴

Tabelle 13 zeigt zusammenfassend die auf dem TdSE 2015 vorgestellte Gesamtspezifikation¹⁸⁵ in einer aktualisierten Version, die an die Formulierungen dieser Arbeit angepasst wurde.

¹⁸² Vgl. Kapitel 5 Analyse der relevanten Ansätze

¹⁸³ Vgl. Kapitel 6 Randbedingungen zur interdisziplinären System-modellierung

¹⁸⁴ Vgl. Kapitel 3.3 Forschungsfragen

¹⁸⁵ MUNKER & ALBERS 2015

Tabelle 13: Gesamtspezifikation des Ansatzes. Aktualisierte, zusammengefasste Darstellung nach der Veröffentlichung des TdSE 2015¹⁸⁵

	Sprache	Methode	Werkzeug
Partialmodelle zur Systemmodellierung	Systemkontext		
		Modelliere Systemkontext	
	Anwendungsfälle		
	Akteure	Modelliere Anwendungsfälle und Akteure	Reiter Anwendungsfälle
	Anwendungsfall	Akteure	
	Anforderungen		
	Anforderung	Modelliere Anforderungen	Reiter Anforderungen
	Validierung		
	Testfall	Modelliere Testfälle	Reiter Testfälle
	Funktionsstruktur (funktionale Architektur)		
	Verhaltensmodellierung beschränkt sich auf Modellierung einer Funktionsstruktur (funktionale Architektur) mit diskreten Zuständen		
	Funktion	Modelliere eine Funktionsstruktur (funktionale Architektur)	Reiter Funktionen
	Schnittstelle		
		Modelliere diskrete Zustände der Funktionsstruktur und Systemstruktur	Zustandsmodellierung der Schnittstellen der Funktionsstruktur und der Zusammenhänge der Systemstruktur
	Systemstruktur (logische Architektur)		
	Komponente	Modelliere Systemstruktur (logische Architektur)	Reiter Systemelemente
Schnittstelle			
Relationen			
Relation	Vernetze Modellelemente		
Zuordnung	systemübergreifend		
Verbindung	Verbinde Schnittstellen		
Infrastruktur	Strukturierung der Modellelemente (Subsumption)		
			Strukturbäume für Anwendungsfälle, Anfordeungen, Testfälle, Funktionen und Systemelemente
	Zusatzinformationen Modellelemente		
			Kommentarfeld an jedem Modellelement
	Parametrische Modellierung		
			Attribute für jedes Modellelement. Verwaltung der Attribute im Tool.
	Unterstützung grafische Modellierung		
			Modellelemente lassen sich beliebig auf einer grafischen Modellierungsfläche anordnen. Grafische Anordnungen lassen sich als Diagramme speichern. Handling der gespeicherten Diagramme erfolgt separat zu den Strukturbäumen der Modellelemente
	Unterstützung Umgang mit Modellelementen		
			Verwaltung aller Modellelemente in einer Bibliothek
			Anwenderorientierte Anwendung von Objektorientierung
			Instanziierung in der Modellelemente Bibliothek
Erweiterungen			
	Erstelle weitere Modellelemente nach Bedarf. Modelliere weitere Partialmodelle nach Bedarf	Generierung und Verwaltung von neuen Modellelement-Typen	
allg. Methode	Vorgehen Modellierung		
			Wiederverwendung durch modellübergreifende Modellelemente Bibliothek
	generische Relationen zwischen beliebigen Modellelementen	Verwende alle Modellelemente zu jedem Zeitpunkt bedarfsgerecht	Beliebige Typen von Modellelementen gemeinsam auf der grafischen Modellierungsfläche

8 Validierungsprototyp – SystemSketcher

In diesem Kapitel wird ein Prototyp namens „SystemSketcher“ als erste Werkzeugrealisierung hinsichtlich des Funktionsumfangs und der Einschränkungen beschrieben. Es wird zur Validierung der grundsätzlichen Tragfähigkeit des Ansatzes verwendet. Aus Gründen der Bearbeitbarkeit im Rahmen einer Dissertation ist der Prototyp auf die exemplarische Umsetzung der wichtigsten Aspekte beschränkt worden. Diese ungesetzten Aspekte sind im wesentlichen die Synthesergebnisse¹⁸⁶ ohne die ergänzenden Randbedingungen.¹⁸⁷ Das Werkzeug ist in der rudimentären Form wahrscheinlich nicht geeignet, um eine vollständige Beurteilung der Usability durchzuführen, eignet sich aber zum Nachweis der grundsätzlichen Tragfähigkeit des Ansatzes. Der selbst zur Entwicklung einer rudimentären Realisierung entstandene Aufwand wird im nachfolgenden Unterkapitel näher beschrieben, um zur Aufwandsabschätzung für weitere Arbeiten zu dienen.

Der hier in Kapitel 8 zur Einführung des Validierungsprototypen dargestellten Modellierung liegt ein Beispielsystem zugrunde. Bei diesem Beispielsystem handelt es sich um das Modell eines Roboterstaubsaugers „RC 3000“, mit dem im Rahmen der Lehrveranstaltung „Mechatronische Systeme und Produkte (MSuP) am IPEK die interdisziplinäre Systemmodellierung mit SysML gelehrt wird.¹⁸⁸

8.1 Implementierung des Validierungsprototyps

Zur Entwicklung des SystemSketcher wurde sich an der Rollenverteilung des IT-Entwicklungsprozesses Scrum¹⁸⁹ orientiert. Der Autor, in der Rolle des „Product Owners“, formuliert statt in klassischen Anforderungsdokumenten die Spezifikation als „User Storys“, also als „Geschichten“ aus Sicht des späteren Anwenders. Eine qualifizierte, studentische Hilfskraft hat in der Rolle des Software-Entwicklers in einem iterativen Vorgehen die User Storys in Programmcode übersetzt. Die fertigen Artefakte sind dann vom Product Owner untersucht und beurteilt worden.

Der SystemSketcher ist in der Entwicklungsumgebung Microsoft .NET¹⁹⁰ mit der Programmiersprache C# realisiert worden. Der Programmieraufwand des Software-Entwicklers hat in Summe über 425 Stunden betragen. Mit Entwicklungspausen hat

¹⁸⁶ Vgl. Kapitel 7 Initiales Zielsystem des Ansatzes

¹⁸⁷ Vgl. Kapitel 6.3 Ergänzende Randbedingungen

¹⁸⁸ MATTHIESEN ET AL. 2014

¹⁸⁹ GLOGER 2016

¹⁹⁰ Vgl. www.microsoft.com/net

die Entwicklungsdauer circa ein Jahr betragen. Innerhalb dieses Zeitraumes sind 98 Iterationen des SystemSketcher entstanden. SystemSketcher ist dabei auf 17.000 Lines of Code mit 24 Software-Modulen angewachsen. Bild 16 bildet das Vorgehen grafisch ab.

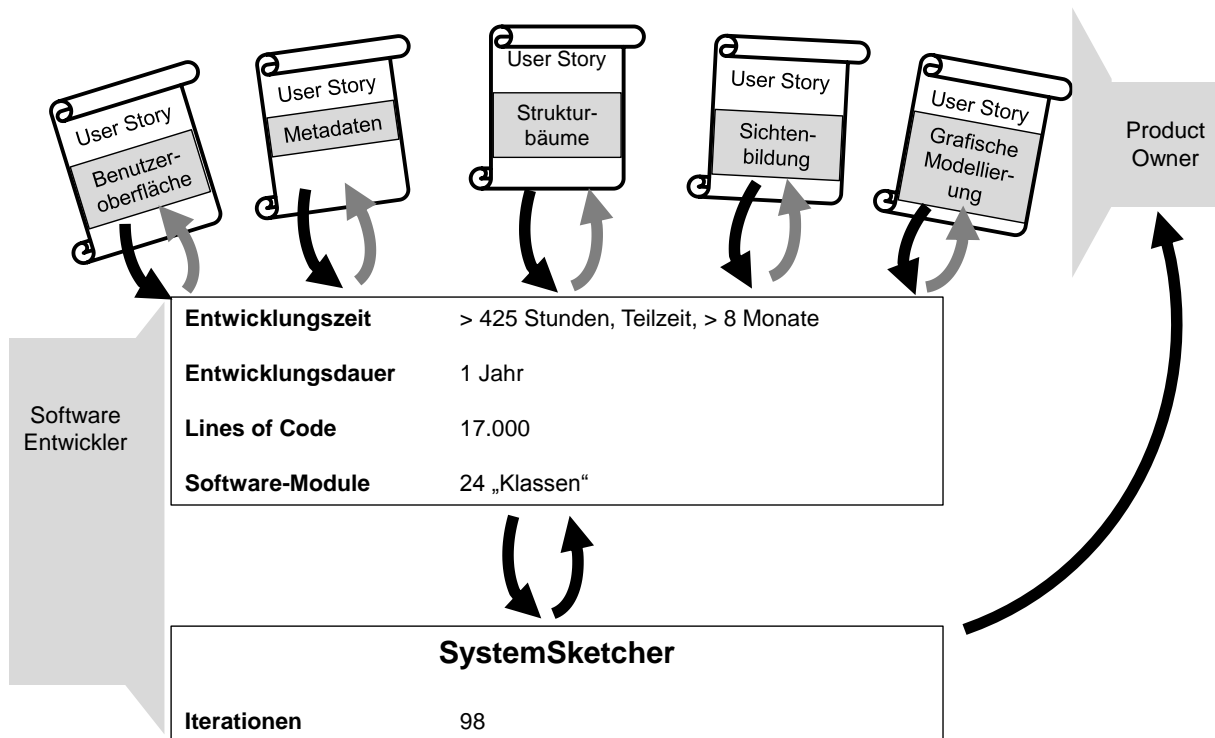


Bild 16: Entwicklung des Prototyp SystemSketcher

Während der Entwicklung ist es dem Software-Entwickler überlassen worden, den Software-Code auf Basis der User-Stories zu erstellen und zu dokumentieren.

Um die wissenschaftliche Qualität dieser Arbeit sicherzustellen, sind sämtliche Arbeiten nachvollziehbar durchgeführt worden, was weitere Arbeiten aufbauend auf dem erarbeiteten Stand ermöglichen und erleichtern soll.

Hierzu ist der Software-Code nachvollziehbar formuliert worden. So sind überwiegend Standard-Programmier-elemente der Programmiersprache C# verwendet worden. Beispielsweise ist das kompakte, aber für die Nachvollziehbarkeit erweiterte Programmierkompetenzen erfordernde Konstrukt von Lambda-Expressions, zugunsten einfacher Schleifen vermieden worden.¹⁹¹ Bei der Benennung von Methoden und Variablen ist auf eine sprechende, eindeutige Bezeichnung Wert gelegt worden.¹⁹¹ Mit diesen Maßnahmen konnte auf eine

¹⁹¹ Entwicklungsgespräch mit dem Entwickler Juli 2016

weitgehende Kommentierung im Code verzichtet werden, ohne die Nachvollziehbarkeit einzuschränken.

Die **Datenstruktur des SystemSketcher** ist vom Entwickler wie folgt umgesetzt worden.¹⁹² Diese Auflistung kann bei weiterführenden Arbeiten als Dokumentation herangezogen werden.

- objektorientierte Programmierung ermöglicht die Wiederverwendbarkeit von Software-Modulen
- 24 Software-Module in Form von Dateien der Endung *.cs mit dazugehörigen *.Designer.cs und *.resx

Die Software-Module lassen sich wie folgt kategorisieren:

- Einlesen des Projekts und der Meta-Daten von der Festplatte
- Grafische Darstellung
- Logische Verarbeitung von Informationen gemäß der User Storys
- Interaktionen des Users detektieren (Mausklicks, Tastatureingaben, etc)
- Speichern von Daten auf der Festplatte

Die **Datenstruktur der Nutzerdaten** ist vom Entwickler wie folgt umgesetzt worden.¹⁹²

- Zugrundeliegende Datenstruktur ist XML, Verwendung zweier XML-Dateien
- XML-Datei der Endung *.ssp bildet ein Projekt ab (SSP = „SystemSketcher Projekt“)
- XML-Datei der Endung *.md bildet die Metadaten des Projekts ab (MD = „Meta Daten“)
- XML-Dateien sind gegliedert nach Sections und Elementen

¹⁹² Entwicklungsgespräch mit dem Entwickler Juli 2016

8.2 Realisierung und Einschränkungen im Validierungsprototyp

Aus Gründen der Bearbeitbarkeit im Rahmen einer Dissertation sind bei der Implementierung des Prototyps Einschränkungen getroffen worden. Der Validierungsprototyp ist als erste Werkzeugrealisierung zu sehen. Die Einschränkungen fokussieren den Funktionsumfang des Prototypen auf wesentliche, aber für die Realisierung einer Erprobung der grundsätzlichen Tragfähigkeit des Ansatzes notwendigen Umfänge. Die Grundlage dazu bilden die als unmittelbar identifizierten erforderlichen Randbedingungen der interdisziplinären Systemmodellierung.¹⁹³ Die Einschränkungen betreffen in Wesentlichen die Beschränkung der Realisierung auf das Partialmodell „Systemstruktur“, so dass eine grundsätzliche Systemmodellierung an diesem erprobt werden kann. Die Modellierung in den restlichen Partialmodellen ist vereinfacht umgesetzt worden, so dass auch Anforderungen, Anwendungsfälle, Testfälle und Funktionen modelliert und mit Relationen vernetzt werden können.

Tabelle 14 zeigt die Realisierung der unmittelbar erforderlichen Randbedingungen im Validierungsprototypen SystemSketcher sowie die Einschränkungen in zusammengefasster Form. Eine detaillierte Beschreibung findet sich in den folgenden Unterkapiteln.

¹⁹³ Vgl. Kapitel 6.2 Unmittelbar erforderliche Randbedingungen

Tabelle 14: Unmittelbar erforderliche Randbedingungen zur interdisziplinären Systemmodellierung und Realisierung in SystemSketcher

unmittelbar erforderliche Randbedingungen	Realisierung in SystemSketcher
Anwenderorientierte Objektorientierung	<p>Komponenten Picker: Verwaltung von Klassen und Instanzen Einschränkung auf Partialmodell "Systemstruktur", Realisierung in restlichen Partialmodellen in rudimentärer Form</p> <p>Strukturbäume: Verwaltung und Abbildung von Strukturen Einschränkung auf Partialmodell "Systemstruktur", Realisierung in restlichen Partialmodellen in rudimentärer Form</p> <p>Metadaten Manager: Verwaltung und Abbildung von Merkmalen und Eigenschaften Einschränkung auf Systemkomponenten und deren Schnittstellen</p>
Anwenderorientierte grafische Modellierung	<p>Grafische Modellierungsfläche Einschränkung auf Systemkomponenten, deren Schnittstellen, sowie Verbindungs-Relation</p> <p>Relationen Manager: Verwaltung und Abbildung von Relationen Einschränkung auf eine Basis-Relation, keine grafische Darstellung</p>
Anwenderorientierte Sichtenbildung	<p>Sichtenbildung durch "Boarding Points", Anwenderspezifische Einstiegspunkte in die globale Systemstruktur Einschränkung auf Partialmodell der Systemstruktur</p>

8.3 Gesamtüberblick über Grafisches User Interface

Bild 17 zeigt die Benutzeroberfläche des SystemSketcher im Gesamtüberblick.

Der Arbeitsbereich teilt sich in die grafische Modellierungsfläche (Bild 11: GM), Schaltflächen (Bild 11: SF) zur Modellierung der Partialmodelle entsprechend der methodischen Schritte und Schaltflächen zur Modellierung von Merkmalen und Eigenschaften („Metadaten“ und „Relationen“), einem Bereich zur Verwaltung von Diagrammen (Bild 11: FAV) und einem Bereich für die Strukturbäume der Partialmodelle (Bild 11: BAUM) inklusive eines Mechanismus zur Sichtenbildung.

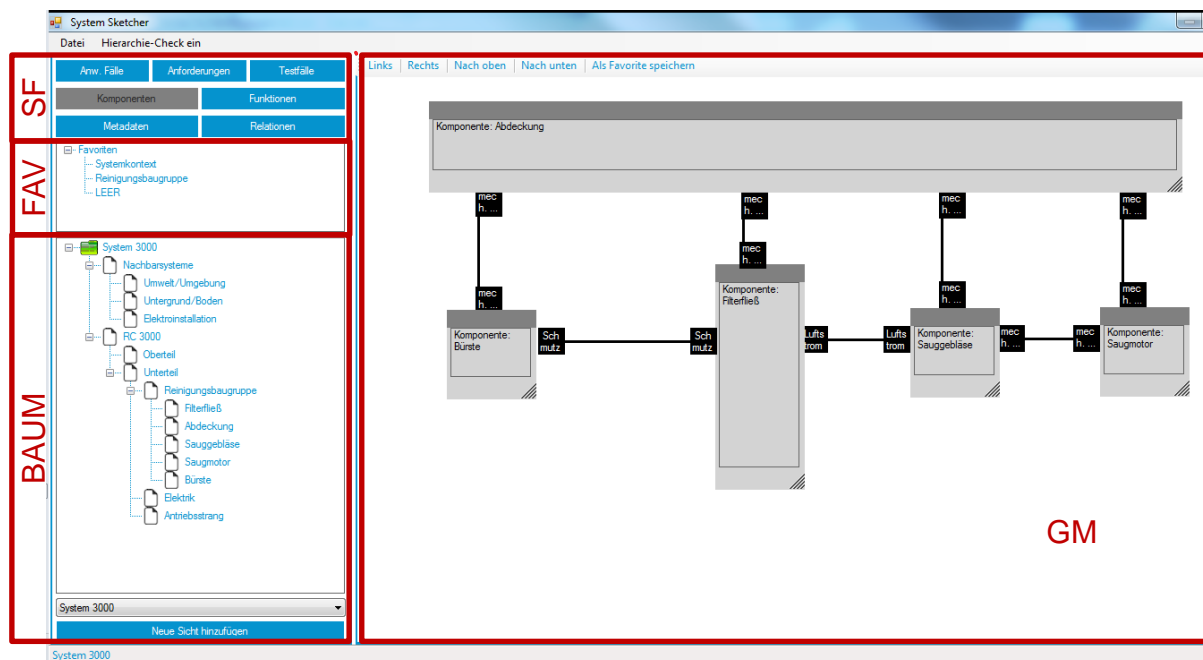


Bild 17: Gesamtüberblick SystemSketcher mit Modell eines Beispielsystems

8.4 Partialmodelle

Bild 19 zeigt den RC3000 Reinigungsroboter mit einigen zugehörigen Anwendungsfällen, Anforderungen und Funktionen. Diese Aspekte werden als Partialmodelle bezeichnet. Die Partialmodelle aus dem initialen Zielsystem des Ansatzes¹⁹⁴ sind „Systemkontext“, „Anwendungsfälle“, „Anforderungen“, „Validierung“, „funktionale Architektur“ und „logische Architektur“. Sie sind im SystemSketcher mit Begriffen belegt worden, die nach einer Einschätzung dem Ingenieur als Anwender zugänglicher sein sollten. Sie sind dort nach Bild 18 als „Anwendungsfälle“, „Anforderungen“, „Testfälle“, „Funktionen“ und „Komponenten“ implementiert worden. Der „Systemkontext“, hier als Einbindung des Systems in seine Nachbarsysteme verstanden, ist nicht separat ausgewiesen worden, sondern

¹⁹⁴ Vgl. Kapitel 7 Initiales Zielsystem des Ansatzes

wird im Partialmodell „Komponenten“ in Form von benachbarten Teilsystemen modelliert, damit der Anwender für den Wechsel zwischen System- und Umfeldbetrachtung nicht das Partialmodell wechseln muss. Er kann im Partialmodell „Komponenten“ das Umfeld und das System mit den gleichen Elementen modellieren.



Bild 18: Partialmodelle im SystemSketcher

Für die Partialmodelle sind nach Bild 18 in fünf Schaltflächen implementiert. Möchte der Anwender die Nachbarsysteme, sowie die Systemstruktur des Reinigungsroboters modellieren, wählt er „Komponenten“. Möchte er die Anforderungen modellieren, wählt er „Anforderungen“.

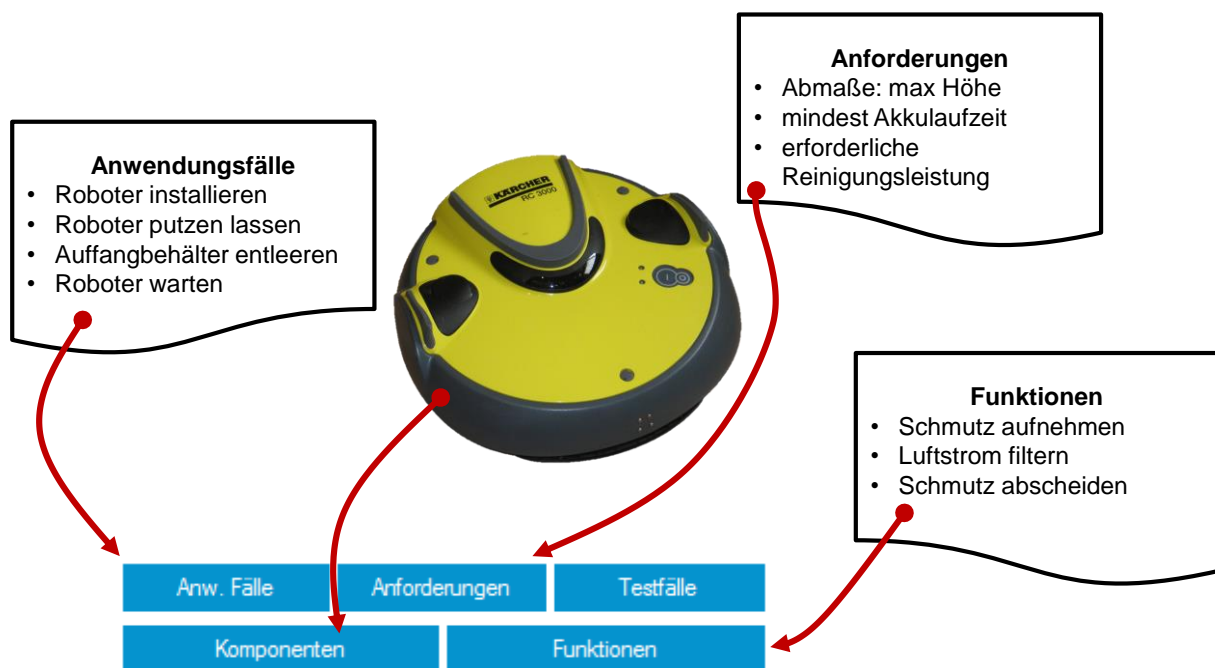


Bild 19: Reinigungsroboter "RC3000", exemplarische Anforderungen, Anwendungsfälle und Funktionen, sowie die Zuordnung zu den Partialmodellen im Validierungsprototyp

Der Anwender kann mit diesen dedizierten Schaltflächen zu Einen schnell und effektiv zu den jeweiligen Modellinhalten geführt werden und zum Anderen an die relevanten Modellinhalte erinnert werden.

8.5 Strukturbäume

Für jedes der einzelnen Partialmodelle ist ein Strukturbaum vorgesehen worden.

Die Strukturbäume gliedern die Modellelemente der Partialmodelle wie in Bild 20 anhand der Systemstruktur eines Beispielsystems dargestellt. Für jedes Partialmodell ist ein eigener Strukturbaum vorgesehen. Die Mechanismen zur Modellierung der Bäume sind partialmodellübergreifend konsistent gestaltet. Die anwenderorientierte Integration von Sprache, Methode und Werkzeug wird hier dadurch realisiert, indem dem Anwender bei der Erstellung von Modellelementen nur diejenigen verfügbar gemacht werden, welche zum jeweiligen Partialmodell gehören.

Möchte der Anwender die Systemstruktur des RC3000 Reinigungsroboters modellieren, wird er zuerst im Partialmodell „Komponenten“ die Baugruppen und Subsysteme des RC3000 in einer Baumstruktur anlegen und mit diesen dann auf der grafischen Modellierungsfläche arbeiten. Wie detailliert das System beschrieben wird, also ob die Abbildung von Baugruppen ausreichend ist oder bis auf Komponentenebene modelliert werden soll, muss und kann der Anwender nach seinem Bedarf selbst entscheiden.

Die Strukturbäume setzen einen Teil des Konzeptes anwenderorientierter Objektorientierung¹⁹⁵ um, indem ausschließlich die Instanzen visualisiert werden. Möchte der Nutzer beispielsweise eine zweite Bürste für eine Reinigungslösung mit Doppelbürste erstellen, kann er die bereits im Strukturbaum vorhandene mit Hilfe des „Komponenten-Picker“¹⁹⁶ erneut instanzieren.

Die Herausforderung bei der Umsetzung der Strukturbäume ist die Implementierung von Mechanismen zur Verwaltung der Modellelemente, so dass beispielsweise im Baum verschoben oder gelöscht werden können. **Um dem Implementierungsaufwand zu begegnen, sind hier ausschließlich die Funktionen des Erstellens und Umbenennens implementiert worden.** Die ermöglicht die Validierung der grundsätzlichen Tragfähigkeit, da einfache Modelle abgebildet werden können. Für die Erprobung im produktiven Einsatz ist es erforderlich, die weiteren Funktionen wie Löschen und Verschieben zum Umgang mit umfangreicheren Modellen zu implementieren.

¹⁹⁵ Vgl. Kapitel 6.2.1 Anwenderorientierte Objektorientierung. Weiteren Beitrag zur Umsetzung der anwenderorientierten Objektorientierung leistet der Komponenten Picker nach Kapitel 8.7.

¹⁹⁶ Vgl. Kapitel 8.7 Komponenten Picker

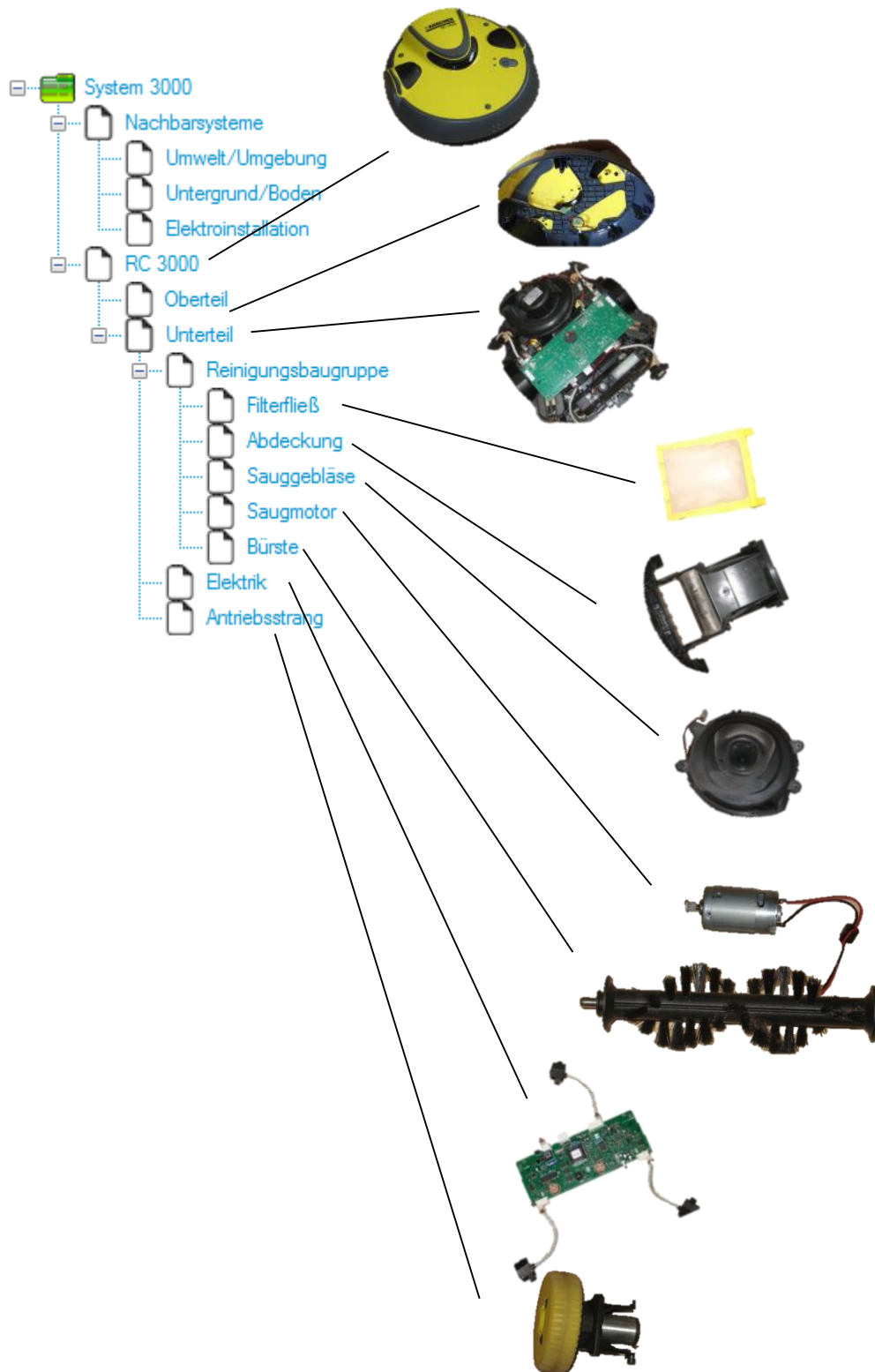


Bild 20: Auszug aus Strukturbaum („Komponenten Baum“) des RC3000

8.6 Sichtenbildung

Die Sichten bieten dem Anwender spezifische Blickwinkel auf das Modell. Im Prototyp ist ein Mechanismus zur Sichtenbildung¹⁹⁷ implementiert, der dem Anwender sichtenspezifische Einstiegspunkte in beliebige Äste des Strukturbaums erlaubt. Diese Einstiegspunkte werden als „Boarding Points“ bezeichnet. Das Prinzip ist in Bild 21 dargestellt.

Ein Entwicklungsingenieur, dessen Entwicklungsaufgabe die mechatronischen Systemkomponenten umfasst, wird sich nicht im selben Umfang für die restlichen Systemkomponenten, wie Abdeckungen oder die rein elektrischen Komponenten interessieren. Mit Hilfe der Sichtenbildung kann er sich eine für seinen Betrachtungshorizont angepasste Sicht erstellen, die dann ausschließlich die von ihm gewünschten mechatronischen Komponenten enthält. Dabei findet in SystemSketcher keine Filterung statt, sondern es werden Einstiegspunkte als Abkürzungen in die jeweiligen Äste des globalen Strukturbaumes erzeugt. Dadurch bleibt die Struktur unterhalb der Einstiegspunkte erhalten. Diese Sichten und die Boarding Points können vom Anwender selbst erstellt und verwaltet werden.

Die Herausforderung bei der Umsetzung der Sichtenbildung ist der Implementierungsaufwand in den Strukturbäumen aller Partialmodelle. **Um dem Aufwand zu begegnen, ist die Sichtenbildung ausschließlich im „Komponenten-Baum“ implementiert worden.** Dies ermöglicht die Validierung der grundsätzlichen Tragfähigkeit des Ansatzes. Für die Erprobung im produktiven Einsatz sollte die Sichtenbildung auch in den weiteren Partialmodellen implementiert werden, um auch Sichten für beispielsweise „Anforderungen“ oder „Funktionen“ erstellen zu können.

¹⁹⁷ Vgl. Kapitel 6.2.3 Anwenderorientierte Sichtenbildung

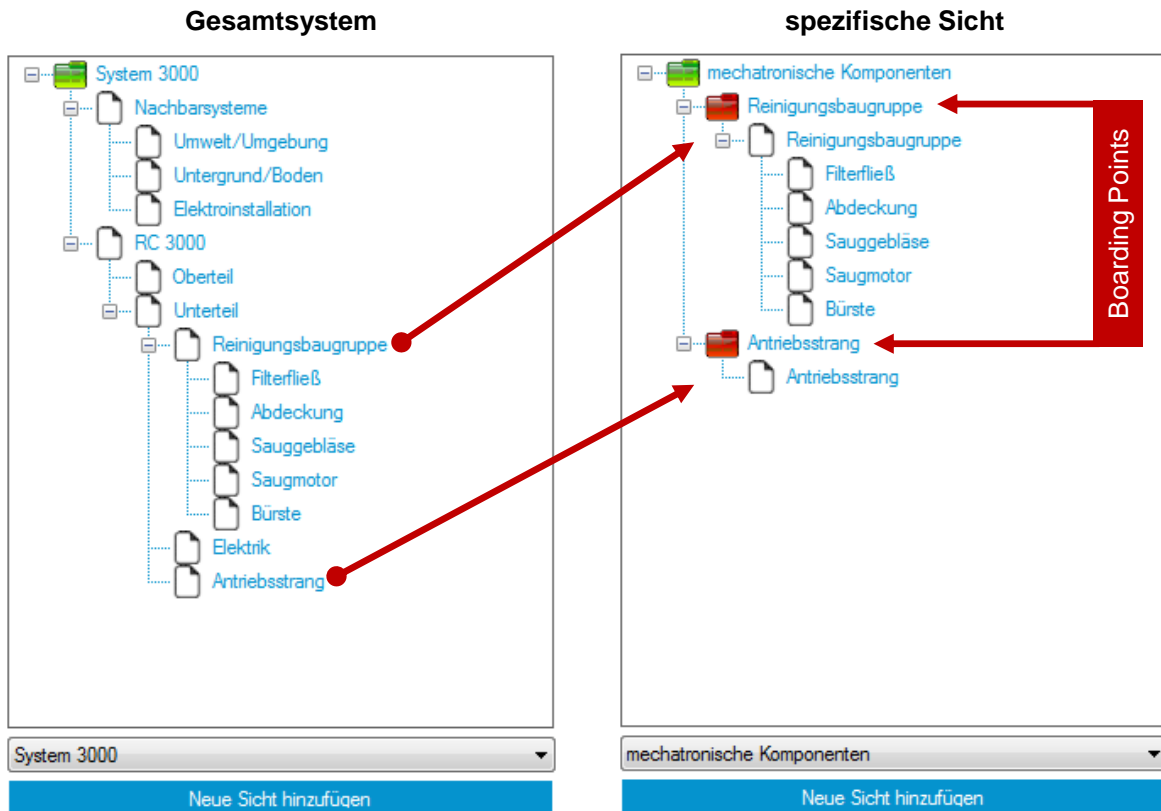


Bild 21: Prinzip und Umsetzung der Sichtenbildung:
Gesamtsystem und spezifische Sicht „mechatronische Komponenten“ durch Ausleitung der
„Reinigungsbaugruppe“ und „Antriebsstrang“

8.7 Komponenten Picker

Der „Komponenten Picker“ nach Bild 22 stellt den modalen Dialog zur Erzeugung neuer Modellelemente unter Berücksichtigung des Konzepts zur anwenderorientierten Objektorientierung¹⁹⁸ dar.

Vor dem Hintergrund der Forderung nach Nutzerführung zur Instanziierung bestehender Klassen wird der Nutzer in einem modalen Dialog geführt. In diesem Dialog kann der Anwender bestehende Modellelemente finden, bevor er neue anlegt. Bestehende Modellelemente können dupliziert¹⁹⁹ werden. Bei der Neuerstellung eines Modellelements wird eine Klasse sowie eine erste Instanz erzeugt. Die Klasse wird als Teil des anwenderorientierten objektorientierten Konzepts intern vom Werkzeug verwaltet und vor dem Anwender versteckt. Für den Anwender sichtbar ist ausschließlich die Instanz im Strukturbaum.

Fügt der Anwender einem Systemelement ein untergeordnetes Element hinzu, so werden nach dem Konzept der anwenderorientierten Objektorientierung unter jeder weiteren existierenden Instanz dieses Systemelements ebenfalls weitere Instanzen des neu erzeugten Elements eingefügt. Ein Mechanismus des Werkzeugs weist dabei durch fortlaufende Nummerierung der automatisch generierten Instanzen den Anwender auf duplizierte Elemente hin. Ein weiterer Mechanismus weist den Anwender darauf hin, Zyklen innerhalb eines Astes eines Strukturbaumes zu vermeiden.

Möchte der Entwicklungsingenieur beispielsweise das „Sauggebläse“ modellieren, so kann er im Komponenten Picker vor der Erstellung eines neuen Sauggebläses alle bereits erstellten Komponenten durchstöbern und alle bereits erstellten Komponenten finden, die den Namen „Sauggebläse“ beinhalten und die Gefundenen wiederverwenden. Dadurch kann vermieden werden, dass gleiche Komponenten mehrfach modelliert werden. Die Chance einer Wiederverwendung kann erhöht werden. Der Modellierungsaufwand kann durch Wiederverwendung bestehender Komponenten reduziert werden.

¹⁹⁸ Vgl. Kapitel 6.2.1 Anwenderorientierte Objektorientierung

¹⁹⁹ Im „Komponenten Picker“ ist der Begriff „duplizieren“ für den Mechanismus der erneuten Instanziierung der zugrunde liegenden Klasse gewählt worden. Der „Komponenten Picker“ kann durch den erfahrenen Anwender mit einer Tastenkombination übersprungen werden.

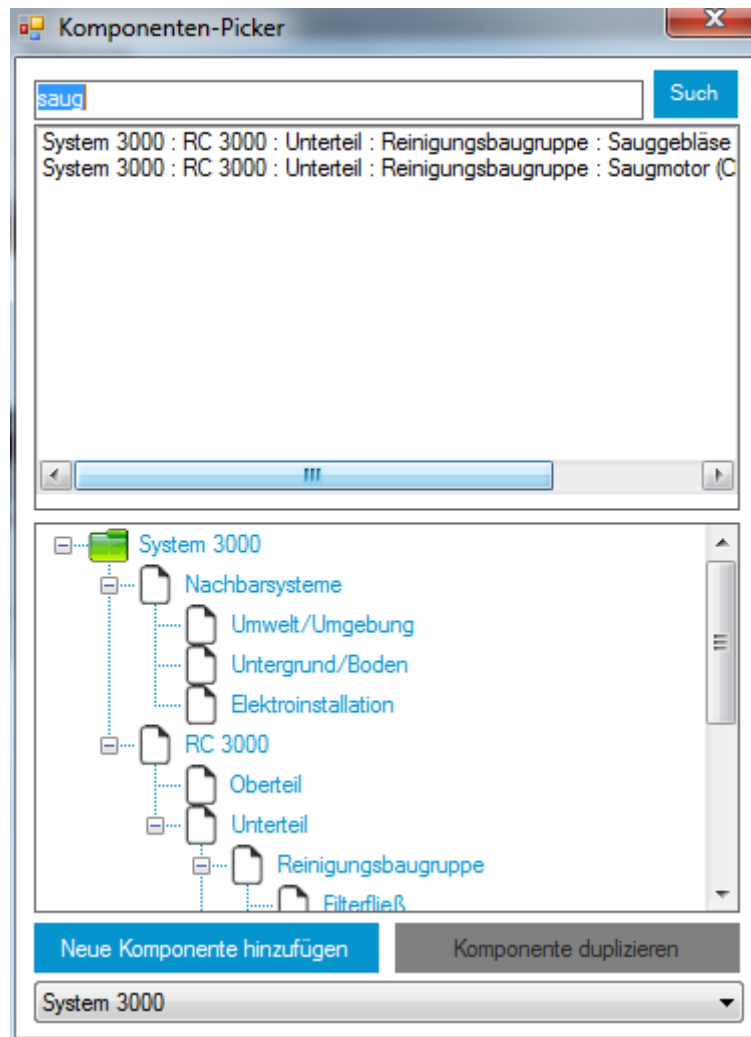


Bild 22: Komponenten Picker mit Suchfunktion und Ergebnisliste, sowie Auswahlmöglichkeit bestehender Modellelemente

Die Herausforderung bei der Umsetzung des Komponenten Pickers ist der Implementierungsaufwand für alle Modellelemente und Partialmodelle. **Um dem Aufwand zu begegnen, ist die Komponenten Picker ausschließlich im „Komponenten-Baum“ implementiert worden.** Elemente in den weiteren Partialmodellen werden mit einem vereinfachten modalen Dialog ohne Objektorientierung erzeugt. Dies ermöglicht die Validierung der grundsätzlichen Tragfähigkeit des Ansatzes. Für die Erprobung im produktiven Einsatz sollte der Komponenten Picker auch für die weiteren Partialmodelle und Modellelemente implementiert werden, um auch die Wiederverwendung von Anforderungen, Funktionen oder weiteren Elementen zu ermöglichen.

8.8 Metadaten Manager

Möchte der Entwicklungsingenieur beispielsweise dem Sauggebläse weitere Schnittstellen oder Metadaten hinzufügen, um es dann auf der grafischen Modellierungsfläche mit anderen Komponenten zu verbinden, oder möchte er prüfen, welche Metadaten dem Sauggebläse bereits hinzugefügt wurden, kann er dazu den Metadaten Manager verwenden.

Der „Metadaten-Manager“ nach Bild 23 stellt den modalen Dialog zur Erzeugung und Verwaltung von Merkmalen und Eigenschaften dar. Er ermöglicht dem Anwender die Prüfung und Sichtung der Merkmale und Eigenschaften der Modellelemente. Vor dem Hintergrund der Forderung nach konsistenter Modellierung²⁰⁰ können allen Modellelementen Merkmale und Eigenschaften bedarfsgerecht mit identischen Mechanismen hinzugefügt werden. Der Anwender legt dazu Merkmale und Eigenschaften zuerst global an und weist sie dann den Modellelementen zu.

Hier wird die anwenderorientierte Integration von Sprache, Methode und Werkzeug umgesetzt indem Merkmale und Eigenschaften, zu denen auch Schnittstellen zählen, nicht als eigenständige Elemente des Sprachraumes abgebildet, sondern im Metadaten-Manager des Werkzeugs verwaltet werden.

Vor dem Hintergrund des Konzeptes zur anwenderorientierten Objektorientierung²⁰¹ werden alle Merkmale und Eigenschaften werkzeugintern den Klassen zugeordnet. Änderungen an einer Klasse, beispielsweise durch Hinzufügen von Merkmalen oder Eigenschaften, wirken sich unmittelbar auf alle Instanzen dieser Klasse aus.

Die Herausforderung bei der Umsetzung des Metadaten Managers ist der Implementierungsaufwand für alle Modellelemente und Partialmodelle. **Um dem Aufwand zu begegnen, ist die Metadaten Manager ausschließlich für System“komponenten“, sowie deren Schnittstellen implementiert worden.**

²⁰⁰ Vgl. Kapitel 5.2 Redundanz und Konsistenz in den Sprachräumen

²⁰¹ Vgl. Kapitel 6.2.1 Anwenderorientierte Objektorientierung

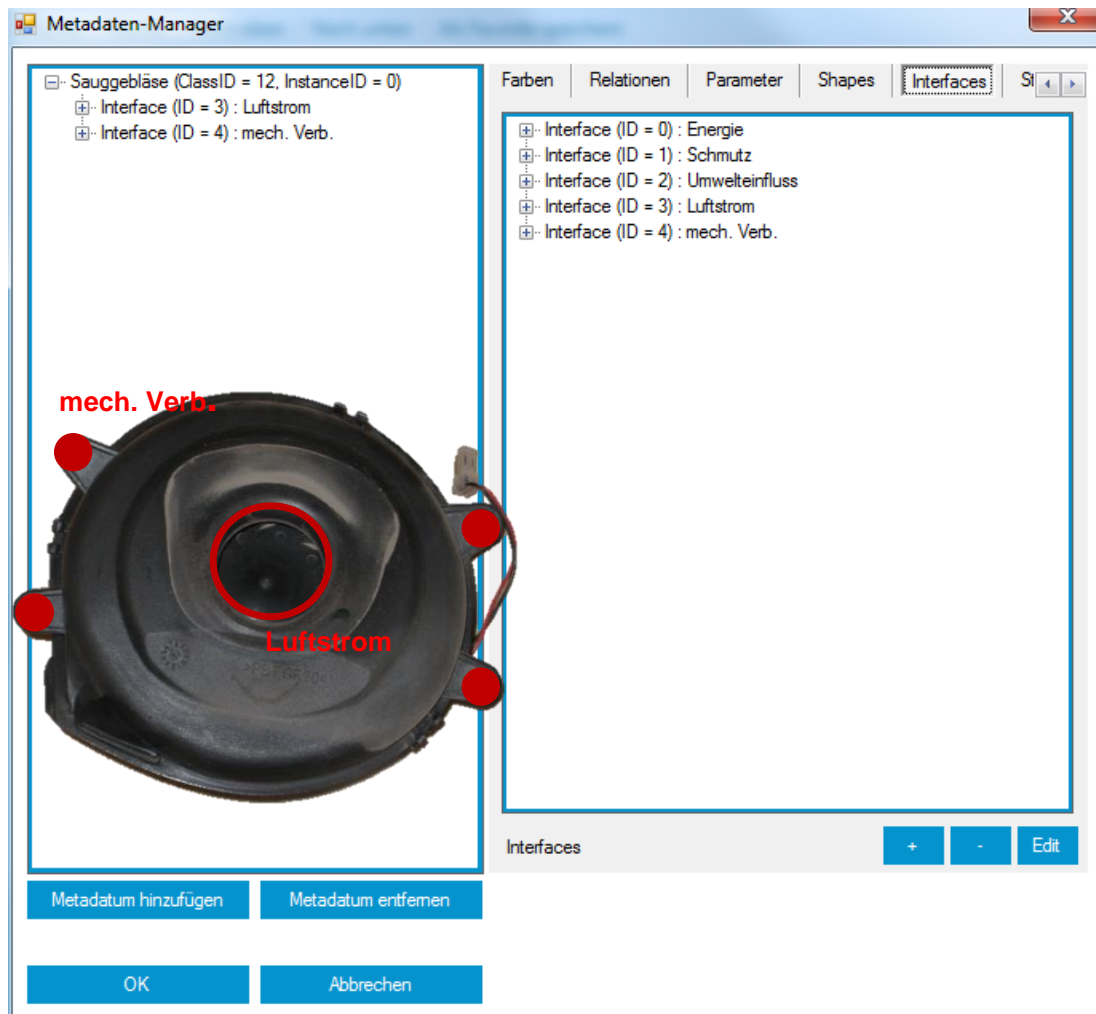


Bild 23: Metadaten Manager am Beispiel Sauggebläse. Linker Bereich: Metadaten des gewählten Systemelements. Rechter Bereich: globale Verwaltung aller Metadaten

8.9 Grafische Modellierung

Möchte der Anwender beispielsweise die Reinigungsbaugruppe näher betrachten, kann er mit dieser der grafischen Modellierungsfläche nach Bild 24 arbeiten.

Auf der grafischen Modellierungsfläche werden Modellelemente platziert und deren Schnittstellen verbunden. Es wird die Forderung nach anwenderorientierter grafischer Modellierung²⁰² aufgegriffen.

Die Herausforderung bei der Umsetzung der grafischen Modellierung liegt in dem Implementierungsaufwand einer grafischen Modellierungsfläche, auf derer alle Elemente platziert und manipuliert werden können und die auf die Eingaben der Maus des Anwenders (verschieben, ziehen, stauchen) reagiert. **Um dem Aufwand zu begegnen, ist die grafische Modellierung nur rudimentär umgesetzt worden.**

²⁰² Vgl. Kapitel 6.2.2 Anwenderorientierte grafische Modellierung

Es lassen sich ausschließlich „Komponenten“ auf der Modellierungsfläche platzieren. Eine Visualisierung von Relationen ist noch nicht realisiert worden. Schnittstellen werden nicht an den Umrissen der Systemkomponenten „gefangen“ und können frei platziert werden. Der Anwender kann jedoch einen Mechanismus im Werkzeug auslösen, der die Schnittstellen wieder an der initialen Position platziert. Eine verschachtelte Darstellung von Systemkomponenten ist nicht umgesetzt worden. Ein für den erfahrenen Anwender abschaltbarer Hierarchie-Check unterbindet daher die Platzierung verschachtelter Komponenten auf der grafischen Modellierungsfläche.

Die rudimentäre Umsetzung der grafischen Modellierung ermöglicht die Validierung der grundsätzlichen Tragfähigkeit des Ansatzes durch vereinfachte grafische Modellierung der „Systemkomponenten“. Für die Erprobung im produktiven Einsatz sollte die Einbindung kommerziell erhältlicher Programmbibliotheken in Betracht gezogen werden, die jedoch zur Implementierung auf SystemSketcher angepasst werden müssen.

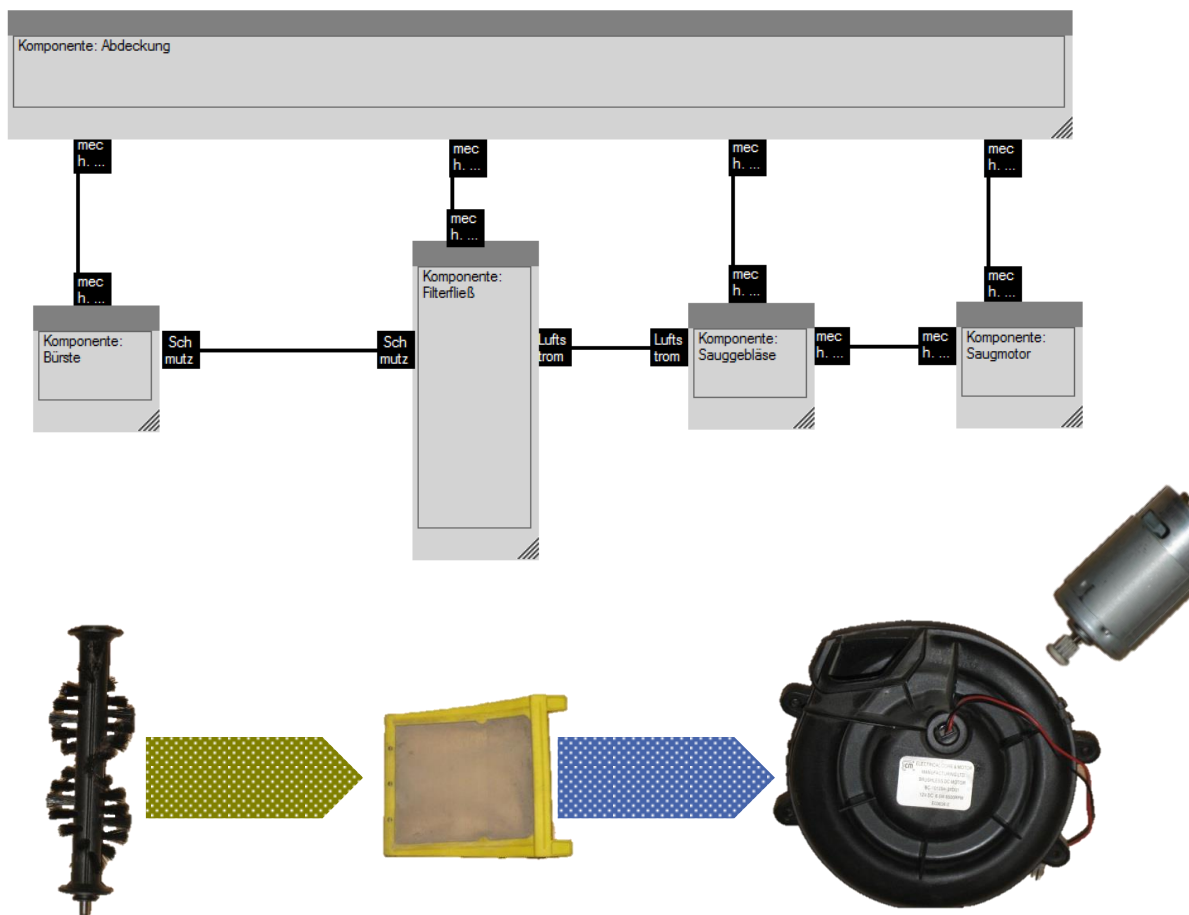


Bild 24: Grafische Modellierung der Reinigungsbaugruppe

8.10 Relationen Manager

Möchte der Entwicklungsingenieur Modellelemente übergreifend vernetzen, beispielsweise die „Funktionen“ den funktionserfüllenden „Komponenten“ zuordnen, oder überprüfen, welche Vernetzungen bereits existieren, kann er den Relationen Manager verwenden.

Der „Relationen-Manager“ nach Bild 25 stellt den modalen Dialog zum Erstellen und Verwalten der Relationen zwischen den Modellelementen dar. Er dient dem Anwender zum Sichten und Prüfen der Relationen eines Modellelements.

Die Herausforderung bei der Umsetzung der Relationen Managers liegt in der Implementierung eines Mechanismus, der die Erstellung und Verwaltung mehrerer Arten von Relationen erlaubt. **Um dem Aufwand zu begegnen, ist statt der beiden Relationen für allgemeine Beziehungen und Zuweisung²⁰³ nur diejenige der allgemeinen Beziehung realisiert worden.**

Aus Beobachtungen während erster Anwendungsversuche des Validierungsprototypen lässt sich ableiten, dass alleinig die allgemeine Relation zur Modellierung ausreichend sein kann. Es ist zu hinterfragen, ob Zuordnungen (beispielsweise Funktion zu Komponenten oder Hardware- zu Softwarekomponenten) eine separate Relation erfordern, oder die allgemeine Relation alleinig weiterhin ausreicht. Dies ermöglicht die Validierung der grundsätzlichen Tragfähigkeit des Ansatzes. Für die Erprobung im produktiven Einsatz sollten anwenderspezifische Relationen angelegt werden können.

Der Anwender wählt im „Relationen Manger“ den Relationspartner eines initial selektierten Modellelements. Er kann hier zur Auswahl des Partners auf die vorhandenen Strukturbäume aller Partialmodelle zugreifen. Entsprechend der bidirektionalen Ausprägung der Relationen werden diese auch beim Relationspartner automatisiert angelegt und angezeigt.

²⁰³ Vgl. Kapitel 7.1 Initiales Zielsystem der Sprache

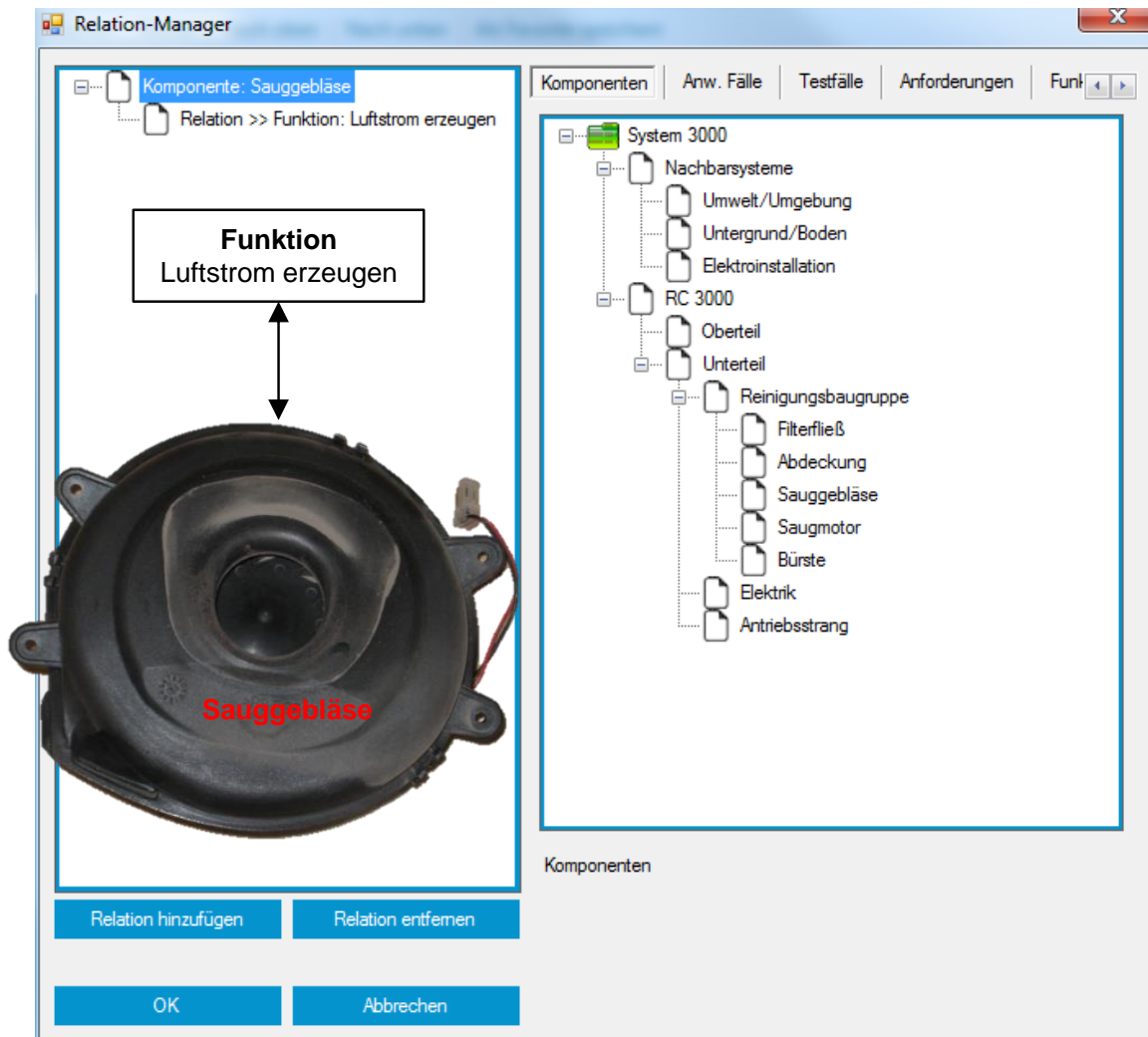


Bild 25: Relationen Manager am Beispiel Sauggebläse. Linker Bereich: Relationen des ausgewählten Sauggebläses. Rechter Bereich: Strukturbäume aller Partialmodelle zur Auswahl des Relationspartners

9 Validierung des Ansatzes

In diesem Kapitel werden Untersuchungen beschrieben, mit denen der Ansatz mit Hilfe des Validierungsprototypen²⁰⁴ auf die grundsätzliche Tragfähigkeit untersucht wird.

Das Karlsruher Lehrmodell für Produktentwicklung (KaLeP) am IPEK setzt auf die Vermittlung praxisrelevanter Kompetenzen in der akademischen Ausbildung junger Ingenieure.²⁰⁵ Neben Fachkompetenzen werden in ergänzenden Lehrveranstaltungen Sozialkompetenzen zur Entwicklungsarbeit in Teams und Problemlösungskompetenzen vermittelt. Eine dieser ergänzenden Lehrveranstaltungen ist die zur Vertiefung im Masterstudiengang angebotene „IP – Integrierte Produktentwicklung“. In IP erleben Studierende in Entwicklungsteams einen realen Entwicklungsprozess mit selbstständiger Entwicklung eines technischen Produktes von der Marktrecherche bis hin zum Prototypen in vier Monaten intensiver Projektarbeit in Vollzeit.²⁰⁶ IP findet jährlich in Kooperation mit einem Industriepartner statt, der den Entwicklungsauftrag und damit die Entwicklungsaufgabe stellt.

Die Teilnehmer sind als Anfänger zu sehen, die in IP mit einem realistischen Produktentwicklungsprozess in Berührung kommen. Durch eine begleitende Lehrveranstaltung, Seminare und Coaching werden sie an die Herausforderungen des Entwicklungsprozesses herangeführt.

IP stellt nicht nur eine Lehrveranstaltung in der universitären Ausbildung dar, sondern kann am IPEK auch als Live-Labor für die Forschung verwendet werden. Zur Validierung des Ansatzes ist diese Möglichkeit genutzt worden und mit einer Probandengruppe aus IP sind Untersuchungen am Validierungsprototyp durchgeführt worden.

²⁰⁴ Vgl. Kapitel 8 Validierungsprototyp – SystemSketcher

²⁰⁵ ALBERS ET AL. 2001

²⁰⁶ ALBERS ET AL. 2006

9.1 Studiendesign

Der **Untersuchungsgegenstand** der Studien zielt auf die Unterstützung des Anwenders in der interdisziplinären Produktentwicklung.²⁰⁷ Hierzu wird mit einem Fragenkatalog nach Bild 26 die Unterstützungsleistung in den Analyse- und Syntheseschritten des erweiterten ZHO-Modells²⁰⁸ abgefragt. Die Bewertung erfolgt durch eine 5-stufige Skala nach Likert²⁰⁹ von „trifft nicht zu“ bis „trifft voll zu“.

Weiterhin wird mit der „Systems Usability Scale“ (SUS) auf eine etablierte Methode zur Bewertung der allgemeinen Gebrauchstauglichkeit von Systemen zurückgegriffen.²¹⁰ Die SUS enthält einem Satz von 10 Fragen, die in der deutschen Übersetzung nach REINHARDT²¹¹ verwendet worden sind. Die Beantwortung der SUS-Fragen erfolgt ebenfalls durch eine 5-stufige Skala nach LIKERT. Die Einzelwertungen nach dem Schema der SUS zu einer dimensionslosen Gesamtbewertung („SUS Score“) verrechnet und mithilfe einer Bewertungsskala interpretiert.²¹⁰ Der SUS Score stellt eine dimensionslose Größe im Bereich von 0-100 dar. Ein Wert über 68 bedeutet eine überdurchschnittliche Gebrauchstauglichkeit. Mit Hilfe der von BANGOR ET AL.²¹² entwickelten Bewertungsskala kann die Bewertung differenzierter in 7 Abstufungen von „worst imaginable“ bis „best imaginable“ erfolgen. Eine Abbildung des Fragebogens kann dem Anhang dieser Arbeit entnommen werden.

Das **Studienkonzept** orientiert sich am Lehransatz der KIT-Lehrveranstaltung „Mechatronische Systeme und Produkte“, bei dem die interdisziplinäre Systemmodellierung mit SysML in kompakten Vorlesungseinheiten in Verbindung mit kompakten Übungseinheiten vermittelt wird.²¹³ Gegenstand dieser Lehrveranstaltung ist die Modellierung eines mechatronischen Systems anhand vorgegebener Elemente mit Stift und Papier. Sie dient also dazu, die Notation der SysML zu vermitteln. In den Studien zur Validierung wird die begleitete Modellierung eines mechatronischen Systems aufgegriffen. Statt mit Stift und Papier wird mit dem Validierungsprototyp modelliert. Hierzu sind Anleitungsblätter vorgesehen worden, welche unten weiter beschrieben werden. Weiterhin werden nicht alle Elemente des Systemmodells vorgegeben, sondern gemeinsam mit den Probanden in Diskussion erarbeitet. Damit soll die Bewertbarkeit der Unterstützungsleistung aus der Anwenderperspektive stimuliert werden.

²⁰⁷ Vgl. Kapitel 3.1 Motivation

²⁰⁸ Vgl. Kapitel 2.3 Erweitertes ZHO-Modell

²⁰⁹ LIKERT 1932

²¹⁰ BANGOR ET AL. 2009

²¹¹ REINHARDT 2012

²¹² BANGOR ET AL. 2009

²¹³ MATTHIESEN ET AL. 2014

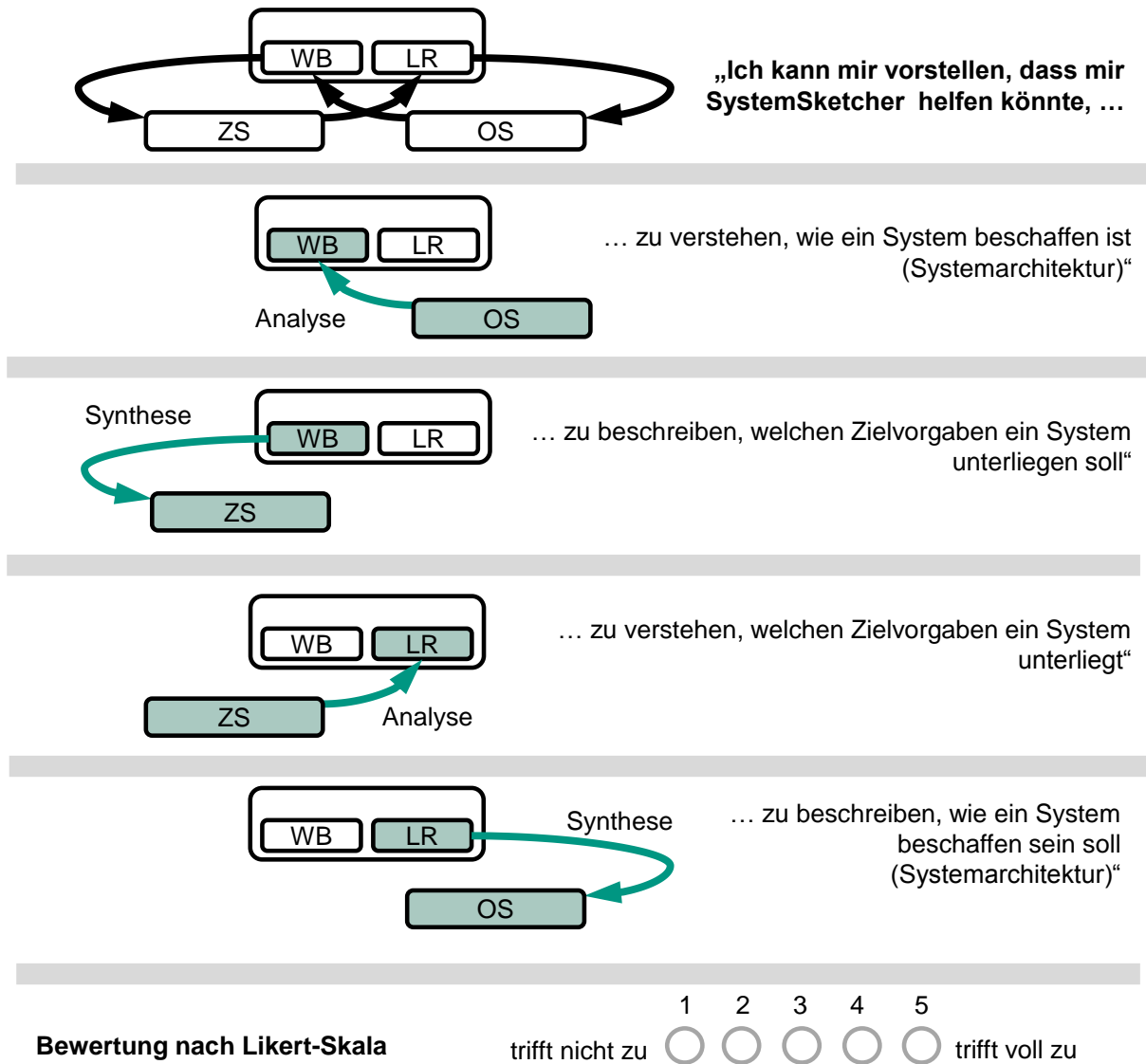


Bild 26: Erfassung der Unterstützungsleistung in Analyse- und Syntheseschritten nach dem erweiterten ZHO-Modell

Für die Erprobung sind **Probanden** aus der IPEK-Lehrveranstaltung „Integrierte Produktentwicklung“ (IP) ausgewählt worden. Die Entwicklungsarbeit in IP geschieht in intensiver Teamarbeit, wobei den Teilnehmern typische Rollen aus der realen Produktentstehung zukommen. Für die Untersuchungen sind Probanden mit den Rollen der „Systemingenieure“ und „Konstrukteure“ ausgewählt worden. Systemingenieure tragen in IP die Verantwortung des Systementwurfs unter Berücksichtigung aller systemrelevanter Anforderungen, während Konstrukteure die Lösungsfindung im Konstruktionsprozess verantworten. Alle Teilnehmer von IP sind im Rahmen begleitender Lehrveranstaltungen in Methoden der Produktentwicklung geschult. Im Rahmen der diesjährigen Aufgabenstellung sind allen Teilnehmern die Grundlagen interdisziplinärer Systemmodellierung vermittelt worden. In Vorgesprächen sind alle Probanden nochmals zu Nutzen und Zweck von Systemmodellierung sensibilisiert worden. Im Rahmen der diesjährigen

Aufgabenstellung hatte die Dokumentation von systemisch relevanten Aspekten durch die Systemingenieure in einem prototypischen Informationssystem zu erfolgen. Gegenstand dieses Informationssystems war die Abbildung systemrelevanter Aspekte wie Ziele, Randbedingungen, Funktionen, Systemkomponenten und deren Relationen in verknüpfbaren Listen und grafischer Visualisierung von Strukturbäumen. Dieses Informationssystem ist Teil eines Konzepts zur modellbasierten Unterstützung in der Produktentwicklung²¹⁴ und kann in diesem Kontext als Vorläufer einer interdisziplinären Systemmodellierung verstanden werden. Die Systemingenieure verfügen somit über grundlegende Erfahrungen zu interdisziplinärer Systemmodellierung und ersten Erfahrungen im Umgang mit einem prototypischen Informationssystem. Weiterhinkonnten unter den Teilnehmern von IP zwei gefunden werden, die durch ihre bisherige Tätigkeit als studentische Hilfskräfte eine Zusatzqualifikation der Systemmodellierung aufweisen, indem sie als studentische Hilfskräfte eine Systemmodellierung mit SysML in einem kommerziell für den produktiven Einsatz erhältlichen Modellierungswerkzeug durchgeführt haben, sowie an der Entwicklung von Modellierungsmethoden beteiligt waren.

Tabelle 15 fasst die Aufteilung der Probandengruppen und deren Kompetenzen zusammen.

Tabelle 15: Probandengruppen zur Studie mit dem Validierungsprototypen

	Gruppe 1	Gruppe 2	Gruppe 3
Anzahl Personen	6	5	2
Rollen	Systemingenieure	Konstrukteure	verschiedene
Kompetenzen	Grundlagen interdisziplinärer Systemmodellierung Modellierung in einem Informationssystem	Grundlagen interdisziplinärer Systemmodellierung	Grundlagen interdisziplinärer Systemmodellierung
Zusatzqualifikation			SysML: Modellierung und Methodenentwicklung

Die **Aufgabenstellung** der Studie umfasst den begleiteten Systementwurf eines Energiespeichersystems für die Photovoltaikanlagen eines landwirtschaftlichen Anwesens. Sie ist fachlich an den Inhalten des IP Produktentwicklungsprojekts

²¹⁴ BRAUN 2013

orientiert worden. Gegenstand der Lehrveranstaltung IP dieses Jahrgangs 2015/2016 war die Entwicklung innovativer Lösungen für Energiespeichersysteme von Elektrofahrzeugen. Alle Teilnehmer sind dadurch mit der Systemarchitektur von Energiespeichersystemen für Elektrofahrzeuge vertraut und sollen ihre Erfahrungen im Systementwurf auf die Bearbeitung der Aufgabenstellung übertragen. Den Probanden ist vermittelt worden, dass sie in ihrer fiktiven Rolle als Entwicklungsingenieure für Energiespeichersysteme der Automobilindustrie zur Erschließung neuer Marktsegmente einen Energiespeicher für landwirtschaftliche Anwesen konzipieren sollen. Die Verwendung vorhandene Systemkomponenten und Module aus der automobilen Applikation sollte angestrebt werden.

Um ein ausgeglichenes Verhältnis von Entwicklungsanteil des Energiespeichersystems und dem Erlernen der interdisziplinären Systemmodellierung zu gewährleisten, sind ein Grundgerüst und vereinzelt Elemente des Systemmodells nach Tabelle 16 vorgegeben worden. In studienbegleitenden Gruppendiskussionen ist das Grundgerüst zum Systemmodell ergänzt worden. Durch Führung der Diskussion wurde die Anzahl der neuen Elemente auf die Wesentlichen begrenzt. Im Rahmen der Studie ist das Partialmodell „Validierung“ erwähnt, aber auf dessen Modellierung verzichtet worden, um die Studie inhaltlich und zeitlich zu begrenzen.

Tabelle 16: Vorgegebene Modellinhalte

Aufgabenstellung: Hofspeicher		
Partialmodelle	vorgegebenes Gerüst	vorgegebene Elemente
Systemkontext*	Energiespeicher Nachbarsysteme	Energiespeicher Photovoltaikanlage Umrichter Thermomanagement Haus Stromnetz
Anwendungsfälle	Akteure Anwendungsfälle	
Anforderungen	Speicherkapazität Sicherheit	
Funktionsstruktur		
Systemstruktur		Zellmodul** Tragstruktur** Batterie Management System** Batterie Disconnect Unit**
* Systemkontext wird im Partialmodell "Systemstruktur" modelliert		
** Beispielhafte Module und Komponenten aus den Energiespeichersystemen für Fahrzeuge		

9.2 Durchführung der Studie

Die Durchführung der Studie ist in die folgenden Schritte gegliedert worden:

- Begrüßung und Einführung in die Aufgabenstellung
- Vorstellung der Systemmodellierung mit SystemSketcher
- Entwurf und Modellierung des Systemmodells
- Bewertung durch Fragebogen
- Abschließende Diskussion

Für die Systemmodellierung ist dem Modellierungsvorgehen des Ansatzes²¹⁵ gefolgt worden. Diese hat die Partialmodelle Systemkontext (Bestandteil des Partialmodells „Systemstruktur“), Anwendungsfälle, Anforderungen, Funktionsstruktur und Systemstruktur beinhaltet.

Die Modellierung im Prototyp ist anhand von 4 Anleitungsblättern erläutert worden. Diese Blätter haben Hinweise auf die Einschränkungen des Prototyps bezüglich des User Interfaces und der Handhabbarkeit der Modellierung enthalten. Die Probanden wurden darauf hingewiesen, diese bei der Bewertung zu berücksichtigen. Eine Abbildung der Anleitungsblätter kann dem Anhang dieser Arbeit entnommen werden. Diese Anleitungsblätter sind während der Bearbeitung des ersten Partialmodells bereitgestellt worden. Damit konnten die wesentlichen Aspekte zur Modellierung und Handhabung bereits während der Modellierung des ersten Partialmodells „Systemkontext“ vermittelt werden.

Der Ablauf der Studie folgt zeitlich und inhaltlich einer Aneinanderreihung der Partialmodelle, wobei die Teilnehmer darauf hingewiesen wurden dass eine Bearbeitung in der Praxis nicht sequenziell, sondern simultan stattfindet. Die Bearbeitung der Partialmodelle ist dem Folgendem Ablaufschema gefolgt:

- Vorstellung des Partialmodells durch den Studienleiter
- ggf. Vorstellung des vorgegebenen Gerüsts und der vorgegebenen Elemente durch den Studienleiter
- Gruppendiskussion zum Systementwurf
- Dokumentation der gefundenen Systemelemente mit Metaplantchnik durch den Studienleiter

²¹⁵ Vgl. Kapitel 7.2 Zwischenfazit

- Modellierung der Systemelemente und Vernetzung in SystemSketcher durch die Probanden

Die Diskussionen zum Systementwurf sind in der Gruppe erfolgt, die Modellierung im SystemSketcher ist jedoch von jedem Probanden individuell durchgeführt worden. Dazu sind die Probanden während der Modellierung auf separate, mit SystemSketcher ausgestattete, Arbeitsplätze vereinzelt worden. Bild 27 zeigt exemplarisch eine der Modellierungen im Validierungsprototyp mit der Visualisierung des Systemkontextes. In Summe sind 5 Nachbarsysteme und 5 Systemkomponenten des Hochvoltspeichers erstellt worden. Die grafische Modellierungsfläche zeigt das Bild des Systemkontextes, bestehend aus dem Hochvoltspeicher, den Nachbarsystemen und deren Kopplung untereinander über Gleichstrom (AC) und Wechselstrom (DC).

Die Gestaltung des Systemmodells unterliegt trotz Vorgaben der Interpretation der Modellierenden, was in Bild 28 an drei exemplarisch dargestellten Modellen durch die individuelle Anordnung auf der grafischen Modellierungsfläche ersichtlich ist.

Nach Abschluss der Systemmodellierung sind die Probanden mit einem Fragebogen befragt worden. Der Fragebogen beinhaltet die an dem erweiterten ZHO-Modell orientierten Fragen nach Bild 26 (ohne die visuelle Darstellung des ZHO), die Fragen zur SUS, sowie die abschließende Wertung von „Ich finde eine interdisziplinäre Systemmodellierung generell sinnvoll“ nach der gleichen 5-stufigen Skala nach LIKERT.²¹⁶ Die Probanden der Gruppe 3 haben ein zusätzliches Exemplar des Fragebogens im Kontext SysML ausgefüllt.

In einem abschließenden, geführten Interview sind die Belange und Einschätzungen der Probanden in offener Form erfasst worden. In der Gruppe der „Systemingenieure“ ist dabei auf Anregungen zur Verbesserung im Konkreten eingegangen worden. In der Rolle eines „Konstruktors“ steht vorwiegend die Nutzung der interdisziplinären Systemmodellierung im Vordergrund, sodass dort auf den Nutzen der Systemmodellierung im Allgemeinen eingegangen worden ist.

Gruppe 1 und Gruppe 3 haben die Studie gemeinsam durchlaufen. Die Durchführung mit Gruppe 2 hat an einem zeitnahen Termin stattgefunden. Der zeitliche Umfang für den Durchlauf einer Studie hat 3 Stunden betragen.

²¹⁶ LIKERT 1932

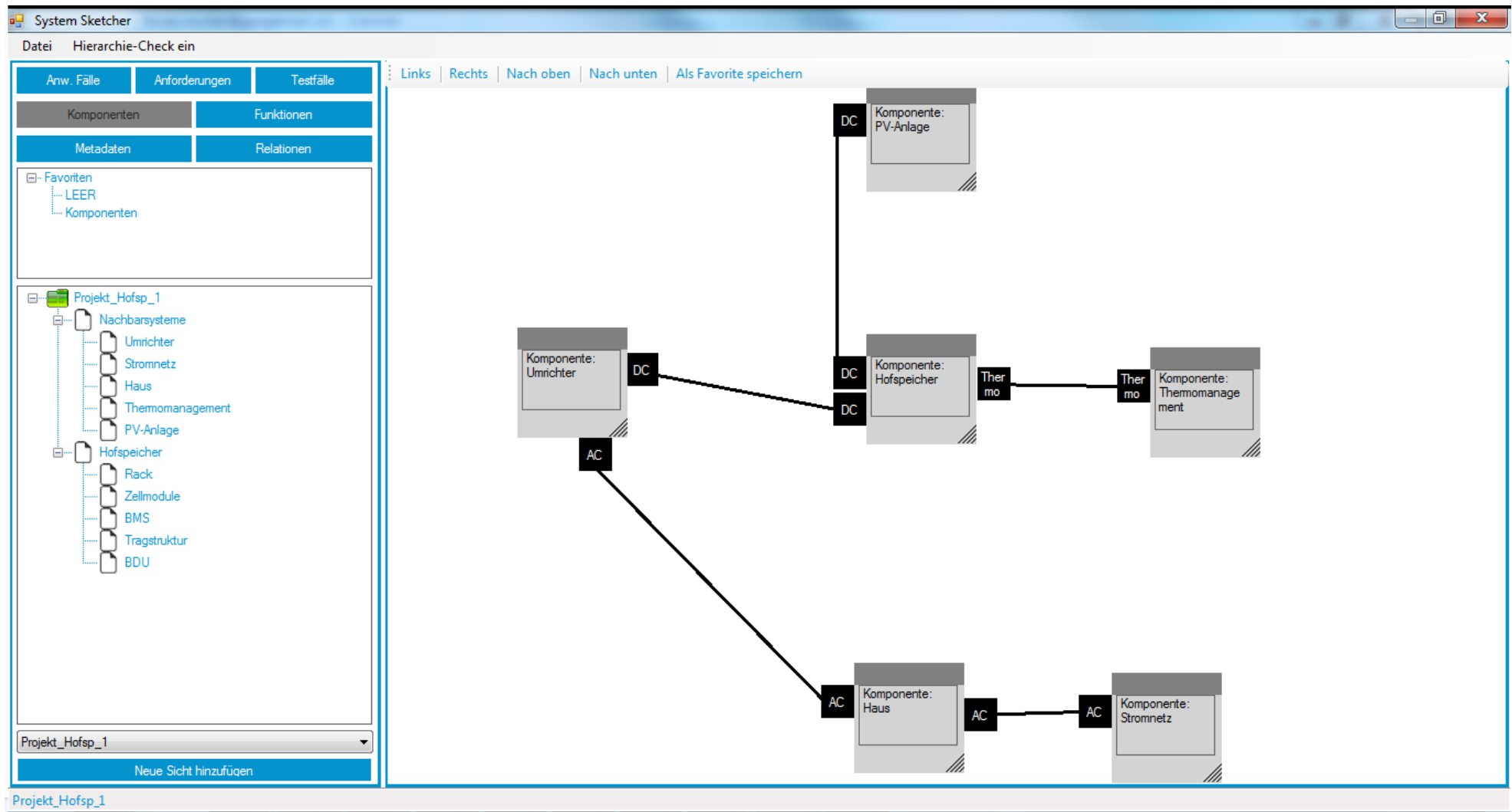


Bild 27: Modellierter Hochvoltspeicher: 5 Nachbarsysteme, 5 Systemkomponenten, Systemkontext mit DC/AC Kopplung

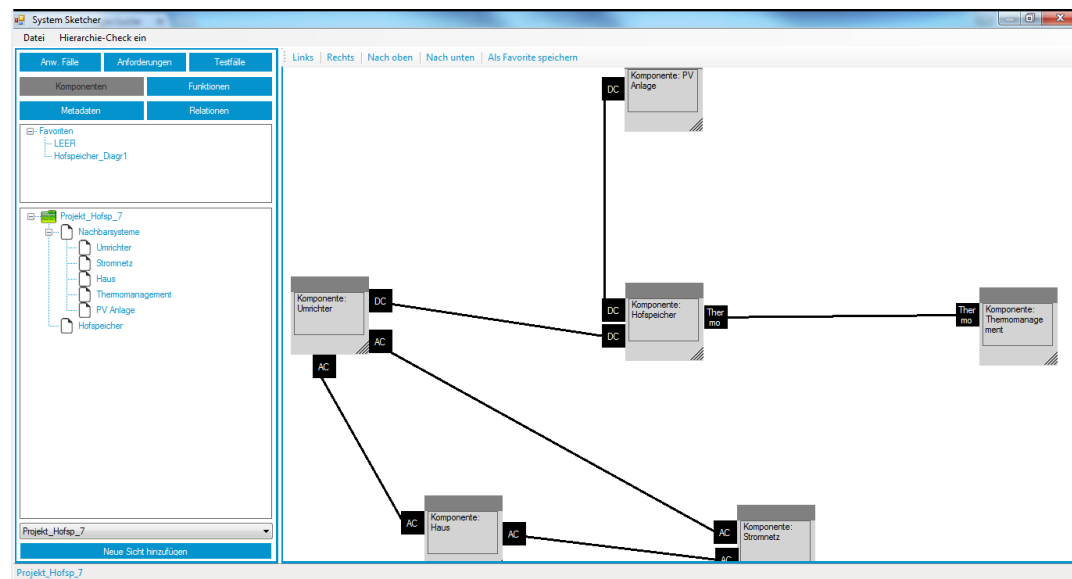
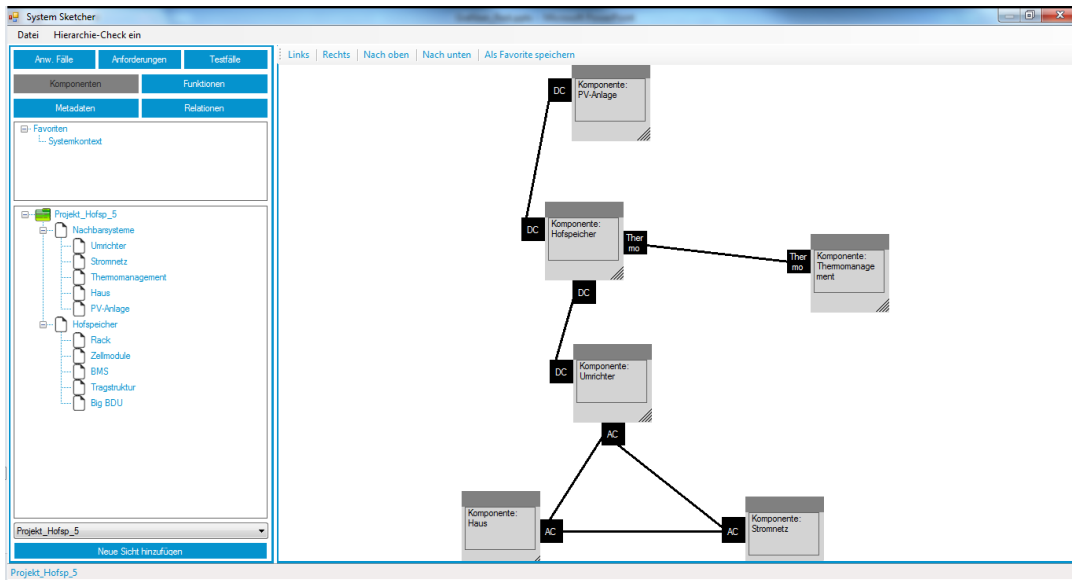
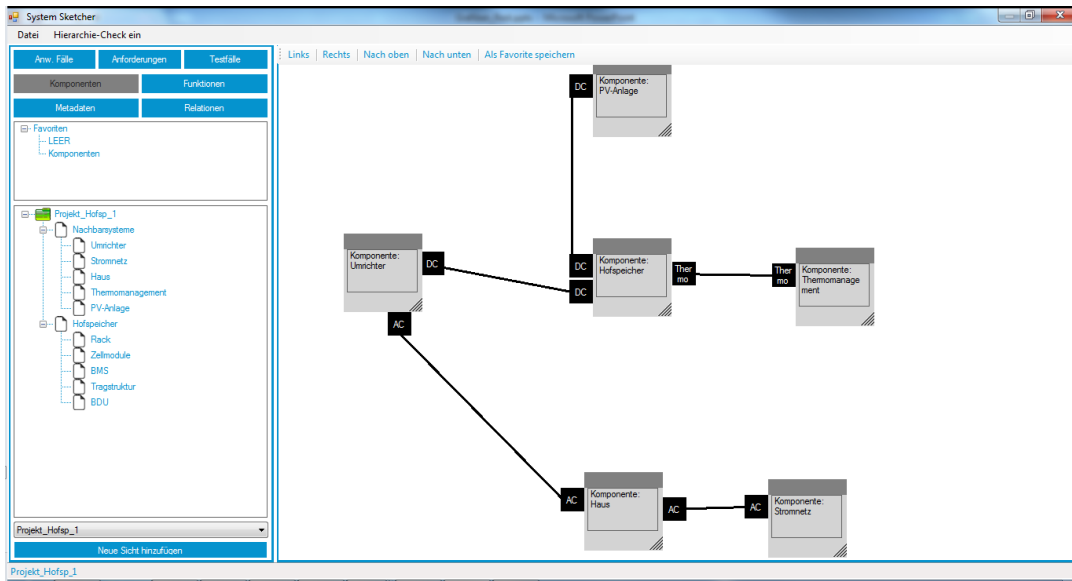


Bild 28: Individuelle Ausprägung der Systemkontextmodellierung

9.3 Studienergebnisse

Zur Bewertung der Gebrauchstauglichkeit nach SUS sind die aus den Einzelfragen errechneten dimensionslosen SUS-Werte („SUS Score“) mithilfe der SUS Bewertungsskala interpretiert worden. In Bild 29 sind die SUS Scores aller Probanden sind nach in einer Häufigkeitsverteilung im Intervall von 5 Einheiten aufgetragen und der Bewertungsskala gegenübergestellt worden. Durch die Bewertungspunkte nach SUS Score mit Standardabweichung und Darstellung im Intervall ergeben sich überlappende Bewertungsbereiche in Bild 29.

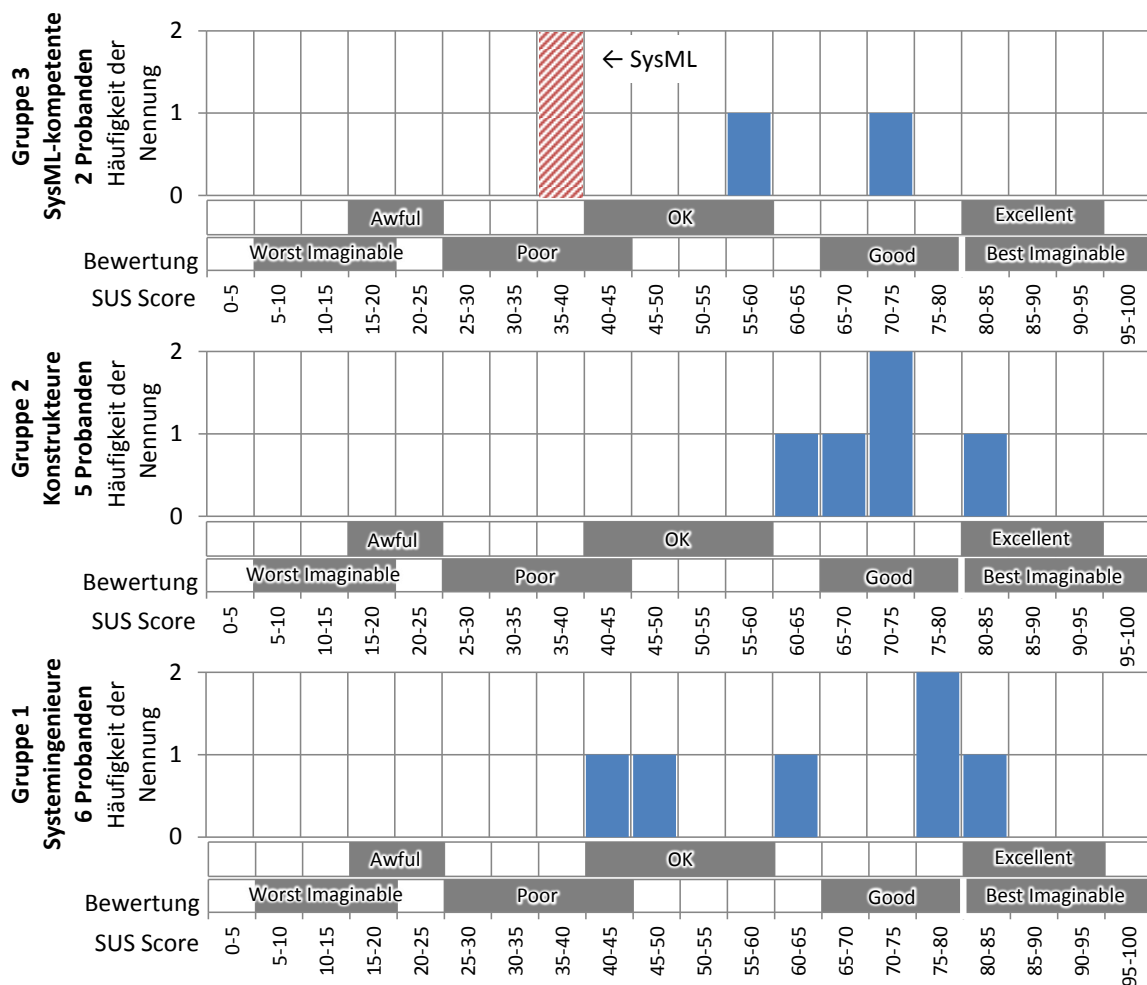


Bild 29: Häufigkeitsverteilung und Bewertung nach SUS-Score²¹⁷ (dimensionslos)
 Darstellung im Intervall über Bewertung und Score
 SystemSketcher: blaue, massive Balken; SysML: roter, schraffierter Balken

²¹⁷ BANGOR ET AL. 2009

Die **Bewertung der Gebrauchstauglichkeit nach SUS** kann wie folgt zusammenfassend beschrieben werden:

- Die Gebrauchstauglichkeit ist von der Probandengruppe „Systemingenieure“ tendenziell schlechter bewertet worden als von der Probandengruppe „Konstrukteure“.
- Die Spanne der Bewertung reicht von der unteren Grenze eines akzeptablen Bereichs (Bewertung „OK“) bis an die untere Grenze der Wertung „Exzellente“ heran.
- Die Bewertung von SysML durch die zwei Probanden der Gruppe 3 ist übereinstimmend. Sie liegt im schlechten Bereich (Bewertung „Poor“) und damit unterhalb aller Bewertungen von SystemSketcher.

Die Ergebnisse der **Bewertung der Unterstützungsleistung nach dem erweiterten ZHO-Modell** sind in Bild 30 dargestellt. Es ist die Häufigkeitsverteilung über die 5 Stufen der Bewertungsskala nach LIKERT²¹⁸ aufgetragen.

Die Bewertung nach dem erweiterten ZHO-Modell kann wie folgt zusammenfassend beschrieben werden:

- Die Probandengruppe der „Systemingenieure“ hat die Unterstützungsleistung tendenziell schlechter bewertet als die Probandengruppe „Konstrukteure“ und die Gruppe 3 „SysML-kompetente“.
- Die Bewertungen liegen tendenziell in der oberen Hälfte der Bewertungsskala.
- Die Bewertung von SysML ist nahezu identisch zu derer von SystemSketcher im oberen Bereich der Bewertungsskala.

²¹⁸ LIKERT 1932

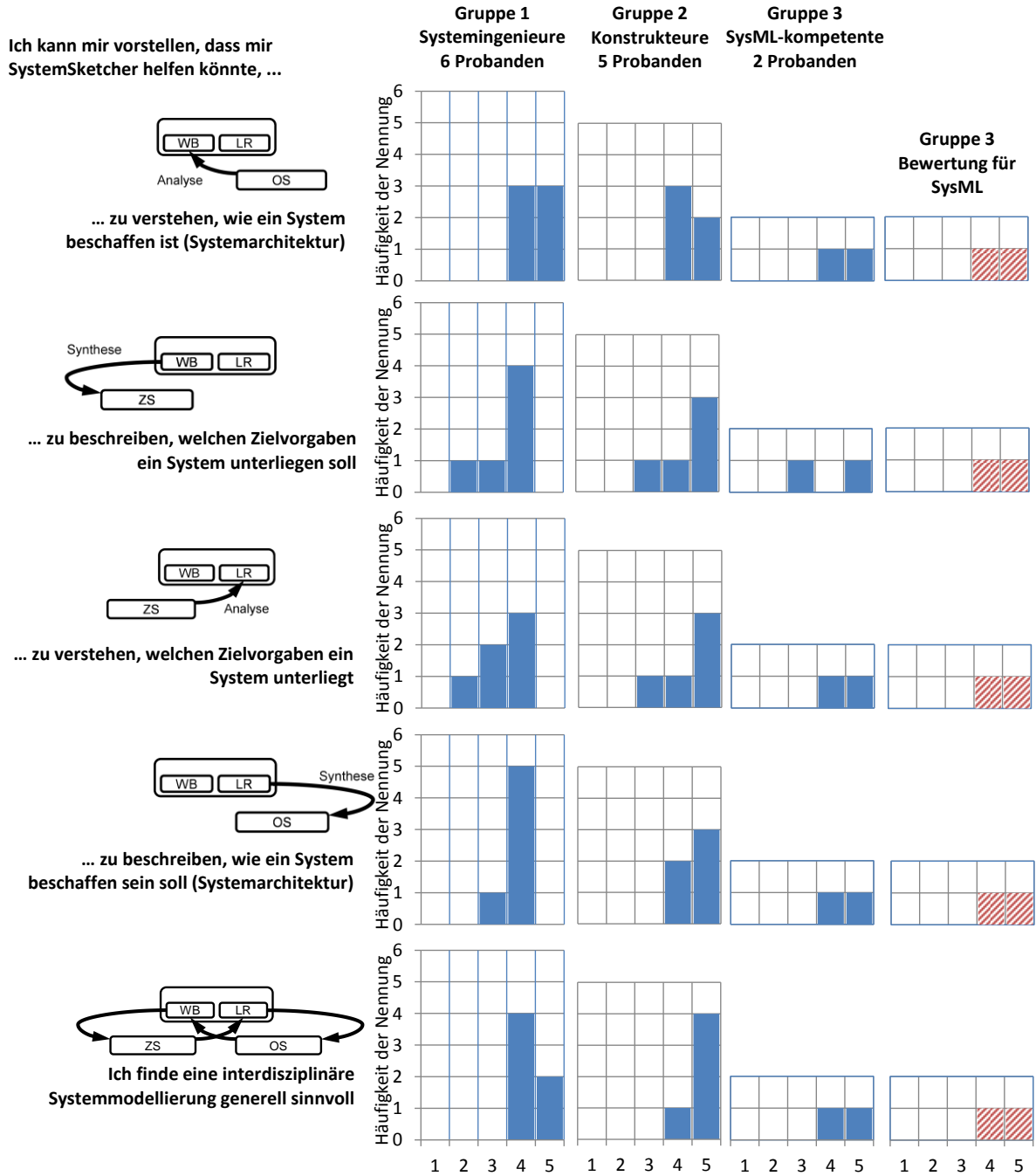


Bild 30: Bewertung von SystemSketcher (und SysML durch Gruppe 3): 5-stufige Bewertungsskala nach LIKERT²¹⁹ von 1 „trifft nicht zu“ bis 5 „trifft voll zu“ SystemSketcher: blaue, massive Balken; SysML: rote, schraffierte Balken

²¹⁹ LIKERT 1932

Die **Reaktionen im geführten Interview** im Anschluss an die Bewertung lassen sich für die Gruppe der „Systemingenieure“ und der zeitgleich befragten Gruppe 3 der „SysML-kompetenten“ zum Thema „Verbesserungsmöglichkeiten“ wie folgt zusammenfassen:

- **Bedienung:** Die Benutzeroberfläche einer Modellierungslösung wird maßgeblichen Einfluss auf das Empfinden des Anwenders haben.
- **Erweiterte Funktionalitäten:** Die Umsetzung von Varianten und Versionen, des Prinzips gekapselter Modellierung von in sich abgeschlossenen Einheiten und Auswertemechanismen wäre vorzusehen.
- **Wiederverwendbarkeit:** Hauptsächlicher Nutzen würde in einer Wiederverwendung von (Teil-) Modellen liegen. Bestehende Module könnten so in weiteren Produktgenerationen effektiv wiederverwendet werden.

Die Reaktionen der „Konstruktoren“ (Gruppe 2) lassen sich zum Thema „Nutzen“ wie folgt zusammenfassen:

- **Systemverständnis:** Der Nutzen der Systemmodellierung zum Systemverständnis durch Abbildung und Verknüpfung relevanter Systeminformationen wird erkannt und geschätzt.
- **Bedienung:** Der Nutzen ist eng verzahnt mit der Nutzbarkeit, welche durch die Gestaltung und Leistungsfähigkeit der Benutzeroberfläche bestimmt wird.

Die Reaktionen greifen Aspekte auf, die in Form der ergänzenden Randbedingungen bereits in der Spezifikation des Ansatzes wiederfinden²²⁰, aber noch nicht im Validierungsprototyp umgesetzt worden sind.

Ziel dieser Validierung des Ansatzes ist die Untersuchung der grundsätzlichen Tragfähigkeit mit Hilfe einer ersten, rudimentäre Werkzeugrealisierung. Aufgrund der zur Bearbeitbarkeit im Rahmen einer Dissertation getroffenen Einschränkungen ist davon auszugehen, dass diese Studie und die Ergebnisse nicht dazu geeignet sind, um eine vollständige Beurteilung der Usability durchzuführen. Vielmehr sind diese als Impulse zu sehen für weitere Untersuchungen im Feld der interdisziplinären Systemmodellierung.

²²⁰ Vgl. Kapitel 7 Initiales Zielsystem des Ansatzes und Kapitel 6.3 Ergänzende Randbedingungen

9.4 Zwischenfazit

Das der Motivation dieser Arbeit zugrunde liegende erweiterte ZHO-Modell ist in der Validierung aufgegriffen worden. An diesem sind Fragen zur Beurteilung der Unterstützungsleistung des Anwenders abgeleitet worden. Mit der Systems-Usability-Scale (SUS) ist eine bewährte Methode zur Bewertung der Gebrauchstauglichkeit eingesetzt worden. In einem geführten Interview sind die Probanden abschließend offen befragt worden.

Durch die geringe Anzahl der Teilnehmer sind die Ergebnisse nicht abschließend zu interpretieren. Es konnte jedoch eine Methode gezeigt werden, die sich zur Beurteilung einer Unterstützungsleistung und Gebrauchstauglichkeit von Ansätzen zur Systemmodellierung eignet.

Die Bewertungsergebnisse deuten auf eine tendenzielle Verbesserung oder Ebenbürtigkeit der Unterstützungsleistung und Gebrauchstauglichkeit im Vergleich zur Modellierung mit SysML hin. Aus Reaktionen des Interviews lässt sich Verbesserungspotenzial ableiten. Die Reaktionen bestätigen die Randbedingungen, die bereits während der Spezifikation des Ansatzes identifiziert und erfasst worden sind.

Diese Ergebnisse werden hier als Nachweis der grundsätzlichen Tragfähigkeit des Ansatzes interpretiert. Dies stellt die Beantwortung der Forschungsfrage F3²²¹ dar, und zeigt damit die grundsätzliche Anwenderakzeptanz einer semi-formalen Modellierungslösung. Dies bestätigt die Erreichung des übergeordneten Ziels,²²² eines tragfähigen Konzeptes eines neuen Ansatzes zur anwenderorientierten interdisziplinären Systemmodellierung, welches dem Anwender erlaubt, sich statt auf Formalismen der Modellierung, auf die Lösung seiner Entwicklungsaufgabe zu konzentrieren.

²²¹ Vgl. Kapitel 3.3 Forschungsfragen

²²² Vgl. Kapitel 3.2 Zielsetzung

10 Zusammenfassung und Ausblick

10.1 Zusammenfassung

Ideengebend für das Thema dieser Arbeit waren die Ergebnisse zweier Befragungen von Anwendern zur interdisziplinären Systemmodellierung mit SysML. Zwar wird das Potenzial dieser Systemmodellierung von den Befragten anerkannt und geschätzt, die Tauglichkeit in der praktischen Anwendung aber als gering eingestuft. Beobachtungen aus Anwendungsstudien bestätigen diese Einschätzung vollumfänglich.

Diese Arbeit knüpft an das menschenzentrierte Verständnis der Karlsruher Schule der Produktentwicklung an. Auf Grundlage des erweiterten ZHO Modells, einem Modell das den Menschen im Mittelpunkt der Produktentstehung begreift, wurden die grundlegenden Anforderungen für eine den Menschen unterstützende interdisziplinäre Systemmodellierung abgeleitet.

Ziel dieser Arbeit war es, einen integrativen Ansatz zu konzipieren, mit dem das Potenzial der Systemmodellierung in der praktischen Anwendung besser gehoben werden kann. Dieser Ansatz soll den Anwendern einen niederschweligen Einstieg in die interdisziplinäre Systemmodellierung bieten. Der Anwender soll sich auf die Lösung seiner Entwicklungsaufgabe konzentrieren können, statt sich mit Formalismen der Modellierung beschäftigen zu müssen. Für den Ansatz wurden bestehende Ansätze unvoreingenommen analysiert und deren Stärken neu kombiniert.

In einer Analyse sind diese bestehenden Ansätze nach ihren Aspekten Sprache, Methode und Werkzeug untersucht worden. Ergebnis dieser Analyse ist eine ansatzübergreifende Basis an charakteristischen Sprachelementen und methodischen Schritten, sowie prägnanten Eigenschaften der Werkzeuge interdisziplinärer Systemmodellierung. Ergebnis des empirisch geprägten Teils dieser Forschungsarbeit sind aus der Zielsetzung und Beobachtungen der Anwendungsstudien abgeleiteten Randbedingungen. Diese sind in den neuen Ansatz überführt worden. Dieser Syntheseprozess beinhaltet Übernahme, Neuordnung oder Verwurf von Einzelaspekten mit dem Ergebnis einer abgestimmten Gesamtlösung.

SystemSketcher stellt im Ergebnis eine erste prototyphafte Realisierung des Werkzeugs des schlanken Ansatzes dar. Der Ansatz beinhaltet eine schlanke Basis an Sprachelementen und methodischen Schritten, die vom Anwender bedarfsgerecht erweitert werden können.

Für die Untersuchung der Tragfähigkeit wesentlich relevanten Aspekte des Ansatzes sind in einem Validierungsprototypen umgesetzt worden. Mit einer Probandengruppe aus systemisch geschulten Produktentwicklern ist die grundsätzliche Tragfähigkeit des Ansatzes während der Konzeptphase einer Produktentwicklung unter Laborbedingungen validiert worden.

Hier wurde das erweiterte ZHO-Modell aufgegriffen. Es konnte eine Methode zur Bewertung der Gebrauchstauglichkeit und zur Bewertung der Unterstützungsleistung gefunden und durchgeführt werden.

Die Ergebnisse sind als erfolgreicher Nachweis der grundsätzlichen Tragfähigkeit des Ansatzes zu interpretieren. Sie bieten Impulse für weitere Untersuchungen im Feld der interdisziplinären Systemmodellierung. Der Validierungsprototyp SystemSketcher als erste, rudimentäre Werkzeugrealisierung kann ein Fundament für weiterführende Arbeiten und Werkzeugentwicklungen bieten.

10.2 Ausblick

Zukünftiger Forschungsbedarf würde in einem Bereich bestehen, der über die Validierung der grundsätzlichen Tragfähigkeit hinaus geht. Sie könnten weitergehende Untersuchungen zur Anwendbarkeit und Nutzen interdisziplinärer Systemmodellierung beinhalten. Der Nutzen einer interdisziplinären Systemmodellierung wird dabei mit der Leistungsfähigkeit des zur Modellierung verwendeten Ansatzes verzahnt sein. Vor diesem Hintergrund sollten Beobachtungen aus diesen Untersuchungen differenziert ausgewertet werden, können aber bei Berücksichtigung dieser Rahmenbedingung an beliebigen Ansätzen wie beispielsweise SysML durchgeführt werden. Forschungsergebnisse aus diesem Bereich könnten helfen, dass der Nutzen interdisziplinärer Systemmodellierung nicht nur anerkannt und geschätzt, sondern auch für Entwickler und weitere Beteiligte in der Produktentwicklung belegbar und quantifizierbar sein wird.

Für weitere Untersuchungen sollte die Weiterentwicklung einer Werkzeugrealisierung in Betracht gezogen werden. Unter Berücksichtigung des hohen Implementierungsaufwandes der ersten Werkzeugrealisierung ist für die Weiterentwicklung von einer hohen Komplexitätsstufe auszugehen. Richtwerte für die Kosten der Entwicklungskosten einer komplexen Applikation können auch im iBusiness Honorarleitfaden²²³ nicht exakt angegeben werden, sondern werden in Form einer Spanne gelistet, die typischerweise von minimal 6.000€, über durchschnittlich 79.000€, bis hin zu maximal 520.000€ reicht.

²²³ GRAF 2015

Weiterer Forschungsbedarf würde in der Ausgestaltung verbesserter Mechanismen zur fraktalen Modellierung bestehen. Mit einbezogen in die fraktale Modellierung sind neben der verschachtelten Darstellung von Systemkomponenten nach gegenwärtigem Stand auch Konzepte zum fraktalen Umgang mit Schnittstellen, beispielsweise mit Proxy- oder Nested Ports in der SysML. Es kann evaluiert werden, ob gegenwärtige Mechanismen der fraktalen Modellierung zur Abbildung eines großen Modellumfangs geeignet sind. Die Einbindung in die Prozess- und Werkzeuglandschaft der Produktentstehung könnte einen weiteren wesentlichen Faktor zur Nutzbarkeit der interdisziplinären Systemmodellierung darstellen. Aktuelle Forschungsarbeiten am IPEK beschäftigen sich beispielsweise mit der Anbindung zu CAD Lösungen.²²⁴

²²⁴ MOESER ET AL. 2015

11 Literaturverzeichnis

- ALBERS, A. ; DENKENA, B. ; MATTHIESEN, S. (eds.) **Albers et al. 2012 a** ACATECH — DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN: *Faszination Konstruktion, acatech Studie*. vol. September 2012. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012
- Albers 2010** ALBERS, A.: Five Hypotheses about Engineering Processes and their Consequences. In: *International Symposium on Tools and Methods of Competitive Engineering (TMCE)*. Ancona, Italy, 2010
- LINDEMANN, U. (ed.) **Albers et al. 2016 a** ALBERS, A. ; BEHRENDT, M. ; KLINGLER, S. ; MATROS, K. ; LINDEMANN, U. (ed.): *Verifikation und Validierung im Produktentstehungsprozess, Handbuch Produktentwicklung*. München: Carl Hanser Verlag, 2016
- Albers et al. 2013 a** ALBERS, A. ; BEHRENDT, M. ; SCHROETER, J. ; ET AL.: X-in-the-Loop: A Framework for Supporting Central Engineering Activities and Contracting Complexity in Product Engineering Processes. In: *Proceedings of the International Conference on Engineering Design, ICED, 2013*
- Albers & Braun 2011** ALBERS, A. ; BRAUN, A.: A generalized framework to compass and to support complex product engineering processes. In: *International Journal of Product Development* vol. 15, 2011
- Albers et al. 2001** ALBERS, A. ; BURKARDT, N. ; MATTHIESEN, S.: New education concepts for the training of creative engineers – The Karlsruhe education model for industrial product development – KaLeP. In: *23rd SEED Annual Design Conference and 8th National Conference on Product Design Education, 12 July 2001, Derby, United Kingdom*
- Albers et al. 2006** ALBERS, A. ; BURKARDT, N. ; MEBOLDT, M.: The Karlsruhe Education Model for Product Development “KaLeP” in Higher Education. In: *International Design Conference*. Dubrovnik, 2006
- Albers et al. 2016 b** ALBERS, A. ; BURSAC, N. ; RAPP, S.: PGE – Product Generation Engineering – Case Study of the Dual Mass Flywheel. In: *International Design Conference, Dubrovnik, 2016*
- BINZ, B. (ed.) **Albers et al. 2015 a** ALBERS, A. ; BURSAC, N. ; WINTERGERST, E.: Produktgenerationsentwicklung – Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In: BINZ, B. (ed.) *Stuttgarter Symposium für Produktentwicklung SSP 2015*
- ANDERL, R. ; EIGNER, M. ; SENDLER, U. ; STARK, R. (eds.) **Albers & Gausemeier 2012** ALBERS, A. ; GAUSEMEIER, J. ; ACATECH - DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN, 2012 ; ANDERL, R. ; EIGNER, M. ; SENDLER, U. ; STARK, R. (eds.): *Smart Engineering, acatech DISKUSSION*. vol. April 2012. Springer Verlag Berlin Heidelberg, 2012
- Albers et al. 2013 b** ALBERS, A. ; KLINGLER, S. ; EBEL, B.: Modeling Systems of Objectives in Engineering Design Practice. In: *Proceedings of the International Conference on Engineering Design, ICED, 2013*
- Albers et al. 2016 c** ALBERS, A. ; KURRLE, A. ; KLINGLER, S.: The Connected Car – A system-of-systems:Exploration of challenges in development from experts view. In: *16. Internationales Stuttgarter Symposium Automobil- und Motorentechnik 2016*
- Albers & Lohmeyer 2012** ALBERS, A. ; LOHMEYER, Q.: Advanced systems engineering – towards a model-based and human-centered methodology. In: *9th International Symposium on Tools and Methods of Competitive Engineering TMCE 2012, Karlsruhe, Germany*

- Albers et al. 2011** ALBERS, A. ; LOHMEYER, Q. ; EBEL, B.: Dimensions of objectives in interdisciplinary product development projects. In: *International Conference on Engineering Design (ICED)*. Lyngby/Copenhagen, Denmark, 2011
- Albers et al. 2012 b** ALBERS, A. ; LOHMEYER, Q. ; RADIMERSKY, A.: Individuelle und organisatorische Akzeptanz von Methoden des Systems Engineering. In: *Tag des Systems Engineering (TdSE)*. Paderborn, 2012
- BIRKHOFFER (ed.) **Albers & Matthiesen 2002** ALBERS, A. ; MATTHIESEN, S.: Konstruktionsmethodisches Grundmodell zum Zusammenhang von Gestalt und Funktion technischer Systeme. In: BIRKHOFFER (ed.) *Konstruktion – Zeitschrift für Produktentwicklung* vol. 54, 2002
- MAURER, M. ; SCHULZE, S.-O. (eds.) **Albers et al. 2014** ALBERS, A. ; MATTHIESEN, S. ; BURSAC, N. ; ET AL.: Abstraktionsgrade der Systemmodellierung – von der Sprache zur Anwendung. In: MAURER, M. ; SCHULZE, S.-O. (eds.) *Tag des Systems Engineering 2014*, Carl Hanser Verlag
- KOHLHAMMER, K. (ed.) **Albers et al. 2016 d** ALBERS, A. ; MATTHIESEN, S. ; BURSAC, N. ; ET AL.: Model-Based Systems Engineering (MBSE) in der Karlsruher Schule: Fünf Jahre Forschung für die Anwendung. In: KOHLHAMMER, K. (ed.) *develop³ systems engineering* (2016), Nr. 01 2016
- Albers et al. 2013 c** ALBERS, A. ; MUNKER, F. ; ZINGEL, C. ; BEHRENDT, M.: Integrative Systemmodellierung von Hardware- und Softwarekomponenten in SysML am Beispiel eines innovativen Datengateways. In: *Tag des Systems Engineering (TdSE)* Stuttgart, 2013
- Albers et al. 2016 e** ALBERS, A. ; REIS S, N. ; BURSAC, N. ; RICHTER, T.: iPeM – integrated Product engineering Model in context of Product Generation Engineering. In: *CIRP Design 2016*. Stockholm, 2016
- CHAKRABARTI, A. ; BLESSING, L. T. M. (eds.) **Albers & Sadowski 2014** ALBERS, A. ; SADOWSKI, E.: An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. In: CHAKRABARTI, A. ; BLESSING, L. T. M. (eds.) *The Contact and Channel Approach (C&C²-A) – relating a system's physical structure to its functionality* : Springer Science & Business Media, 2014
- Albers et al. 2015 b** ALBERS, A. ; SCHERER, H. ; BURSAC, N. ; RACHENKOVA, G.: Model Based Systems Engineering in Construction Kit Development – Two Case Studies. In: *CIRP Design Conference*. Cape Town, South Africa, 2015
- Albers et al. 2016 f** ALBERS, A. ; WALCH, M. ; BURSAC, N.: Entscheidungsunterstützung durch die Variationsanteile der Produktgenerationsentwicklung. In: *Konstruktion* (2016), Nr. 4
- Albers & Zingel 2013** ALBERS, A. ; ZINGEL, C.: Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML. In: *CIRP Design Conference*. Bochum, Germany, 2013
- Bangor et al. 2009** BANGOR, A. ; KORTUM, P. ; MILLER, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. In: *Journal of usability studies* vol. 4 (2009), Nr. 3
- Bergenthal 2011** BERGENTHAL, J.: *Final Report of Model Based Engineering (MBE) Subcommittee* : NDIA Systems Engineering Division M&S Committee, 2011
- Blessing & Chakrabarti 2009** BLESSING, L. T. M. ; CHAKRABARTI, A.: *DRM, a design research methodology*. Dordrecht; New York, Springer Verlag, 2009
- Bone & Cloutier 2010** BONE, M. ; CLOUTIER, R.: The Current State of Model Based Systems Engineering: Results from the OMGTM SysML Request for Information 2009. In: *Conference on Systems Engineering Research (CSER)*. Hoboken, NJ, USA, 2010
- Braun 2013** BRAUN, A.: *Modellbasierte Unterstützung der Produktentwicklung – Potentiale der Modellierung von Produktentstehungsprozessen am Beispiel des integrierten Produktentstehungsmodells (iPeM)*. IPEK Forschungsberichte Band 72, Karlsruhe, 2013

- Cantamessa 2003** CANTAMESSA, M.: An empirical perspective upon design research. In: *Journal of Engineering Design* vol. 14 (2003), Nr. 1
- MAURER, M. ; SCHULZE, S.-O. (eds.) **Dänzer et al. 2014** DÄNZER, M. ; KLEINER, S. ; LAMM, JESKO G. ; ET AL.: Funktionale Systemmodellierung nach der FAS-Methode: Auswertung von vier Industrieprojekten. In: MAURER, M. ; SCHULZE, S.-O. (eds.) *Tag des Systems Engineering (TdSE)*. Stuttgart, 2014
- Dassault 2016** DASSAULT: Catia Systems Engineering - Dymola, Dassault Systèmes Société Européenne, Produktdatenblatt auf <http://www.3ds.com/productsservices/catia/products/dymola>, abgerufen am 4.2.2016
- Delligatti et al. 2014** DELDIGATTI, L. ; STEINER, R. ; SOLEY, R.: *SysML distilled: a brief guide to the systems modeling language*. Upper Saddle River, NJ : Addison-Wesley, 2014
- Denger et al. 2012** DENGGER, A. ; FRITZ, J. ; KISSEL, M. ; ET AL.: Potentiale einer funktionsorientierten Lenkung mechatronischer Produkte in der Automobilindustrie. In: *Tag des Systems Engineering (TdSE)*. Paderborn, Germany, 2012
- Dori 2011** DORI, D.: *Object-process methodology: A holistic systems paradigm* : Springer Science & Business Media, 2011
- von Dungern:** Semantic Model Integration for System Specification, Whitepaper erhältlich auf www.reqif.de, abgerufen am 29.1.2016
- Ebel 2015** EBEL, B.: *Modellierung von Zielsystemen in der interdisziplinären Produktentstehung, IPEK Forschungsberichte*, Band 85, Karlsruhe, Germany, 2015
- Eigner et al. 2015** EIGNER, M. ; DICKOPF, T. ; SCHULTE, T. ; SCHNEIDER, M.: mecPro²-Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML. In: *Tag des Systems Engineering (TdSE)*. Ulm, Deutschland, 2015
- Eigner & Gilz 2012** EIGNER, M. ; GILZ, T.: Proposal for Functional Product Description as Part of a PLM Solution in Interdisciplinary Product Development. In: *International Design Conference (DESIGN)*. Dubrovnik, Croatia, 2012
- Estefan 2007** ESTEFAN, J. A.: Survey of model-based systems engineering (MBSE) methodologies, INCOSE MBSE Focus Group (2007)
- Frank 2006** FRANK, U. ; GAUSEMEIER, J.: *Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme, HNI-Verlagsschriftenreihe*. vol. 175. Universität Paderborn, 2006
- Friedenthal et al. 2012** FRIEDENTHAL, S. ; MOORE, A. ; STEINER, R.: *A practical guide to SysML: the systems modeling language*. Waltham, MA : Morgan Kaufmann, 2012
- PROF. DR.-ING. MARTIN EIGNER (ed.) **Gilz 2014** GILZ, T. ; PROF. DR.-ING. MARTIN EIGNER (ed.): *PLM-Integrated Interdisciplinary System Models in the Conceptual Design Phase Based on Model-Based Systems Engineering, Schriftenreihe VPE*. vol. 13, 2014
- Gloger 2016** GLOGER, B.: *Scrum: Produkte zuverlässig und schnell entwickeln*. 5., überarbeitete Auflage. ed. München : Hanser, 2016.
- Graf 2015** GRAF, J.: *iBusiness Honorarleitfaden: Honorare und Produktionskosten bei der Entwicklung interaktiver Anwendungen in Deutschland, Österreich und der Schweiz; Leitfaden für interaktive Medien*. München : HighText-Verlag, 2015.
- Hoffmann 2014** HOFFMANN, H.-P.: Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering Deskbook Release 4.1, IBM Corporation (2014)
- INCOSE SE Vision Project Team 2014** INCOSE SE VISION PROJECT TEAM: A World in Motion.

Systems Engineering Vision 2025, INCOSE (2014)

- Iwanek et al. 2013** IWANEK, P. ; KAISER, L. ; DUMITRESCU, R. ; NYSSSEN, A.: Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS: Modellbasierte Systementwicklung. In: *Tag des Systems Engineering (TdSE)*, 2013
- ABRAMOVICI, M. ; STARK, R. (eds.) **Kleiner & Kramer 2013** KLEINER, S. ; KRAMER, C.: Model Based Design with Systems Engineering based on RFLP Using V6. In: ABRAMOVICI, M. ; STARK, R. (eds.) *CIRP Design Conference*. vol. Smart Product Engineering. Bochum, Germany : Springer, 2013
- Lamm & Weilkiens 2010** LAMM, J. G. ; WEILKIENS, T.: Funktionale Architekturen in SysML. In: *Tag des Systems Engineering (TdSE)*. München, 2010
- Likert 1932** LIKERT, R.: A technique for the measurement of attitudes. In: *Archives of psychology* (1932)
- Lohmeyer 2013** LOHMEYER, Q.: *Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme*, IPEK Forschungsberichte, Band 59, 2013
- Marxen 2014** MARXEN, L.: *A Framework for Design Support Development based on the integrated Product Engineering Model iPeM*. IPEK Forschungsberichte Band 7, Karlsruhe, 2014
- Marxen & Albers 2012** MARXEN, L. ; ALBERS, A.: Supporting validation in the development of Design methods. In: *International Design Conference (DESIGN)*. Dubrovnik, Croatia, 2012
- Matthiesen 2002** MATTHIESEN, S.: *Ein Beitrag zur Basisdefinition des Elementmodells“ Wirkflächenpaare & Leitstützstrukturen“ zum Zusammenhang von Funktion und Gestalt technischer Systeme*, IPEK Forschungsberichte. Band 6, Karlsruhe, 2002
- Matthiesen et al. 2014** MATTHIESEN, S. ; SCHMIDT, S. ; MOESER, G. ; MUNKER, F.: The Karlsruhe SysKIT Approach – A Three-Step SysML Teaching Approach for Mechatronic Students. In: *CIRP Design Conference*, 2014
- DIETER KRAUSE, K. P. (ed.) **Moeser et al. 2015** MOESER, G. ; ALBERS, A. ; KLINGLER, S.: Modellbasierte Funktions-Gestalt-Synthese: Unterstützung der Entwickler bei der Definition der Produktgestalt. In: DIETER KRAUSE, K. P. (ed.) *26. Design for X Symposium*. Hamburg : TuTech Verlag, 2015
- Munker & Albers 2015** MUNKER, F. ; ALBERS, A.: SystemSketcher – Entstehung eines anwenderorientierten Ansatzes zur interdisziplinären Systemmodellierung. In *Tag des Systems Engineering (TdSE)*. Ulm, Deutschland, 2015
- Munker et al. 2014 a** MUNKER, F. ; ALBERS, A. ; WAGNER, D. ; BEHRENDT, M.: Multi-View Modeling in SysML: Thematic Structuring for Multiple Thematic Views. In: *Procedia Computer Science* (2014), Conference on Systems Engineering Research (CSER)
- Munker et al. 2015** MUNKER, F. ; BEHRENDT, M. ; ALBERS, A.: Abschlussbericht Projekt ELISE - Autonome Ladeeinheit und Systemintegrierter Daten-Gateway für Elektrofahrzeuge. Teilvorhaben MeValDi - Methodenforschung und Validierung. Technische Informationsbibliothek TIB, Hannover (2015)
- Munker et al. 2014 b** MUNKER, F. ; KOTSCHENREUTHER, T. ; BEHRENDT, M. ; RUPALLA, A.: ELISE – Ladeeinheit und Datengateway für Elektrofahrzeuge. In: *Automobiltechnische Zeitschrift*, Sept 2014, Ausgabe "Elektromobilität"
- No Magic 2012** NO MAGIC: Introducing Cameo Systems Modeler. *Pressemitteilung* auf www.nomagic.com, abgerufen am 29.1.2016
- PTC 2014** PTC: Data Sheet PTC Integrity Modeler, PTC Inc. (2014). Erhältlich auf www.ptc.com,

abgerufen am 4.2.2016

- Reinhardt 2012** REINHARDT, W.: Crowdsourcing the translation of SUS. Blogbeitrag auf www.itsjustme.de, abgerufen am 31.1.2016
- Ropohl 1975** ROPOHL, G.: Systemtechnik – Grundlagen und Anwendung. Carl Hanser Verlag, 1975,
- Ropohl 2009** ROPOHL, G.: *Allgemeine Technologie – Eine Systemtheorie der Technik*. Karlsruhe, Deutschland : Universitätsverlag Karlsruhe, 2009
- Scherer 2016** SCHERER, H.: Modellbasierte Methoden zur Modellierung des Zielsystems und des Funktions-Gestalt- Zusammenhangs zur Unterstützung der Serienentwicklung von Baukästen am Beispiel von Hybrid-Triebstrangsystemen, IPEK Forschungsbericht, Karlsruhe, 2016
- Sparx 2015** SPARX: Enterprise Architect 12.1 - Ultimate Modeling/Lifecycle Platform, Sparx Systems Pty Ltd. (2015), Whitepaper auf www.sparxsystems.com, abgerufen am 4.2.2016
- Vitech 2011** VITECH: CORE 8 Systems Engineering Guided Tour, Vitech Corporation (2011), Whitepaper auf www.vitechcorp.com, abgerufen am 23.6.2014
- Vitech 2013 a** VITECH: CORE 9 Systems Definition Guide, Vitech Corporation (2013), Whitepaper auf www.vitechcorp.com, abgerufen am 23.6.2014
- Vitech 2013 b** VITECH: CORE 9 Architecture Definition Guide, Vitech Corporation (2013), Whitepaper auf www.vitechcorp.com, abgerufen am 23.6.2014
- Wagner 2015** WAGNER, D.: *Methodengestützte Entwicklung eines elektrischen Energiespeichers zur Erschließung von Leichtbaupotenzialen als Beitrag zur Produktgenerationsentwicklung*, IPEK Forschungsberichte, Karlsruhe, 2015
- Wagner et al. 2013** WAGNER, D. ; ALBERS, A. ; KLINGLER, S. ; ET AL.: Next Generation Electric Energy Storage Systems – Systematic Design Approaches. In: *Grazer Symposium Virtuelles Fahrzeug*. Graz, Österreich, 2013
- (INCOSE), I. C. ON S. E. (ed.) **Walden et al. 2015** WALDEN, D. ; ROEDLER, G. ; FORSBERG, K. ; ET AL. ; (INCOSE), I. C. ON S. E. (ed.): *Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities*. vol. Fourth Edition, 2015
- Weilkiens 2008** WEILKIENS, T.: *Systems Engineering mit SysML/UML – Modellierung, Analyse, Design*, dpunkt Verlag, 2008
- Winzer 2013** WINZER, P.: Das Systems Engineering (SE) – altes Denken in neuem Gewand. In: *Generic Systems Engineering: Ein methodischer Ansatz zur Komplexitätsbewältigung* : Springer Berlin Heidelberg, 2013.
- Zingel 2013** ZINGEL, C.: *Basisdefinition einer gemeinsamen Sprache der Produktentwicklung im Kontext der Modellbildung technischer Systeme und einer Modellierungstechnik für Zielsystem und Objektsystem technischer Systeme in SysML auf Grundlage des ZHO-Prinzips*, IPEK Forschungsberichte. Band 70, 2013
- HABERFELLNER, R. ; DE WECK, O. ; FRICKE, E. ; VÖSSNER, S. (eds.) **Haberfellner et al. 2012** *Systems Engineering Grundlagen und Anwendung*. 12. ed. Zürich : Orell Füssli AG, 2012
- OMG (ed.) **OMG 2012** OMG Systems Modeling Language (OMG SysML) Version 1.3, OMG (2012), erhältlich auf www.omg.org
- OMG (ed.) **OMG 2015** OMG Systems Modeling Language (OMG SysML) Version 1.4, OMG (2015), erhältlich auf www.omg.org

12 Anhang: Analyse der Sprachräume

Tabelle 17: Sprachenanalyse, Entitäten mit abstrahierter Bedeutung

Nr.	Entität	Gruppe	Bedeutung	Sprache
1	Partialmod. Anforderungen	Anforderungen	Partialmodell Anforderungen	CONSENS
2	Partialmod. Zielsystem	Anforderungen	Partialmodell Zielsystem	CONSENS
3	Ziel	Anforderungen	Ziel	CONSENS
4	Requirement	Anforderungen	Requirement	Harmony SE
5	Requirement	Anforderungen	Requirement	RFLP
6	Requirement	Anforderungen	Requirement	SysML
7	Requirement	Anforderungen	Requirement	SysML Lite
8	Category	Anforderungen	Kategorisierung	VITEC
9	DefinedTerm	Anforderungen	Begriffsdefinition	VITEC
10	Document	Anforderungen	Quelldokument	VITEC
11	ExternalFile	Anforderungen	Quelldatei	VITEC
12	Organization	Anforderungen	Organisationseinheit	VITEC
13	ProgramActivity	Anforderungen	ProgramActivity	VITEC
14	ProgramElement	Anforderungen	ProgramElement	VITEC
15	Requirement	Anforderungen	Requirement	VITEC
16	Risk	Anforderungen	schwerwiegendes Problem	VITEC
17	Partialmod. Anwendungsszenarien	Anwendungsfälle	Partialmodell Anwendungsszenarien	CONSENS
18	Actor	Anwendungsfälle	Actor	Harmony SE
19	UseCase	Anwendungsfälle	UseCase	Harmony SE
20	Actor	Anwendungsfälle	Actor	SysML
21	Subject	Anwendungsfälle	System Element	SysML
22	UseCase	Anwendungsfälle	UseCase	SysML
23	Component Type ExternalSystem	Anwendungsfälle	Technical Neighbor System	VITEC
24	Component Type Human	Anwendungsfälle	Actor	VITEC
25	UseCase	Anwendungsfälle	UseCase	VITEC
26	Information	Modell Infrastruktur	Kommentar	CONSENS
27	Model	Modell Infrastruktur	Modell	SysML
28	Package	Modell Infrastruktur	Package	SysML
29	View	Modell Infrastruktur	Sicht	SysML
30	Viewpoint	Modell Infrastruktur	Beschreibung Sicht	SysML
31	Package	Modell Infrastruktur	Package	SysML Lite
32	DocumentFormat	Modell Infrastruktur	DocumentFormat	VITEC
33	ReportScript	Modell Infrastruktur	ReportScript	VITEC
34	Section	Modell Infrastruktur	Berichtswesen Sektion	VITEC
35	Text	Modell Infrastruktur	Kommentar	VITEC
36	Verknüpfen	Parametr. Mod.	Parameterics Element	CONSENS
37	Constraint Block	Parametr. Mod.	Parameterics Element	Harmony SE
38	Constraint Parameters	Parametr. Mod.	Berechnungs Parameter	Harmony SE
39	Constraint Block	Parametr. Mod.	Parameterics Element	SysML
40	Constraint Parameters	Parametr. Mod.	Berechnungs Parameter	SysML
41	Constraint Property	Parametr. Mod.	Parameterics Element Instanz	SysML
42	Constraint Parameters	Parametr. Mod.	Berechnungs Parameter	SysML Lite
43	Constraint Property	Parametr. Mod.	Parameterics Element Instanz	SysML Lite
44	Einfluss	Systemstruktur	Einfluss auf das System	CONSENS
45	Erfassung einer Größe	Systemstruktur	Required&Provided Features	CONSENS
46	Parameter	Systemstruktur	Attribut	CONSENS
47	Partialmod. Gestalt	Systemstruktur	Partialmodell Gestalt	CONSENS
48	Partialmod. Umfeld	Systemstruktur	Partialmodell Umfeld	CONSENS
49	Partialmod. Wirkstruktur	Systemstruktur	Partialmodell Wirkstruktur	CONSENS
50	Systemelement	Systemstruktur	System Element	CONSENS
51	Block	Systemstruktur	System Element	Harmony SE
52	FlowProperty	Systemstruktur	Fluss über Port	Harmony SE
53	FullPort	Systemstruktur	erweiterte Schnittstelle	Harmony SE
54	Port	Systemstruktur	basic Schnittstelle	Harmony SE
55	Required&Provided Interfaces	Systemstruktur	Required&Provided Features	Harmony SE
56	Value Property	Systemstruktur	Attribut	Harmony SE
57	Attribute	Systemstruktur	Attribut	OPM
58	Object	Systemstruktur	System Element	OPM
59	Logical Port	Systemstruktur	erweiterte Schnittstelle	RFLP

108 Anhang: Analyse der Sprachräume

Nr.	Entität	Gruppe	Bedeutung	Sprache
60	Logical Reference	Systemstruktur	System Element	RFLP
61	Abstract	Systemstruktur	abstrakte Klasse, die nicht instanziiert werden kann	SysML
62	Block	Systemstruktur	System Element	SysML
63	connector property	Systemstruktur	connector "part" (Enden)	SysML
64	Enumeration	Systemstruktur	Wertetyp mit einer Auswahlliste	SysML
65	FlowProperty	Systemstruktur	Fluss über Port	SysML
66	FullPort	Systemstruktur	erweiterte Schnittstelle	SysML
67	InstanceSpecification	Systemstruktur	Instanziierung	SysML
68	Interface	Systemstruktur	Software Interface	SysML
69	Interface Block	Systemstruktur	Software Schnittstelle	SysML
70	participant property	Systemstruktur	association "part" (Enden)	SysML
71	Port	Systemstruktur	basic Schnittstelle	SysML
72	Port (nested)	Systemstruktur	Port Nested	SysML
73	Property	Systemstruktur	Part Property	SysML
74	ProxyPort	Systemstruktur	Proxy Port	SysML
75	Required&Provided Features	Systemstruktur	Required&Provided Features	SysML
76	Required&Provided Interfaces	Systemstruktur	Required&Provided Interfaces	SysML
77	StereotypeProperty	Systemstruktur	Stereotyp	SysML
78	Value Property	Systemstruktur	Attribut	SysML
79	Value Type	Systemstruktur	Parameter Wert	SysML
80	Block	Systemstruktur	System Element	SysML Lite
81	Port	Systemstruktur	basic Schnittstelle	SysML Lite
82	Value Property	Systemstruktur	Attribut	SysML Lite
83	Component	Systemstruktur	System Element	VITEC
84	Interface	Systemstruktur	basic Schnittstelle	VITEC
85	Item	Systemstruktur	erweiterte Schnittstelle	VITEC
86	Link	Systemstruktur	Fluss über Port	VITEC
87	TestCase	Validierung	Testfall	Harmony SE
88	TestCase	Validierung	Testfall	SysML
89	TestActivity	Validierung	Testfunktion	VITEC
90	TestConfiguration	Validierung	Validierungssystem	VITEC
91	TestItem	Validierung	Einzelkomponente Test	VITEC
92	TestProcedure	Validierung	Einzelelemente eines VerificationEvents	VITEC
93	VerificationEvent	Validierung	Testfall	VITEC
94	VerificationRequirement	Validierung	TestRequirement	VITEC
95	Aktivität	Verhalten	Instanz einer Funktion	CONSENS
96	Alternative	Verhalten	Aternative Beziehungen	CONSENS
97	Ende	Verhalten	Final Node	CONSENS
98	Ereignis	Verhalten	Trigger	CONSENS
99	Funktion	Verhalten	Funktion	CONSENS
100	Partialmod. Aktivitäten	Verhalten	Partialmodell Aktivitäten	CONSENS
101	Partialmod. Funktionen	Verhalten	Partialmodell Funktionen	CONSENS
102	Partialmod. Zustände	Verhalten	Partialmodell Zustände	CONSENS
103	Start	Verhalten	Initial Node	CONSENS
104	Teilen	Verhalten	Fork Node	CONSENS
105	Verbinden	Verhalten	Verbinden	CONSENS
106	Zusammenführen	Verhalten	Join Node	CONSENS
107	Zustand	Verhalten	Zustand	CONSENS
108	Action	Verhalten	Funktion	Harmony SE
109	Activity Final	Verhalten	Final Node	Harmony SE
110	Actor Pin	Verhalten	basic Schnittstelle	Harmony SE
111	And Line	Verhalten	And Line	Harmony SE
112	Condition Mark	Verhalten	Condition Mark	Harmony SE
113	Decision Node	Verhalten	Decision Node	Harmony SE
114	Fork Node	Verhalten	Fork Node	Harmony SE
115	History Connector	Verhalten	History Connector	Harmony SE
116	Initial Node	Verhalten	Initial Node	Harmony SE
117	Instance Line	Verhalten	Lifeline	Harmony SE
118	Interaction Operator	Verhalten	Interaction Operator	Harmony SE

Nr.	Entität	Gruppe	Bedeutung	Sprache
119	Join Node	Verhalten	Join Node	Harmony SE
120	MergeNode	Verhalten	Join Node	Harmony SE
121	Send Action State	Verhalten	Send Action State	Harmony SE
122	State	Verhalten	Zustand	Harmony SE
123	Time Interval	Verhalten	Time Constraint	Harmony SE
124	Command	Verhalten	Command	JPL SA
125	Goal	Verhalten	Goal	JPL SA
126	Measurement	Verhalten	Measurement	JPL SA
127	State Variable	Verhalten	Zustand	JPL SA
128	time point	Verhalten	time point	JPL SA
129	Operation	Verhalten	Operation/Methode	OPM
130	Process	Verhalten	Funktion	OPM
131	State	Verhalten	Zustand	OPM
132	Flow Association Control	Verhalten	Flow Association Control	RFLP
133	Flow Association Data	Verhalten	Flow Association Data	RFLP
134	Functional Reference	Verhalten	Funktion	RFLP
135	AcceptEventAction	Verhalten	Event und TimeEvent	SysML
136	Action	Verhalten	Instanz einer Funktion	SysML
137	Action (SM)	Verhalten	Transistion Action	SysML
138	Activity	Verhalten	Funktion	SysML
139	Activity Final	Verhalten	Final Node	SysML
140	Activity Parameter Node	Verhalten	basic Schnittstelle	SysML
141	ActivityPartition	Verhalten	ActivityPartition	SysML
142	CallBehavior Action	Verhalten	Instanz einer Funktion	SysML
143	CombinedFragment	Verhalten	CombinedFragment	SysML
144	Composite state	Verhalten	Zustand	SysML
145	Constraint Note	Verhalten	Time Constraint	SysML
146	Control Operator (Stereotyp Behaviors)	Verhalten	Control	SysML
147	Coregion	Verhalten	Coregion	SysML
148	CreationEvent	Verhalten	CreationEvent	SysML
149	Decision Node	Verhalten	Decision Node	SysML
150	Destruction Event	Verhalten	Destruction Event	SysML
151	Duration Constraint	Verhalten	Duration Constraint	SysML
152	Entry point	Verhalten	Entry point	SysML
153	ExecutionSpecification	Verhalten	ExecutionSpecification	SysML
154	Exit point	Verhalten	Exit point	SysML
155	Final Node	Verhalten	Final Node	SysML
156	Final State	Verhalten	Final Node	SysML
157	Flow Final	Verhalten	Final Node	SysML
158	Fork Node	Verhalten	Fork Node	SysML
159	History Deep Pseudo state	Verhalten	History Deep Pseudo state	SysML
160	History shallow Pseudo state	Verhalten	History shallow Pseudo state	SysML
161	Initial Node	Verhalten	Initial Node	SysML
162	Initial pseudo state	Verhalten	Initial pseudo state	SysML
163	InteractionUse	Verhalten	InteractionUse	SysML
164	InterruptibleActivityRegion	Verhalten	InterruptibleActivityRegion	SysML
165	isControl	Verhalten	isControl	SysML
166	isStream	Verhalten	isStream	SysML
167	Join Node	Verhalten	Join Node	SysML
168	Junction pseudo state	Verhalten	Junction pseudo state	SysML
169	Lifeline	Verhalten	Lifeline	SysML
170	Local pre- and postconditions	Verhalten	Local pre- and postconditions	SysML
171	MergeNode	Verhalten	Join Node	SysML
172	NoBuffer (Stereotyp ObjectNode)	Verhalten	NoBuffer (Stereotyp ObjectNode)	SysML
173	ObjectNode	Verhalten	ObjectNode	SysML
174	Optional (Stereotyp "Parameters")	Verhalten	optional: Muss nicht mit einem Wert belegt sein	SysML
175	OverWrite (Stereotyp ObjectNode)	Verhalten	OverWrite (Stereotyp ObjectNode)	SysML
176	ParameterSet	Verhalten	ParameterSet	SysML

110 Anhang: Analyse der Sprachräume

Nr.	Entität	Gruppe	Bedeutung	Sprache
177	Probability (Stereotype ControlFlow and ObjectFlow)	Verhalten	Probability (Stereotype ControlFlow and ObjectFlow)	SysML
178	Rate (Stereotype ControlFlow and ObjectFlow)	Verhalten	Rate (Stereotype ControlFlow and ObjectFlow)	SysML
179	Recieve signal action	Verhalten	Recieve signal action	SysML
180	Region	Verhalten	Region	SysML
181	Send signal action	Verhalten	Send signal action	SysML
182	SendSignalAction	Verhalten	SendSignalAction	SysML
183	Simple State	Verhalten	Simple State	SysML
184	State List	Verhalten	State List	SysML
185	State Machine	Verhalten	State Machine	SysML
186	StateInvariant / Continuations	Verhalten	StateInvariant / Continuations	SysML
187	StructuredActivityNode	Verhalten	StructuredActivityNode	SysML
188	Submachine State	Verhalten	Submachine State	SysML
189	Terminate node	Verhalten	Terminate node	SysML
190	Time Constraint	Verhalten	Time Constraint	SysML
191	Action	Verhalten	Instanz einer Funktion	SysML Lite
192	Activity Parameter Node	Verhalten	basic Schnittstelle	SysML Lite
193	Final Node	Verhalten	Final Node	SysML Lite
194	Initial Node	Verhalten	Initial Node	SysML Lite
195	Event	Verhalten	Trigger	VITEC
196	Function	Verhalten	Funktion	VITEC
197	Mode	Verhalten	Ansammlung an Zuständen	VITEC
198	State	Verhalten	Zustand	VITEC
199	Transition	Verhalten	Zustandsübergang	VITEC
200	Anmerkungen/offene Punkte	Zusatzinformationen	Kommentar	CONSENS
201	Requirement (Type: Constraint)	Zusatzinformationen	Requirement (Type: Constraint)	OPM
202	Comment	Zusatzinformationen	Kommentar	SysML
203	Problem	Zusatzinformationen	Problem generell	SysML
204	Rationale	Zusatzinformationen	Design-Entscheidung	SysML
205	ChangeRequestPackage	Zusatzinformationen	Change Management	VITEC
206	Concern	Zusatzinformationen	Problem generell	VITEC
207	Product	Zusatzinformationen	Produkt	VITEC
208	Resource	Zusatzinformationen	Sektion	VITEC

Tabelle 18: Sprachenanalyse, Relationen mit abstrahierter Bedeutung, verkürzte Darstellung durch Verzicht der Aufschlüsselung nach Gruppierung

Nr.	Relation	Bedeutung	Sprache
1	Implement Relation	Abbildung Funktion auf Gestalt	RFLP
2	SwimLane (SysML Spez schwammig)	Abbildung Funktion auf Gestalt	SysML
3	Swimlane	Abbildung Funktion auf Gestalt	SysML Lite
4	performs	Abbildung Funktion auf Gestalt	VITEC
5	«derive»	Ableitung	SysML
6	«derive»	Ableitung	Harmony SE
7	accomplishes	accomplishes	VITEC
8	«allocate»	allg Zusammenhänge	SysML
9	dependency	allg Zusammenhänge	SysML
10	dependency	allg Zusammenhänge	Harmony SE
11	assigned to	assigned to	VITEC
12	augments	augments	VITEC
13	agent link	Bedingung	OPM
14	instrument link	Bedingung	OPM
15	bidirectional connector	bidirectional connector	SysML
16	categorizes	categorizes	VITEC
17	causes	causes	VITEC
18	contains	contains	VITEC
19	describes	describes	VITEC
20	documents	documents	VITEC
21	Einfluss auf	Einfluss auf	CONSENS
22	elaborates	elaborates	VITEC
23	elicits	elicits	VITEC
24	employs	employs	VITEC
25	agent link	Enabling	OPM
26	instrument link	Enabling	OPM
27	encompasses	encompasses	VITEC
28	Energiefluss	Energiefluss	CONSENS
29	enters	enters	VITEC
30	"unfolds"	Entfalten	OPM
31	«satisfy»	Erfüllung	SysML
32	specifies	Erfüllung	VITEC
33	«satisfy»	Erfüllung	Harmony SE
34	«satisfy»	Erfüllung	SysML Lite
35	Implement Relation	Erfüllung	RFLP
36	wird realisiert durch	Erfüllungsbeziehung	CONSENS
37	"expressing"/"surpressing"	Erscheinen	OPM
38	establishes	establishes	VITEC
39	executes	executes	VITEC
40	exits	exits	VITEC
41	formats	formats	VITEC
42	forms	forms	VITEC
43	fulfills	fulfills	VITEC
44	specifies	Funktion Gestalt Zuordnung	VITEC
45	GeneralOrdering	GeneralOrdering	SysML
46	generates	generates	VITEC
47	"zooms into"	Hineinzoomen	OPM
48	impacts	impacts	VITEC
49	incorporates	incorporates	VITEC
50	Informationsfluss	Informationsfluss	CONSENS
51	inputs	inputs	VITEC
52	Instantiation	Instanziierung	OPM
53	ItemFlow	Item Flow	SysML
54	Item Flow	Iten Flow	SysML
55	läuft auf	läuft auf	CONSENS
56	logische Beziehung	logische Beziehung	CONSENS
57	Lost_Found Message	Lost Found Message	SysML
58	Message	Message	SysML
59	Message	Message	Harmony SE

Nr.	Relation	Bedeutung	Sprache
60	contains	Mode Verknüpfung	VITEC
61	object node	object node	SysML
62	originates	originates	VITEC
63	outputs	outputs	VITEC
64	«access»	Package Access	SysML
65	«import»	Package Import	SysML
66	connects to	Port Zuweisung	VITEC
67	processes	processes	VITEC
68	Realization	Realization	SysML
69	references	references	VITEC
70	refines	refines	VITEC
71	reports on	reports on	VITEC
72	resolves	resolves	VITEC
73	results in	results in	VITEC
74	joins	Schnittstelle Zuweisung	VITEC
75	services	services	VITEC
76	signs	signs	VITEC
77	specifies	specifies	VITEC
78	Stofffluss	Stofffluss	CONSENS
79	Störbeziehung	Störbeziehung	CONSENS
80	Störfluss	Störfluss	CONSENS
81	reference association	Strukturbeziehung	SysML
82	structural link	Strukturbeziehung	OPM
83	reference association	Strukturbeziehung	Harmony SE
84	includes	Subsumption	VITEC
85	composition	Subsumption	SysML
86	Aggregation	Subsumption	OPM
87	Package Containment	Subsumption	SysML
88	ist Teil von	Subsumption	CONSENS
89	logische Gruppe	Subsumption	CONSENS
90	decomposes	Subsumption	VITEC
91	containment	Subsumption	SysML
92	«extend»	Subsumption	SysML
93	«extend» with condition	Subsumption	SysML
94	aggregation	Subsumption	SysML
95	(namespace) containment	Subsumption	SysML
96	containment	Subsumption	SysML Lite
97	Dekomposition (nicht näher spez)	Subsumption	SysML Lite
98	composition	Subsumption	SysML Lite
99	aggregation	Subsumption	SysML Lite
100	extends	Subsumption	VITEC
101	built in	Subsumption	VITEC
102	containment	Subsumption	Harmony SE
103	«extend»	Subsumption	Harmony SE
104	composition	Subsumption	Harmony SE
105	supplies	supplies	VITEC
106	time constraint	time constraint	JPL SA
107	Rate	time interval	SysML
108	«trace»	Trace	SysML
109	«trace»	Trace	Harmony SE
110	traces to	Trace	VITEC
111	transfers	transfers	VITEC
112	result link	Transformation	OPM
113	consumption link	Transformation	OPM
114	invocation link	Transformation	OPM
115	Transition	Transition	SysML
116	Transition	Transition	Harmony SE
117	affects	Transition	JPL SA
118	triggers	triggers	VITEC

Nr.	Relation	Bedeutung	Sprache
119	unidirectional connector	unidirectional connector	SysML
120	involves	Verbindung Aktor zu Use-Case	VITEC
121	generalization	Vererbung	SysML
122	Generalization	Vererbung	OPM
123	generalization	Vererbung	Harmony SE
124	ist Ausprägung von	Vererbung	CONSENS
125	qualification link	Vererbung	OPM
126	«refine»	Verfeinerung	SysML
127	ist Untermenge von	Verfeinerung	CONSENS
128	refines	Verfeinerung	VITEC
129	Functional Connection	Verhaltensverknüpfung	RFLP
130	ControlFlow	Verhaltensverknüpfung	SysML
131	ObjectFlow	Verhaltensverknüpfung	SysML
132	ControlFlow	Verhaltensverknüpfung	SysML Lite
133	ControlFlow	Verhaltensverknüpfung	Harmony SE
134	verifies	Verifizierung	VITEC
135	«verify»	Verifizierung	SysML
136	«verify»	Verifizierung	Harmony SE
137	binding connector	Verknüpfung Berechnung	Harmony SE
138	binding connector	Verknüpfung Berechnung	SysML Lite
139	based on	Verknüpfung Funktion Anforderung	VITEC
140	attach to	Verknüpfung mit tiefergehender Beschreibung	SysML
141	comprises	Verknüpfung mit tiefergehender Beschreibung	VITEC
142	binding connector	Verknüpfung Ports	Harmony SE
143	binding connector	Verknüpfung Ports	SysML
144	binding connector	Verknüpfung Ports	SysML Lite
145	Logical Connection	Verknüpfung Ports	RFLP
146	Verweise	Verweise	CONSENS
147	«import»	View Import	SysML
148	Probability	Wahrscheinlichkeit	SysML
149	Wechselwirkung	Wechselwirkung	CONSENS
150	«copy»	Wiederverwendung	SysML
151	«conform»	Zuordnung Sicht - Beschreibung	SysML
152	includes	Zusammenführung	VITEC
153	«include»	Zusammenführung	SysML
154	«include»	Zusammenführung	Harmony SE
155	effect link	Zustand Transoformation	OPM
156	imput link	Zustand Transoformation	OPM
157	output link	Zustand Transoformation	OPM
158	implicit status representation	Zustand Verknüpfung	OPM
159	"hosts"	Zustand Verknüpfung Expression	OPM
160	exhibits	Zustand Verknüpfung Expression	VITEC
161	used in	Zuweisung Begriffsdefinition	VITEC
162	Exhibition	Zuweisung Operation	OPM
163	Exhibition	Zuweisung Parameter	OPM

Tabelle 19: Sprachenanalyse, Entitäten mit Vorkommenshäufigkeit von ≥ 2 , Aufschlüsselung nach Ansätzen

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Teilsystem / Systemelement	X	X	X	X		X	X	X
	Block Subject	Block	Block	Component		Object	Logical Reference	Systemelement
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Funktion	X	X	X	X		X	X	X
	Activity Action*	Action*	Action*	Function		Process	Functional Reference	Funktion Aktivität*
* bezeichnet die Instanz einer Funktion	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Zustand	X		X	X	X	X		X
	Composite state		State	State	State Variable	State		Zustand

		<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Anforderung		X	X	X	X			X	
	Requirement							Requirement	
		<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Attribut (Parameter)		X	X	X			X		X
	Value Property						Attribute		Parameter
		<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Schnittstelle		X	X	X	X			X	
	Activity Parameter Node Port / Full Port								
	Activity Parameter Node Port								
	Actor Pin Port / Full Port								
	Interface Item								
	Logical Port								

Akteur	X		X	X					
	Actor		Actor	Actor					
Anwendungsfall	X		X	X					
	UseCase		UseCase	UseCase					
Startpunkt	X	X	X						X
	Initial Node	Initial Node	Initial Node						Start

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Fluss über Schnittstelle	X		X	X				
	Flow Property		Flow Property	Link				
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Trigger Ereignis				X				X
				Event				Ereignis
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Kommentar / Text	X			X				X
	Comment			Text				Anmerkungen Information

Tabelle 20: Sprachenanalyse, gruppierungsübergreifende Relationen mit Vorkommenshäufigkeit von ≥ 2 , Aufschlüsselung nach Ansätzen

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Erfüllung	X	X	X	X			X	
	«satisfy»	«satisfy»	«satisfy»	specifies			Implement Relation	
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Relation zw. Funktion und Systemarchitektur	X	X		X			X	
	SwimLane	SwimLane		performs			Implement Relation	
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Verifizierung	X		X	X				
	«verify»		«verify»	verifies				

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Relation zu Zuständen				X		X		
				exhibits		"hosts"		
Relation zu parametrischen Elementen	X	X	X					
	binding connector	binding connector	binding connector					
allg Zusammenhänge / Relation	X		X					
	«allocate» dependency		dependency					

Tabelle 21: Sprachenanalyse, Relationen innerhalb von Gruppierungen mit Vorkommenshäufigkeit von ≥ 2 , Aufschlüsselung nach Ansätzen

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Subsumtion	X	X	X	X		X		X
	containment «extend» composition aggregation	containment composition aggregation	containment «extend» composition	extends decomposes includes built in		Aggregation		ist Teil von logische Gruppe
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Vererbung	X		X			X		X
	generalization		generalization			Generalization		ist Ausprägung von

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Ableitung	X		X					
	«derive»		«derive»					
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Trace	X		X	X				
	«trace»		«trace»	traces to				
	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Verknüpfung mit weiterführender Beschreibung	X			X				
	attach to			comprises				

	<i>SysML</i>	<i>SysML Lite</i>	<i>IBM Telelogic Harmony SE</i>	<i>Vitech SDL</i>	<i>JPL State Analysis</i>	<i>Dori OPM</i>	<i>RFLP in Catia V6</i>	<i>CONSENS</i>
Message	X		X					
Message			Message					

13 Anhang: Grafischer Plot der Sprachräume

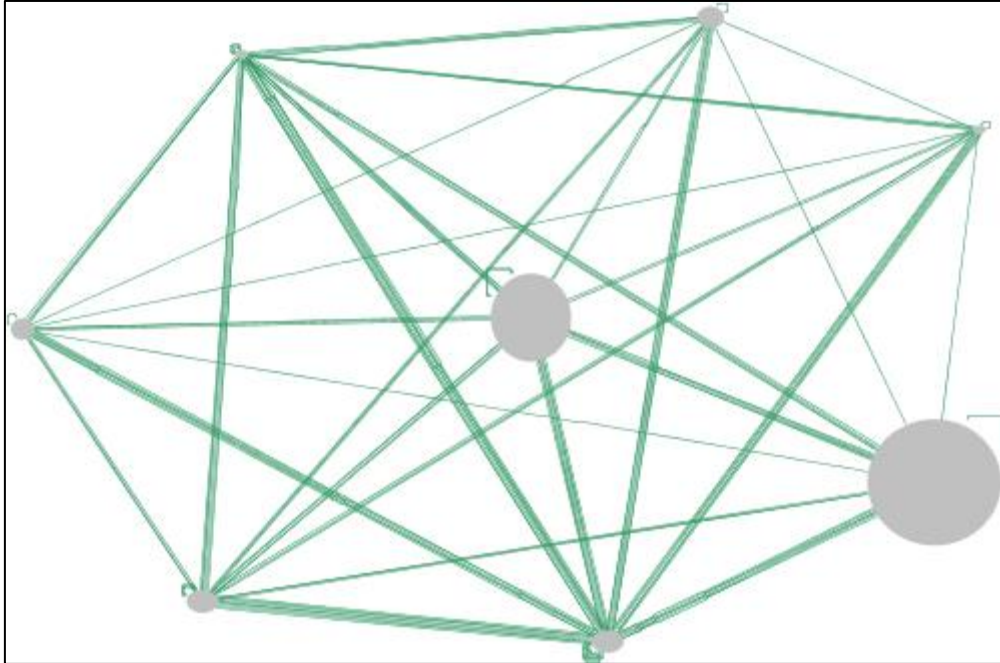


Bild 31: SysML

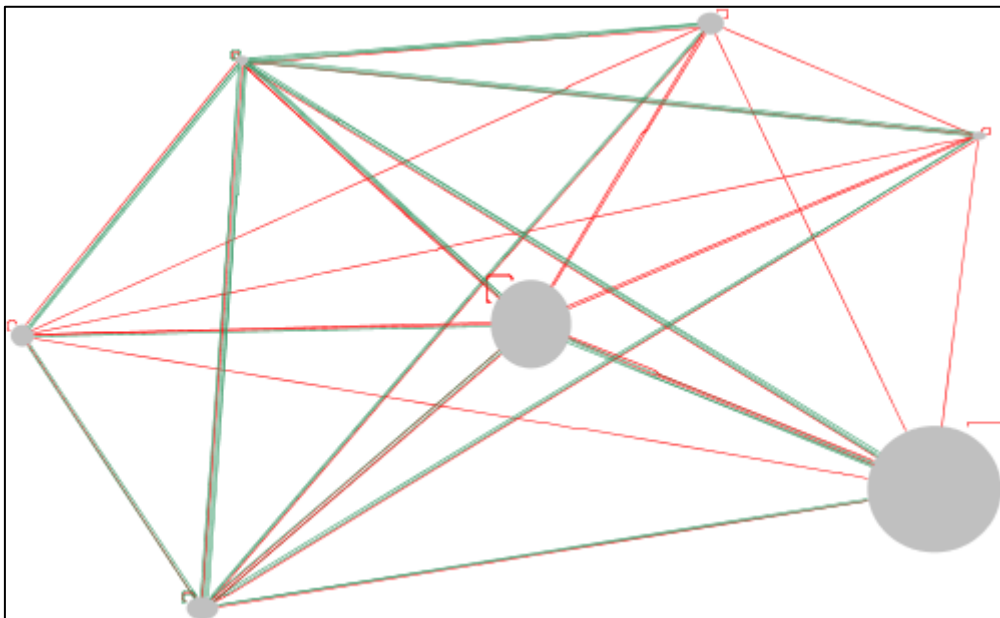


Bild 32: SysML, generische Relationen in roter Farbe markiert

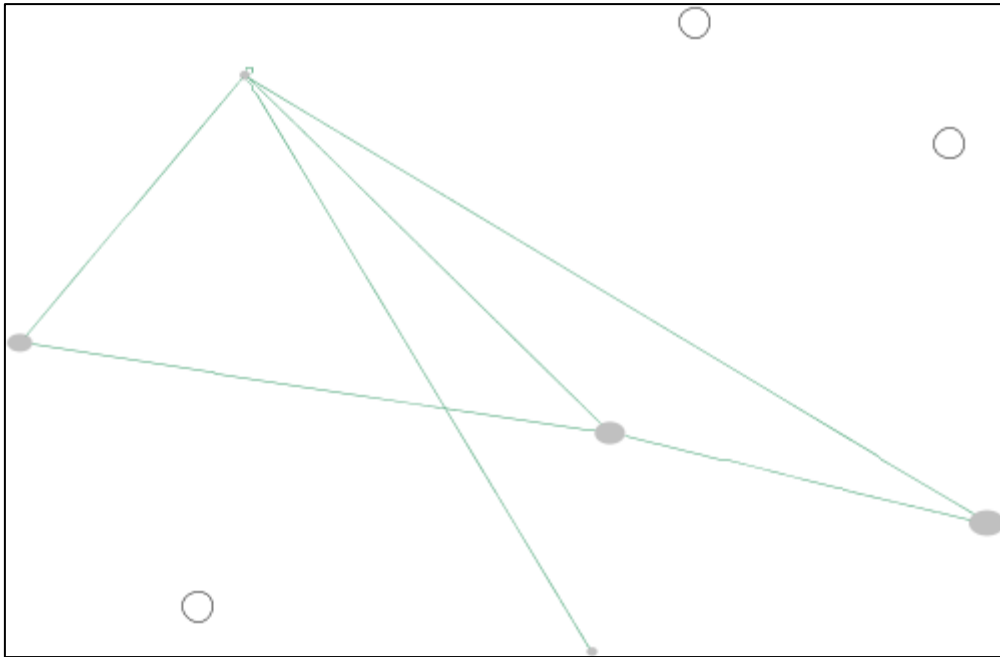


Bild 33: SysML LITE

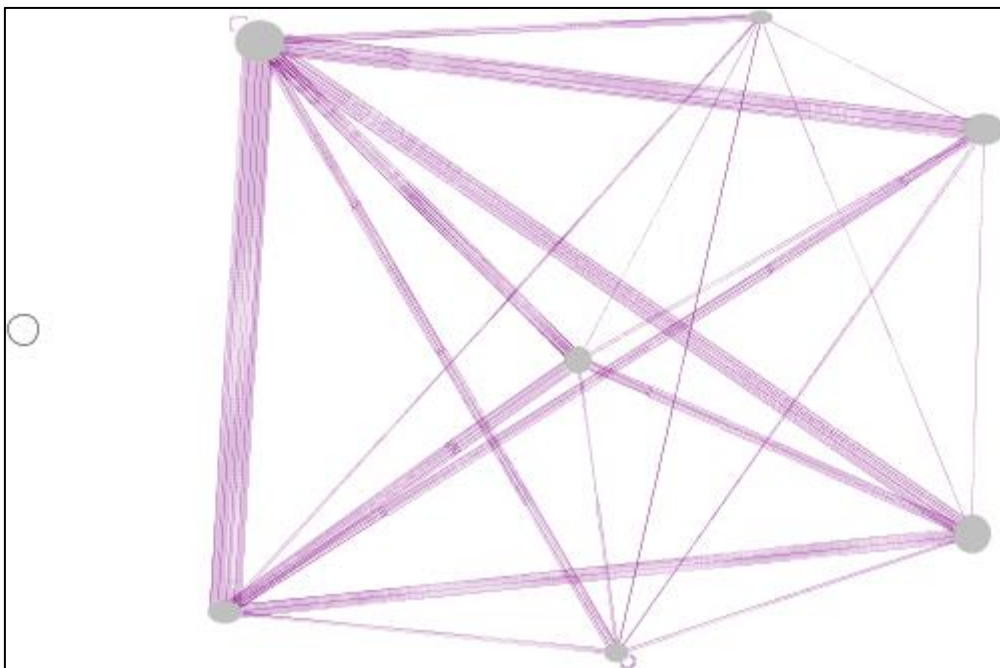


Bild 34: VITEC

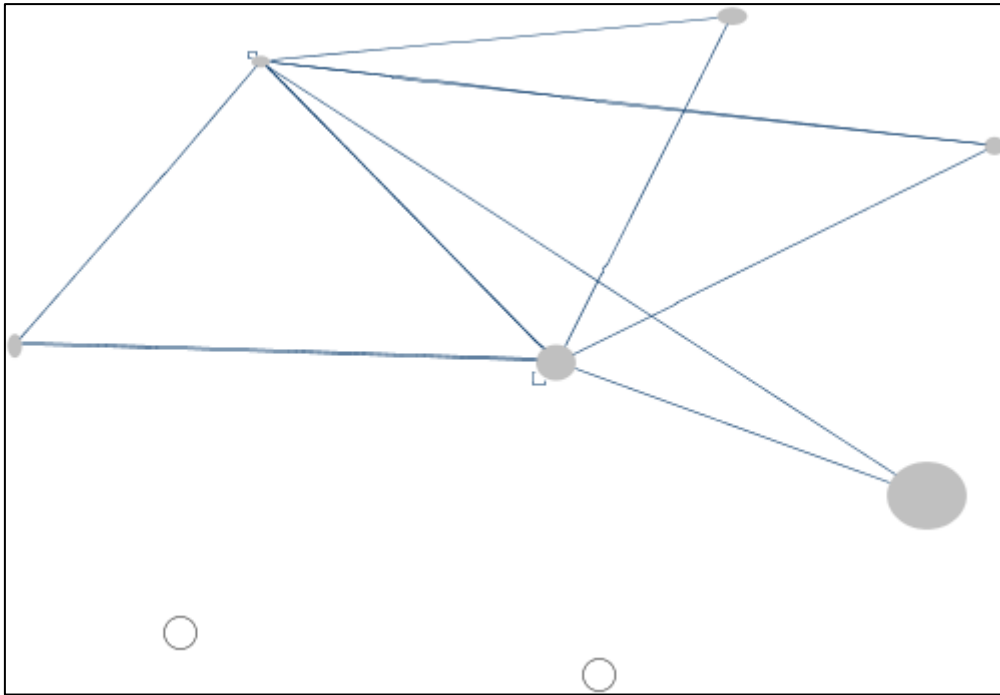


Bild 35: Harmony SE

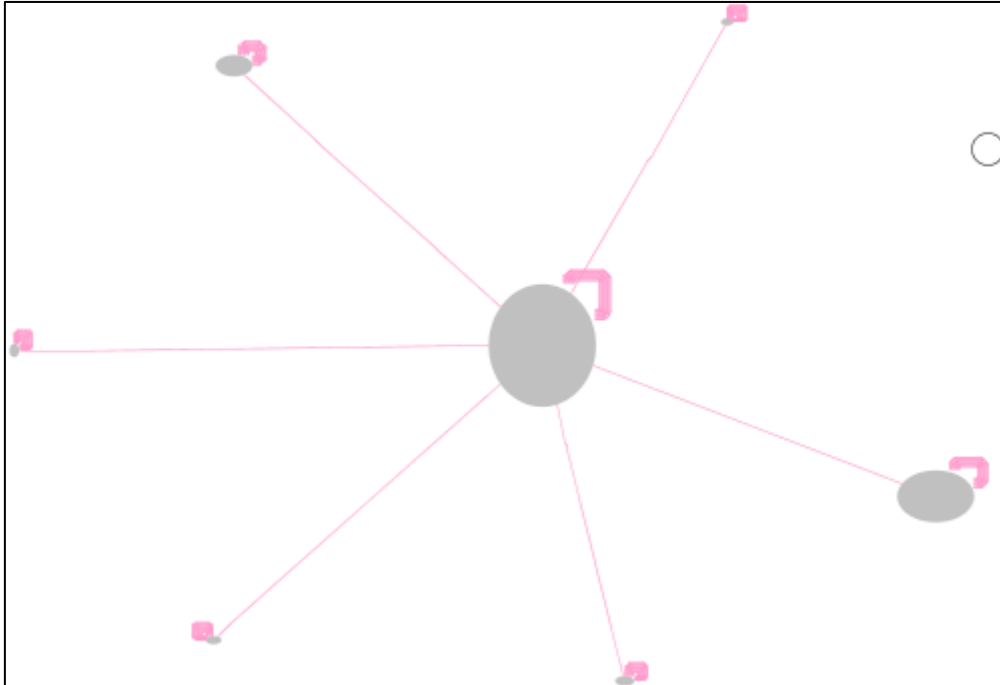


Bild 36: CONSENS

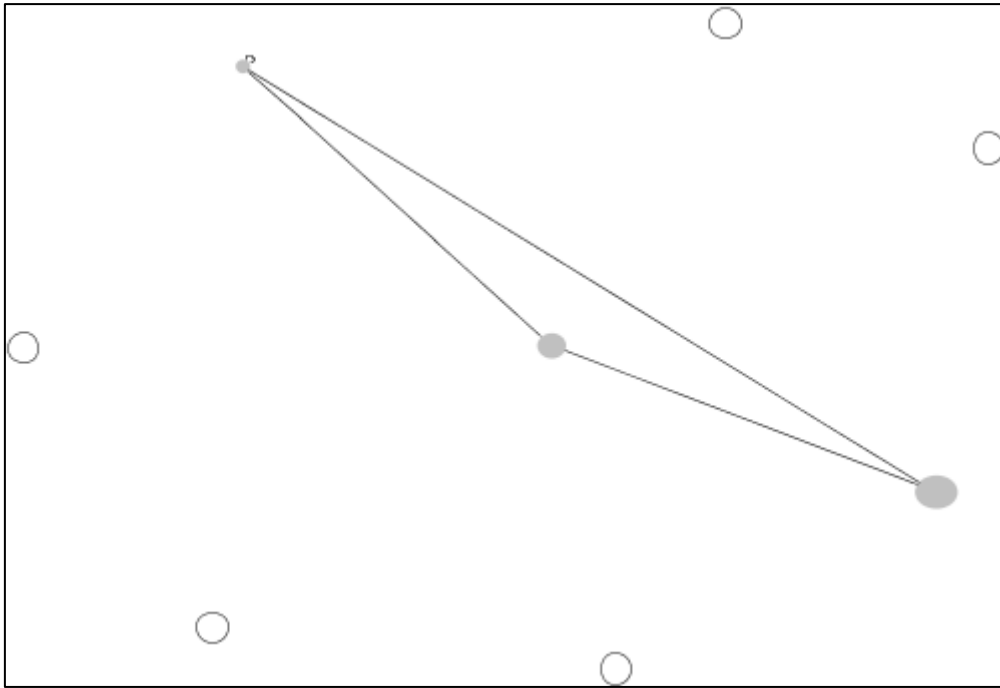


Bild 37: RFLP

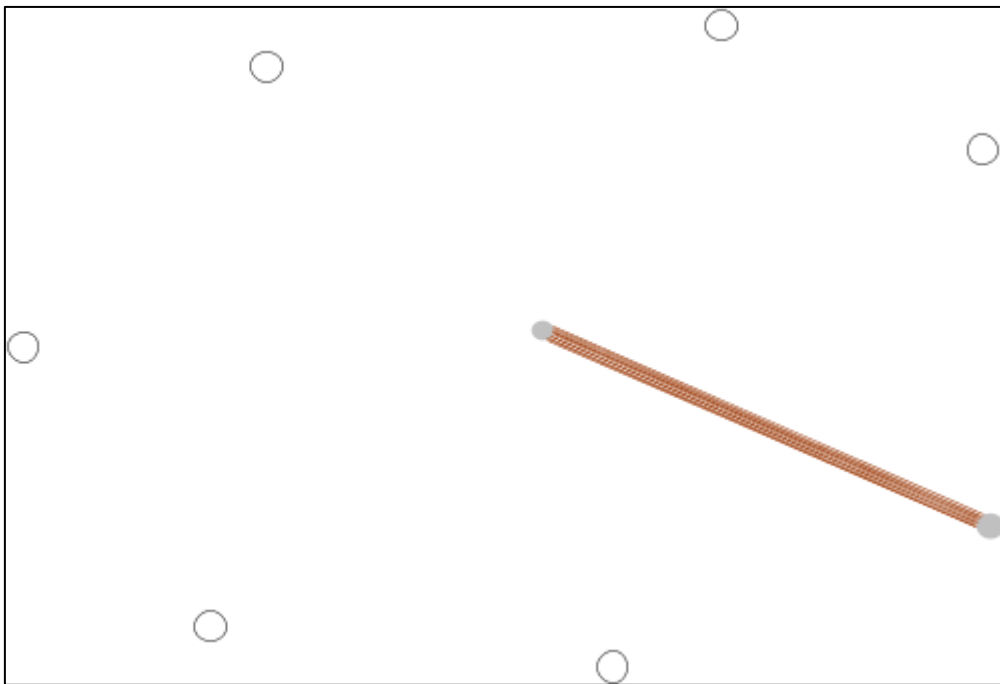


Bild 38: OPM

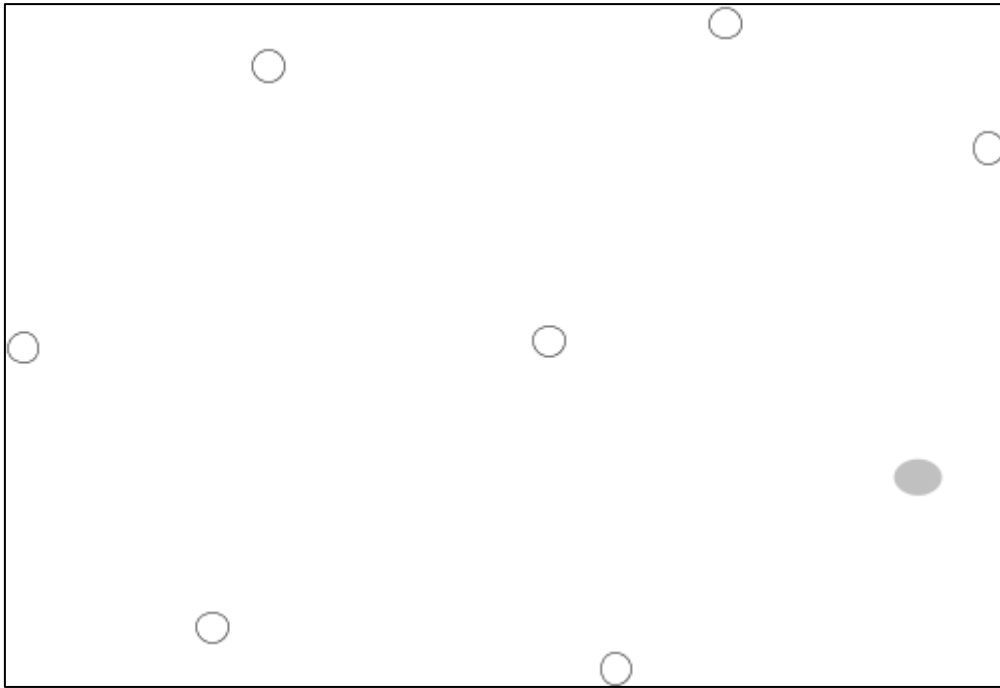

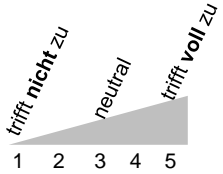


Bild 39: JPL SA

14 Anhang: Materialien zur Validierungsstudie

Fragebogen Studie





1 2 3 4 5

Ich denke, dass ich **SystemSketcher** gerne häufig benutzen würde _____

Ich fand **SystemSketcher** unnötig komplex _____

Ich fand **SystemSketcher** einfach zu benutzen _____

Ich glaube, ich würde die Hilfe einer technisch versierten Person benötigen,
um **SystemSketcher** benutzen zu können _____

Ich fand, die verschiedenen Funktionen in **SystemSketcher** waren gut integriert _____

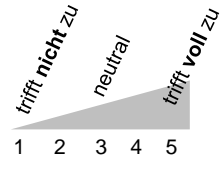
Ich denke, **SystemSketcher** enthielt zu viele Inkonsistenzen _____

Ich kann mir vorstellen, dass die meisten Menschen den Umgang
mit diesem **SystemSketcher** sehr schnell lernen. _____

Ich fand **SystemSketcher** sehr umständlich zu nutzen _____

Ich fühlte mich bei der Benutzung von **SystemSketcher** sehr sicher _____

Ich musste eine Menge lernen, bevor ich anfangen konnte
SystemSketcher zu verwenden _____



1 2 3 4 5

Ich kann mir vorstellen, dass mir **SystemSketcher** helfen könnte, ...

... zu verstehen, wie ein System beschaffen ist (Systemarchitektur) _____

... zu beschreiben, welchen Zielvorgaben ein System unterliegen soll _____

... zu verstehen, welchen Zielvorgaben ein System unterliegt _____

... zu beschreiben, wie ein System beschaffen sein soll (Systemarchitektur) _____

Ich finde eine interdisziplinäre Systemmodellierung generell sinnvoll _____

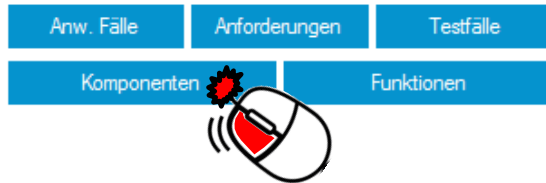
Bild 40: Fragebogen

SYSTEMELEMENTE modellieren

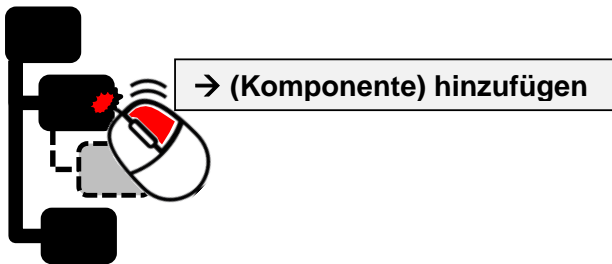


(Komponenten, Funktionen, Anforderungen, Anwendungsfälle)

1. Gewünschtes Partialmodell auswählen



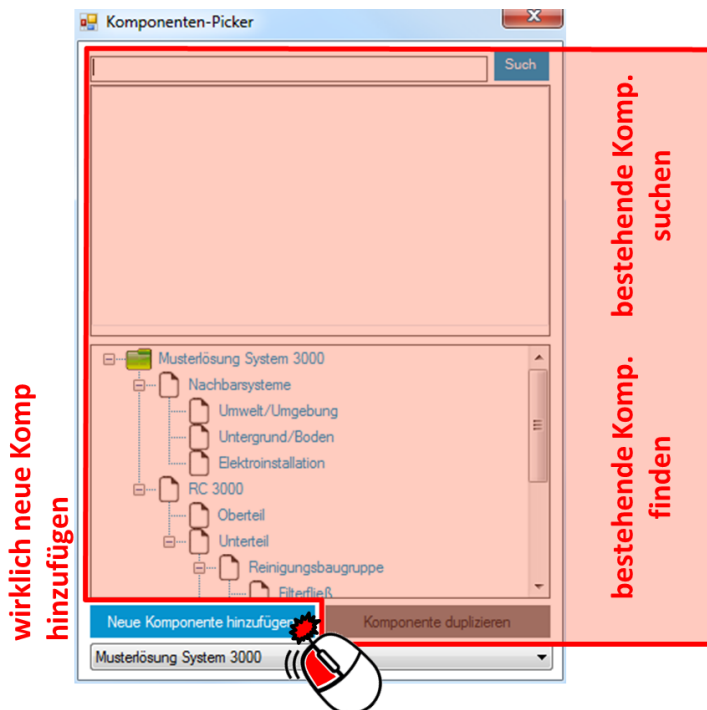
2. Neues Element in Baumstruktur anlegen



3. Komponenten Picker öffnet sich

Hinweis Prototyp: nur bei Komponenten

Gerade in großen und komplexen Systemmodellen kann es vorkommen, dass Systemkomponenten mehrfach verwendet werden. Wir haben uns daher entschieden, bei SystemSketcher einen „Komponenten Picker“ einzubauen, mit dem man bestehende Systemkomponenten auswählen und wiederverwenden kann. **In dieser Schulung werden wir jedoch vorerst einmalige, wirklich neue Komponenten verwenden.**



4. Neues Element hinzufügen

Name eingeben → OK

Bild 41: Anleitungsblatt Systemelemente

METADATEN modellieren

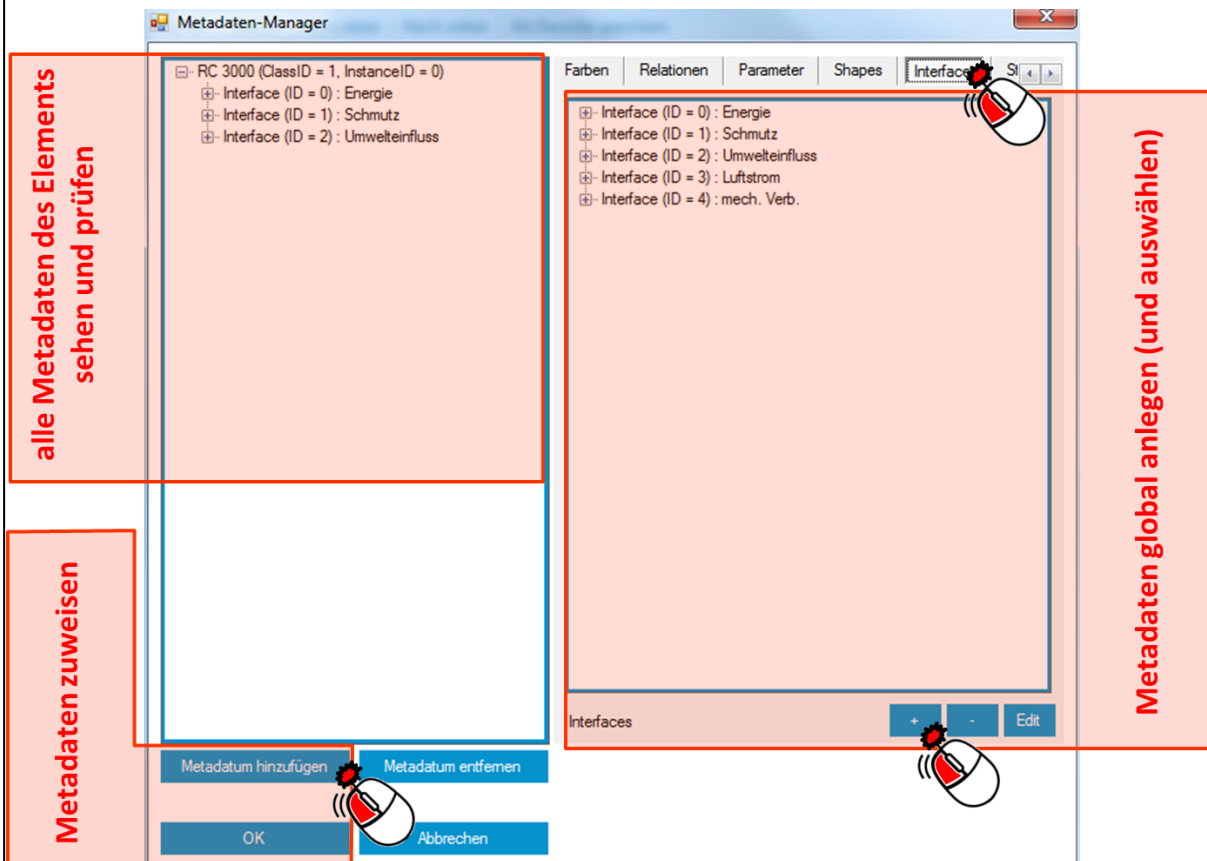
(Schnittstellen, Parameter, Form, Farben, etc.)

1. Metadaten Manager starten



2. Metadaten anlegen und zuweisen

Im Metadaten-Manager werden die Metadaten zuerst global angelegt, dann ausgewählt und den Elementen zugeordnet. Durch das globale Anlegen können die gleichen Metadaten effektiv wiederverwendet werden. Schnittstellen zählen als Eigenschaften von Komponenten in SystemSketcher auch zu den Metadaten



Hinweis Prototyp: Wir wissen, dass sich der Metadaten Manager noch relativ umständlich anwenden lässt. Leider ist es uns im Prototyp nicht gelungen, eine elegantere Lösung umzusetzen. Es ist geplant, eine Berechnungsfunktion in SystemSketcher zu programmieren, mit der ein Anwender selbst Berechnungen und Auswertungen nach den Metadaten konfigurieren kann.

Bild 42: Anleitungsblatt Metadaten

GRAFISCH modellieren

1. Elemente auf der Diagrammfläche platzieren

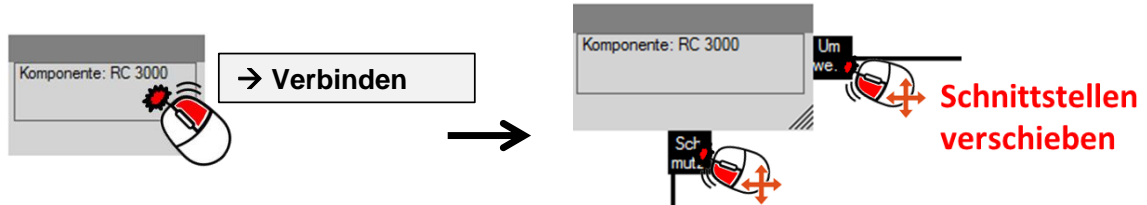


In SystemSketcher können beliebige Elemente auf jeder Diagrammfläche platziert werden.

Hinweis Prototyp: Die grafische Modellierung ist nur für Komponenten umgesetzt.

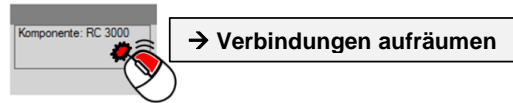
Tip: Zuerst Elemente platzieren, dann verbinden

2. Elemente über Schnittstellen verbinden



Hinweis Prototyp: Die Schnittstellen bleiben an den Kanten nicht kleben, sie können frei verschieben.

Haben Sie die Schnittstellen hoffnungslos verschoben, können Sie die initiale Position wiederherstellen:



3. Favoriten speichern

In SystemSketcher können beliebig viele Diagramme als Favoriten gespeichert werden.

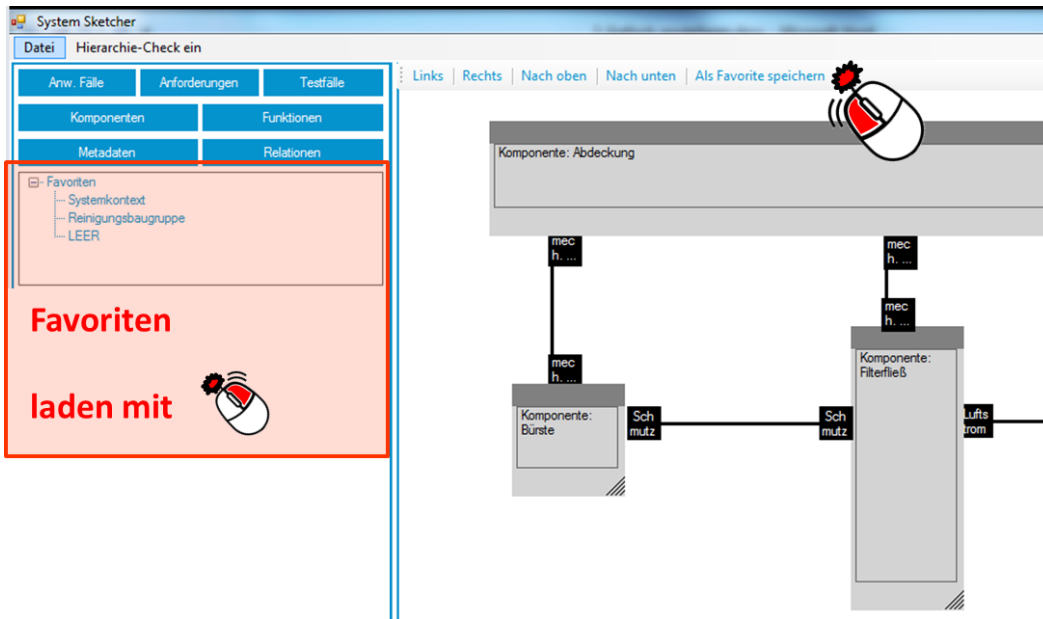


Bild 43: Anleitungsblatt grafische Modellierung

RELATIONEN modellieren

1. Relationen Manager starten

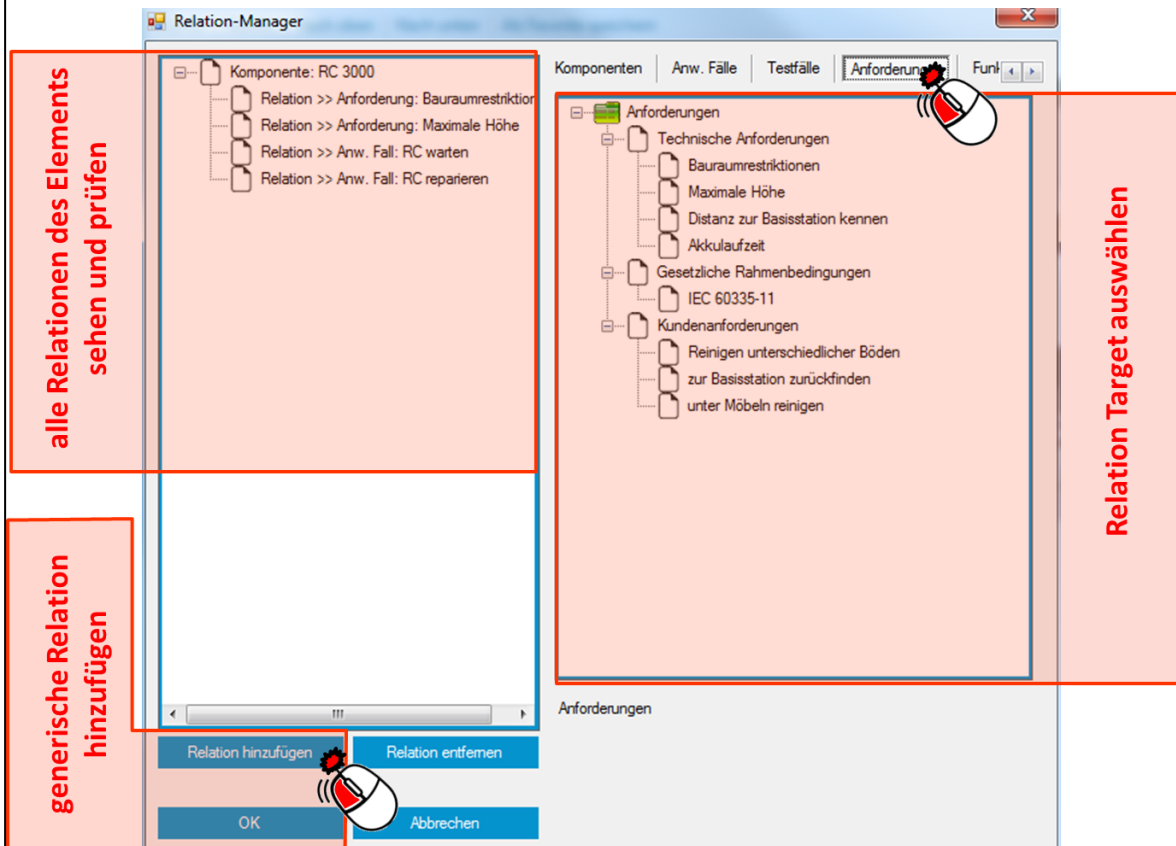


2. Relationen anlegen

In der Systemmodellierung scheint meist eine einzige Relation generischer art ausreichend zu sein. Die Bedeutung soll sich aus dem Kontext der verbundenen Elemente ergeben.

Es ist geplant, dass der Anwender zukünftig eigene Arten von Relationen anlegen kann.

Im Relationen Manager wird das „Target“-Element ausgewählt und die Relation dorthin hinzugefügt.



Die generische SystemSketcher Relation ist immer bidirektional. Sie erscheint auch beim „Target“-Element.

Hinweis Prototyp: Es ist erstrebenswert, Relationen auf der grafischen Modellierungsfläche zu verwenden. Im Prototyp kann für Relationen jedoch ausschließlich der Relationen Manager verwendet werden.

Bild 44: Anleitungsblatt Relationen

Lebenslauf

Persönliche Daten

Name: Florian Munker
Geburtsdatum: 1. Mai 1983
Geburtsort: Lauf an der Pegnitz
Staatsangehörigkeit: Deutsch

Bildungsgang

1989 – 1993 Grund- und Teilhauptschule Heuchling bei Lauf (Grundschule)
1993 – 2002 Christoph-Jacob-Treu Gymnasium Lauf (Abschluss: Allgemeine Hochschulreife)
2002 – 2008 Studium der Mechatronik an der Friedrich-Alexander-Universität Erlangen
(Abschluss: Diplom)

Berufstätigkeit

2008 – 2012 Validierungsingenieur bei Brose Fahrzeugteile GmbH & Co. KG in
Hallstadt bei Bamberg:

- Validierung mechatronischer Türsysteme
- Identifikation, Planung, Konzipierung, begleitete Durchführung und Auswertung von Validierungsaktivitäten
- Verantwortlich für die Validierungsaktivitäten zur Bewertung der technischen Eignung in den Phasen des PEP (Versuchsfreigaben)

2012 – 2016 Akademischer Mitarbeiter am IPEK – Institut für Produktentwicklung am
Karlsruher Institut für Technologie (KIT):

- Projektleitung und Sachbearbeitung von Innovationsprojekten im Bereich der Elektromobilität
- Projektkoordination im Verbund von Forschungspartnern aus Industrie und Wissenschaft
- Fachgruppenleiter der Gruppe „Interdisziplinäre Systemmodellierung“
- Teilnahme am Förderprogramm „X-Ment“ des KIT für Nachwuchskräfte