

Matthias Günther

Rahmenwerk zur musterbasierten Gestaltung von System of Systems

Framework for pattern-based design of System of Systems

Geleitwort

Advanced Systems Engineering – neue Methoden und Werkzeuge für die Wertschöpfung von Morgen – ist die verbindende Leitidee unserer Forschungsarbeiten. In der gleichnamigen Fachgruppe am Heinz-Nixdorf-Institut der Universität Paderborn sowie am Fraunhofer-Institut für Entwurfstechnik Mechatronik IEM arbeiten wir daran. Unser generelles Ziel ist die Steigerung der Innovationskraft von Industrieunternehmen. Schwerpunkte an beiden Instituten sind die Strategische Planung und das Systems Engineering.

Im Zuge der fortschreitenden Digitalisierung entwickeln sich technische Systeme zunehmend zu hochgradig vernetzten Systemverbünden, sogenannten System of Systems. Diese ermöglichen neue Funktionalitäten, innovative Geschäftsmodelle und bislang unerreichte Formen der Zusammenarbeit über Unternehmensgrenzen hinweg. Gleichzeitig führen die hohe Dynamik, die Vielzahl beteiligter Akteure und die wachsende Interdependenz technischer und organisatorischer Strukturen zu einem Umfeld steigender Unsicherheit und Komplexität.

Vor diesem Hintergrund gewinnt eine strukturierte und wissenschaftlich fundierte Gestaltung von System of Systems entscheidend an Bedeutung. Genau hier setzt die vorliegende Dissertation mit einem Rahmenwerk zur musterbasierten Gestaltung von System of Systems an. Ziel der Arbeit ist es, Unternehmen und Interessensgemeinschaften bei der Planung, Analyse und Entwicklung kooperierender Systeme systematisch zu unterstützen.

Besonders hervorzuheben ist die systematische Integration von Wissenselementen, die einen nachhaltigen Wissenstransfer zwischen Projekten und Organisationen ermöglichen. Die Evaluation in realen industriellen Anwendungskontexten unterstreicht die praktische Relevanz des Ansatzes und zeigt zugleich Perspektiven für die weitere Verankerung in unternehmerischen Prozessen auf.

Mit dieser Arbeit wird ein bedeutender Beitrag zur methodischen Fundierung des Advanced Systems Engineering geleistet. Die Dissertation verbindet wissenschaftliche Stringenz mit praktischer Anwendbarkeit und schafft eine tragfähige Grundlage für die Gestaltung komplexer, evolvierender Systemverbünde in Forschung und Industrie. Ich bin überzeugt, dass diese Dissertation breite Anerkennung in Wissenschaft und Praxis finden wird und wichtige Impulse für die Weiterentwicklung des Engineerings vernetzter technischer Systeme setzt.

Rahmenwerk zur musterbasierten Gestaltung von System of Systems

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
der Fakultät Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

genehmigte
DISSERTATION

von
M. Sc. Matthias Günther
aus Bielefeld

Tag des Kolloquiums: 10. Februar 2026
Referent: Prof. Dr.-Ing. Roman Dumitrescu
Korreferent: Prof. Dr.-Ing. Dr. h. c. Albert Albers

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Entwurfstechnik Mechatronik IEM im Forschungsbereich Produktentstehung. Sie ist das Ergebnis meiner wissenschaftlichen Arbeit im Rahmen von unterschiedlichen Forschungs- und Industrieprojekten.

Mein herzlicher Dank gilt Herrn Professor Dr.-Ing. Roman Dumitrescu für die Möglichkeit der Promotion. Vielen Dank für die herausragenden Voraussetzungen, die du mit deinen Ideen, deiner Vision und deiner Persönlichkeit am Fraunhofer IEM geschaffen hast. Du hast einen Forschungsbereich etabliert, in dem relevante Zukunftsfragen mitgestaltet werden und vielfältige Entwicklungsmöglichkeiten gegeben sind. In diesem Zuge möchte ich mich ganz besonders bei Herrn Dr. Harald Anacker bedanken. Mit deinen Impulsen hast du dafür gesorgt, dass ich sowohl fachlich als auch persönlich wachsen konnte. Du hast wesentlich zu meiner Entwicklung beigetragen und mich während meiner Zeit am Institut als Mentor begleitet – vielen Dank!

Zusätzlich möchte ich mich bei der Prüfungskommission in Person von Prof. Dr.-Ing. Dr. h. c. Albert Albers, Prof. Dr. Stefan Sauer und Prof. Dr. Eric Bodden bedanken. Insbesondere der wertvolle Austausch mit Prof. Albers und den Kollegen vom IPEK am KIT hat zusätzlichen Anteil an der fachlichen Tiefe der Dissertation und wird mir stets in Erinnerung bleiben.

Darüber hinaus gilt mein Dank meinen ehemaligen Kolleginnen und Kollegen. Ihr habt meine Zeit am Institut zu einem unvergesslichen Lebensabschnitt gemacht und dafür gesorgt, dass ich immer wieder gerne für Events oder einen Kaffee vorbeikomme. Insbesondere möchte ich dem Automotiv-Projektteam rund um Aschot und dem MBSE-Team mit Tobias, Rik und Fabian einen herzlichen Dank aussprechen – die Zusammenarbeit mit euch hat mir immer sehr viel Freude bereitet. Zusätzlich möchte ich mich bei den Projektpartnern aus dem MoSyS-Projekt und insbesondere bei Jörg, Denis, Carsten und Constantin bedanken. Vielen Dank für die wertvolle und konstruktive Zusammenarbeit.

Ganz besonders möchte ich mich bei meiner Familie und meinen Freunden bedanken. Vor allem meinen Eltern, die mir mein ganzes Leben Rückhalt geben, aber auch meinen Brüdern, auf die ich mich immer verlassen kann. Am allermeisten möchte ich mich bei dir, meine liebe Kathi, bedanken. Du hast mich ermutigt weiterzumachen und auch in schwierigen Phasen stehst du hinter mir. Mit deiner Lebensfreude und liebevollen Art habe ich dich in mein Herz geschlossen. Zusammen mit Leni zaubert ihr mir jeden Tag ein Lächeln aufs Gesicht. Danke, Danke, Danke!

Zusammenfassung

Die Vernetzung und Zusammenarbeit von Systemen gelten als wesentlicher Erfolgsfaktor heutiger und insbesondere zukünftiger Lösungen. Durch das Zusammenwirken unterschiedlicher Systeme werden Ziele erreicht, die durch einzelne Systeme nicht realisierbar sind. Die ganzheitliche Betrachtung von Systemverbünden über Unternehmensgrenzen hinweg eröffnet neue Lösungswege und erweitert Funktionalitäten. Die Herausforderung besteht darin, die komplexen technischen Möglichkeiten vernetzter Systeme unter Berücksichtigung unterschiedlicher Interessen und möglicher Zielkonflikte systematisch zu betrachten.

Ziel der vorliegenden Arbeit ist ein Rahmenwerk zur musterbasierten Gestaltung von System of Systems (SoS). Das Rahmenwerk soll Unternehmen und Interessensgemeinschaften eine bedarfsgerechte Gestaltung von Systemverbünden ermöglichen. Unternehmen werden befähigt, Entscheidungen über Unternehmenszusammenarbeiten zu treffen und unter verschiedenen Voraussetzungen die passenden Entwicklungstätigkeiten zu identifizieren und durchzuführen. Das Rahmenwerk besteht aus fünf Gestaltungskonzepten, die die unterschiedlichen Voraussetzungen und Ziele von SoS berücksichtigen und entsprechende Leitlinien zur Verfügung stellen. Des Weiteren werden Methoden und ein Vorgehensmodell für eine systematische Bearbeitung bereitgestellt. Die Gestaltung von SoS wird darüber hinaus durch die integrative Nutzung von Lösungswissen in Form von Lösungsmustern und Lösungsprinzipien unterstützt. Die Arbeit ist das Resultat der Forschungsarbeit im Projekt MoSyS und wurde mit drei Unternehmen evaluiert.

Abstract

Connectivity and collaboration of systems is considered as a success factor for current and, especially, future solutions. Through the interaction of different systems goals are achieved, that cannot be realized by the individual systems alone. A holistic view of system networks across company boundaries opens up new ways for solutions and expands functionalities. The challenge lies in systematically addressing the complex technical possibilities of interconnected systems while taking into account different interests and potential conflicts of objectives.

The goal of this work is to develop a framework for the design of system of systems (SoS) according to the specific needs of the use case. This framework enables companies and interest groups to develop and manage system networks according to their goals. Companies will be able to make decisions about collaborations and to identify and carry out suitable development activities under various conditions. The framework consists of five design concepts that take into account the different requirements and objectives of SoS. It provides guidance by providing suitable methods and a process model for systematic application. To further support the design of SoS, solution knowledge is available in the form of solution patterns and principles. This work is the result of research conducted within the MoSyS project and has been evaluated with three companies.

Vorveröffentlichungen

- [AGW+22] ANACKER, H.; GUENTHER, M.; WYRWICH, F.; DUMITRESCU, R.: Pattern based engineering of System of Systems - a systematic literature review. In: 17th Annual System of Systems Engineering Conference (SOSE). 07. - 11. Juni 2022, Rochester, NY, USA, IEEE, 2022, S. 178–183
- [MGM+22] MANDEL, C.; GUENTHER, M.; MARTIN, A.; WINDISCH, E.; BURSAC, N.; RAPP, S.; ANACKER, H.; ALBERS, A.: Towards a System of Systems Engineering Architecture Framework. In: 17th Annual System of Systems Engineering Conference (SOSE). 07. - 11. Juni 2022, Rochester, NY, USA, IEEE, 2022, S. 221–226
- [AG22] ANACKER, H.; GÜNTHER, M.: Menschenorientiertes Engineering. In: VDI (Hrsg.): wt Werkstattstechnik online. 3, 2022
- [KGA+22] KHARATYAN, A.; GÜNTHER, M.; ANACKER, H.; JAPS, S.; DUMITRESCU, R.: Security- and Safety-Driven Functional Architecture Development Exemplified by Automotive Systems Engineering. In: Procedia CIRP. 109, 2022, S. 586–591
- [SGM+23] SCHÄFER, L.; GÜNTHER, M.; MARTIN, A.; LÜPFERT, M.; MANDEL, C.; RAPP, S.; LANZA, G.; ANACKER, H.; ALBERS, A.; KÖCHLING, D.: Systematics for an Integrative Modelling of Product and Production System. In: 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering CIRP ICME '22. Italien, 118, 2023, S. 104–109
- [GA23] GÜNTHER, M.; ANACKER, H.: Komplexe Medizinsysteme entwerfen. In: medical design. 02, WEKA Fachmedien GmbH, 2023
- [GGH+23] GÜNTHER, M.; GÖLLNER, D.; HEIHOFF-SCHWEDE, J.; ANACKER, H.; DUMITRESCU, R.: Engineering und Management von Systems of Systems – Gestaltungskonzepte im SoS-Engineering. In: Koch, W.; Wilke, D.; Dreiseitel, S.; Kaffenberger, R. (Hrsg.): Tag des Systems Engineering. 15. - 17. November 2023, Würzburg, 2023, S. 137–143
- [WMG+24] WIECHER, C.; MANDEL, C.; GÜNTHER, M.; FISCHBACH, J.; GREENYER, J.; GREINERT, M.; WOLFF, C.; DUMITRESCU, R.; MENDEZ, D.; ALBERS, A.: Model-based analysis and specification of functional requirements and tests for complex automotive systems. In: Systems Engineering. 27 (4), 2024, S. 728–744
- [GSA+24] GÜNTHER, M.; SEIDENBERG, T.; ANACKER, H.; DUMITRESCU, R.: Principles for the design of system of systems exemplified using modularisation. In: Proceedings of the Design Society. 18th International Design Conference, 20. - 23. Mai 2024, Dubrovnik, Kroatien, 4, 2024, S. 2565–2574

Rahmenwerk zur musterbasierten Gestaltung von System of Systems

Inhaltsverzeichnis

Seite

1	Einleitung	1
1.1	Problematik.....	1
1.2	Zielsetzung	2
1.3	Vorgehen	3
2	Problemanalyse	6
2.1	Verständnis und Begriffsdefinitionen	6
2.2	Von mechanischen Systemen zu komplexen SoS	7
2.2.1	Der Systembegriff.....	8
2.2.2	Wandel hin zu vernetzten intelligenten technischen Systemen	9
2.2.2.1	Von mechatronischen Systemen zu intelligenten technischen Systemen.....	9
2.2.2.2	Vernetzte intelligente technische Systeme (SoS).....	12
2.2.3	Wandel hin zu vernetzten Produktionssystemen (Industrie 4.0).....	16
2.2.4	Herausforderungen von System of Systems	18
2.3	Das Engineering im Kontext von SoS.....	20
2.3.1	Grundlagen des Systems Engineering	21
2.3.2	Vorgehensmodelle im Engineering	25
2.3.3	Model-based Systems Engineering.....	28
2.3.4	SoS-Engineering	31
2.4	Lösungswissen im Gestaltungsprozess	35
2.4.1	Wissensmanagement.....	35
2.4.2	Wissen in Form von Lösungsmustern	37
2.4.3	Wissen in Form von Designprinzipien	41
2.5	Problemabgrenzung	42
2.6	Anforderungen an das Rahmenwerk	45

3	Stand der Forschung.....	47
3.1	Ansätze zur Beschreibung von SoS	47
3.1.1	Charakterisierungen von SoS	47
3.1.1.1	Charakterisierung nach MAIER.....	47
3.1.1.2	Charakterisierung nach BOARDMAN und SAUSER	49
3.1.1.3	Merkmale nach ABBOTT	50
3.1.2	Klassifikationen von SoS.....	51
3.1.2.1	Taxonomie nach MAIER	52
3.1.2.2	Erweiterte Klassifikation nach DAHMANN und BALDWIN	53
3.1.2.3	Multidimensionale Klassifikation nach TEKINERDOGAN.....	54
3.2	Ansätze zur Gestaltung von SoS.....	56
3.2.1	Ansätze aus dem Forschungsbereich SoS	56
3.2.1.1	Wellenmodell der SoS-Entwicklung nach DAHMANN et al.	56
3.2.1.2	SoSE-Prozess nach SAGE und BIEMER	58
3.2.1.3	SoSE-Methodik nach ADAMS und KEATING	60
3.2.1.4	SoS Lifecycle Engineering nach FORTE et al.....	61
3.2.1.5	Rahmenwerk für SoS-Gestaltungsprozesse nach SHAKED und REICH	63
3.2.1.6	Modellbasierter SoSE-Ansatz nach BAEK et al.....	64
3.2.1.7	Rahmenwerk für die SoS-Synthese, -Bewertung und gesteuerte Entwicklung nach SELBERG und AUSTIN	66
3.2.1.8	Prozessrahmenwerk für SoS nach ISO 21840.....	68
3.2.2	Ansätze aus verwandten Forschungsbereichen.....	69
3.2.2.1	Planung von CPS nach WESTERMANN	69
3.2.2.2	Modellbasiertes Engineering von kollaborativen eingebetteten Systemen nach BÖHM et al.	71
3.2.2.3	Multiagenten-System Engineering Methode nach WOOD und DELOACH	73
3.2.2.4	Ansätze der Systemdynamik nach LANE	74
3.3	Wissensbasierte Ansätze zur Unterstützung der SoS-Gestaltung	75
3.3.1	Wissen in der SoS-Gestaltung	76
3.3.1.1	Modellierungsmuster für SoS-Architekturen nach INGRAM et al.	76
3.3.1.2	Rekonfigurationsmuster für SoS nach PETITDEMANGE et al. ...	77
3.3.1.3	Interoperabilitätsmuster nach BRÖRING et al.....	78

3.3.1.4	Muster für vertragsbasierte SoS nach FALDIK et al.....	79
3.3.1.5	Designprinzipien für resiliente SoS nach UDAY und MARAIS...	80
3.3.1.6	Naturbasierte Designprinzipien für SoS nach AZANI.....	81
3.3.1.7	Evolvabilitätsbezogene Prinzipien für SoS nach RICCI et al. ..	82
3.3.2	Wissensbasierte Ansätze der Produktentwicklung.....	83
3.3.2.1	Musterkatalog nach GAMMA et al.....	83
3.3.2.2	Systemarchitekturmuster nach CLOUTIER.....	84
3.3.2.3	Lösungsmusterbasierter Entwurf fortgeschrittener mechatronischer Systeme nach ANACKER.....	86
3.4	Bewertung und Handlungsbedarf	87
4	Rahmenwerk zur musterbasierten Gestaltung von SoS	91
4.1	Das Rahmenwerk im Überblick	91
4.2	Definition und Klassifizierung von SoS	93
4.2.1	Definition und Charakterisierung von technischen SoS	94
4.2.2	Klassifizierung von SoS.....	96
4.3	Gestaltungsrahmen für das SoS-Engineering	100
4.3.1	Gestaltungskonzepte des SoS-Engineerings.....	101
4.3.1.1	Konzept 1: SoS-Initialisierungsgestaltung.....	105
4.3.1.2	Konzept 2: Managementgesteuerte SoS-Gestaltung	107
4.3.1.3	Konzept 3: Gremiengesteuerte SoS-Gestaltung	109
4.3.1.4	Konzept 4: Managementunabhängige SoS-Gestaltung	110
4.3.1.5	Konzept 5: SoS-Gestaltung aus Unternehmenssicht	112
4.3.1.6	Zusammenfassung der Gestaltungskonzepte	114
4.3.2	Methoden für die Gestaltung von SoS.....	115
4.3.2.1	Konzipierung von abgrenzbaren SoS (Konzept 2)	118
4.3.2.2	Rollenbasierte Konzipierung von SoS (Konzept 3)	119
4.3.2.3	Analyse von SoS-Zusammenarbeitsmodellen (Konzept 5) ..	120
4.3.3	Vorgehensmodell zur Gestaltung von SoS.....	121
4.3.4	Erweiterte Nutzung des Gestaltungsrahmens	126
4.4	Hilfsmittel für die wissensbasierte SoS-Gestaltung	128
4.4.1	Designprinzipien.....	130
4.4.2	Lösungsmuster für die SoS-Gestaltung.....	134
4.4.3	Musterbasierte Gestaltung	138

5	Demonstration und Evaluation.....	141
5.1	Anwendungsbeispiel 1: Automatisierter Überladevorgang im Smart Farming.....	141
5.1.1	Phase 1: Grobanalyse.....	142
5.1.2	Phase 2: Ermittlung SoS-Gestaltungskonzept	144
5.1.3	Phase 3: Musterbasierte Gestaltung	146
5.1.4	Erweiterte Betrachtung.....	151
5.2	Anwendungsbeispiel 2: Fehlerkommunikation innerhalb einer Verpackungslinie	156
5.2.1	Phase 1: Grobanalyse.....	157
5.2.2	Phase 2: Ermittlung SoS-Gestaltungskonzept	159
5.2.3	Phase 3: Musterbasierte Gestaltung	160
5.3	Bewertung der Anforderungserfüllung des Rahmenwerks.....	163
5.3.1	Bewertung der Anwendbarkeit und Effektivität auf Basis der Anwendung in der Praxis	164
5.3.2	Bewertung der Anforderungserfüllung des Rahmenwerks	166
6	Zusammenfassung und Ausblick.....	169
	Abkürzungsverzeichnis	175
	Literaturverzeichnis	177
	Anhang.....	189
A1	Ergänzungen zur Klassifizierung von SoS.....	191
A1.1	Übersicht der Anwendungsbeispiele	191
A1.2	Clusteranalyse anhand der Merkmale von SoS	195
A2	Ergänzungen zum Gestaltungsrahmen	199
A2.1	Methodensteckbriefe	199
A2.2	Designprinzipien.....	204
A2.3	Lösungsmuster.....	207

1 Einleitung

Die vorliegende Arbeit entstand im Rahmen der anwendungsorientierten Forschung am FRAUNHOFER INSTITUT FÜR ENTWURFSTECHNIK MECHATRONIK IEM. Die Ergebnisse resultieren aus verschiedenen Forschungs- und Industrieprojekten. Als wesentliches Projekt für die Forschungsarbeiten sei an dieser Stelle das vom BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG (BMBF) geförderte Verbundprojekt „MoSyS – Menschorientierte Gestaltung komplexer System of Systems“ genannt. In einem Konsortium aus 18 Forschungs- und Industriepartnern wurde das Ziel verfolgt, Methoden, Hilfsmittel und IT-Werkzeuge im Rahmen des Advanced Systems Engineering zu erforschen. Betrachtungsgegenstand war sowohl das technische Ökosystem von Produkten als auch die zugehörige Wertschöpfung. Die zugrundeliegende Dissertation ist Teil der in dem Forschungsprojekt erarbeiteten Methoden und Hilfsmittel und beschreibt ein *Rahmenwerk zur musterbasierten Gestaltung von System of Systems*. Die Evaluation der Arbeit erfolgte sowohl im Rahmen des Forschungsprojekts als auch in Kooperation mit weiteren Industriepartnern.

1.1 Problematik

Die Veränderung der Gesellschaft und der technischen Systeme sind Komplexitätsfaktoren, die Unternehmen vor große Herausforderungen stellen. Ein wesentlicher Treiber ist die Digitalisierung, welche sowohl die Funktionalität der technischen Systeme und ihre Vernetzungsfähigkeit erweitert als auch die Entwicklung der Systeme verändert. Neben der Digitalisierung beschreibt die INCOSE SE Vision 2035 weitere Megatrends, wie die „Vernetzte Welt“ oder „Industrie 4.0“. Zentrales Element dieser Trends sind voneinander abhängige Systeme, welche miteinander interagieren, um ein gemeinsames Ziel zu erreichen [INC21-ol].

Die Interaktion von Systemen in Systemverbünden zur Erreichung eines übergeordneten Gesamtziels, welches durch ein einzelnes System nicht erreicht werden kann, wird auch **System of Systems** (SoS) genannt [ISO21841, S. 2]. Diese Systemverbünde charakterisieren verschiedene Merkmale, die sie von eigenständigen Systemen unterscheiden. Beispielsweise werden die Systeme im SoS von unterschiedlichen Unternehmen entwickelt und betrieben und eine übergeordnete Entscheidungsinstanz fehlt [Mai98, S. 27]. Im Gegensatz zu eigenständigen Systemen findet die Vernetzung dynamisch statt und Systemgrenzen verschieben sich.

Die **Herausforderungen für Unternehmen**, die es bei der Gestaltung von SoS zu bewältigen gilt, sind vielfältig. Dies umfasst u. a. die Notwendigkeit zur Kooperation über Unternehmensgrenzen hinweg und die Lösung der damit einhergehenden Zielkonflikte, die Abstimmung von Schnittstellen zwischen Systemen aber auch die stetige Anpassung an die Veränderungen der Umgebung. Bereits heute werden die Unternehmen mit den gestiegenen Anforderungen der dynamischen Vernetzung konfrontiert. Dies wird nicht

zuletzt durch den Cyber Resilience Act der Europäischen Kommission deutlich. Dieser stellt Sicherheitsanforderungen an vernetzte Systeme und schreibt u. a. vor, dass Systeme auch im Betrieb noch über fünf Jahre durch entsprechende Maßnahmen beobachtet und ggf. erweitert und weiterentwickelt werden müssen, um die Sicherheit zwischen den vernetzten Systemen zu gewährleisten [Eur22-ol, S. 39]. Um die Anforderungen zu erfüllen, müssen Unternehmen die erweiterte Lebenszyklusbetrachtung und die Erweiterung des Betrachtungsfokus über die Systemgrenzen hinaus als integralen Bestandteil im Engineering etablieren.

Neben der zunehmenden Systemkomplexität und den gleichzeitig zunehmenden Herausforderungen im Engineering, erfordern stetig steigende Marktanforderungen und kürzere Produktlebenszyklen eine höhere Effizienz im Engineering. Um dies zu erreichen und die Komplexität zu bewältigen, wird die Wiederverwendung von bestehenden Artefakten und damit die **Wiederverwendung von Wissen** immer relevanter. Ein geeignetes Management von Wissen spielt für Unternehmen eine entscheidende Rolle als Basis für Innovationen und Erfolg [AES+12, S. 25], [PRR10, S. 4]. Ein vielversprechender Ansatz für die Operationalisierung von Wissensmanagement sind Lösungsmuster. Ein Lösungsmuster beschreibt ein wiederkehrendes Problem sowie die grundlegende Lösung für dieses Problem. Es kann beliebig oft eingesetzt werden, ohne dass sich die Lösung wiederholt [AIS+95, S. 10]. Eine Vielzahl von dokumentierten Mustern gibt Entwicklerinnen und Entwicklern¹ Impulse, ihre Denkmuster mit dem Wissen des Kollektivs zu reflektieren und zu erweitern. Gleichzeitig bieten Lösungsmuster eine Kommunikationsbasis, da sie Zusammenhänge in abstrakter Weise explizieren und so für ein einheitliches Vokabular sorgen [Ris07, S. 49ff.]. Die Gestaltung von SoS ist geprägt durch den Austausch von Informationen über Unternehmensgrenzen hinweg, sodass ein geeigneter Wissensansatz ein wichtiger Faktor zur Gestaltung von SoS darstellt.

Insgesamt zeigt sich, dass der Erfolg der heutigen Systeme maßgeblich von ihrer Integration und Interaktion innerhalb des SoS abhängt. Die komplexe Gestaltung dieser Systeme ist verbunden mit einer unternehmensübergreifenden Betrachtung der Interaktion von Systemen unter kontinuierlichen Veränderungen der Rahmenbedingungen. Es besteht Bedarf, dieses komplexe Gestaltungsumfeld durch methodische Hilfsmittel zu unterstützen.

1.2 Zielsetzung

Das Ziel der vorliegenden Arbeit ist ein *Rahmenwerk zur musterbasierten Gestaltung von System of Systems*. Das Rahmenwerk soll Unternehmen sowie Kooperations- und Interessensgemeinschaften bei der Planung, Analyse und Entwicklung von Systemverbünden und den interagierenden Systemen unterstützen.

¹ Aus Gründen der Lesbarkeit werden im Folgenden nicht konsequent beide Geschlechter mit aufgeführt, sondern teilweise ausschließlich die männliche Form verwendet. Dennoch sind bspw. bei der Verwendung des Begriffs Entwickler auch Entwicklerinnen gemeint.

Das Rahmenwerk setzt sich aus drei Bestandteilen zusammen. Erster Bestandteil ist eine Definition und Charakterisierung von System of Systems. Diese soll das Verständnis von SoS vereinheitlichen und die unterschiedlichen Ausprägungen von System of Systems strukturieren. Der zweite Bestandteil umfasst einen Gestaltungsrahmen und ein Vorgehensmodell. Der Gestaltungsrahmen strukturiert die unterschiedlichen Arten der SoS-Gestaltung und stellt jeweils passende Leitlinien und Hilfsmittel bereit. Das Vorgehensmodell unterstützt die Anwendung des Gestaltungsrahmens und beschreibt die Tätigkeiten, die es zur Auswahl eines geeigneten Gestaltungskonzepts benötigt. Dafür wird zunächst die Ausgangsbasis in Form von einem Grobkonzept skizziert. Anschließend erfolgt die Auswahl anhand eines kurzen Fragebogens und die anschließende Anwendung des passenden Gestaltungskonzepts. Als dritter Bestandteil unterstützen Hilfsmittel die Gestaltung von SoS. Diese werden in Form von Lösungsmustern und Designprinzipien zur Verfügung gestellt und anhand des Gestaltungsrahmens strukturiert.

1.3 Vorgehen

Die Arbeit hat das Ziel, Unternehmen bei der Gestaltung von SoS zu unterstützen und ist im Kontext der anwendungs- und gestaltungsorientierten Forschung verortet. Daher wird in der zugrundeliegenden Dissertation die Design-Science-Research-Methodology (DSRM) nach PEFFERS et al. angewendet. Diese unterstützt eine systematische Herangehensweise im Rahmen des Design Science und legt den Fokus auf die Relevanz der erarbeiteten Lösungen [PTG+06, S. 93].

Wesentliche Phasen der DSRM sind:

- 1) **Problemidentifizierung und Motivation:** In der ersten Phase wird das spezifische Forschungsproblem definiert und die Erarbeitung einer Lösung begründet.
- 2) **Zielsetzung einer Lösung:** Die Zielsetzung einer Lösung wird aus der Problemstellung abgeleitet und kann quantitativ oder qualitativ erhoben werden. Sie bezieht Problembeschreibungen und den aktuellen Stand der Forschung mit ein.
- 3) **Entwurf und Entwicklung:** In dieser Phase wird eine Lösung (Artefakt) erstellt. Dabei werden die Funktionalität und das grundlegende Konzept des Artefakts bestimmt sowie das eigentliche Artefakt unter Einbeziehung von Ressourcen, einschließlich des Wissens über die Theorie, erstellt.
- 4) **Demonstration:** Die Wirksamkeit der Lösung wird durch Experimente, Simulationen, Case Studies oder weiteren sinnvollen Aktivitäten demonstriert.
- 5) **Evaluation:** Die Unterstützung der Lösung bei Problemen wird durch Beobachtungen oder Messungen evaluiert. Durch die Nutzung der entwickelten Lösung können anhand der erzielten Ergebnisse Rückschlüsse u. a. auf die Anwendbarkeit und Effektivität der Lösung gezogen werden.

- 6) **Kommunikation:** In dieser Aktivität wird die Forschungsarbeit Forschern und Fachleuten kommuniziert. Dies kann u. a. in Form von Publikationen erfolgen [PTG+06, S. 94].

Die einzelnen Phasen werden sukzessive durchlaufen, wobei je nach Ausgangssituation und Wissensbasis der Einstieg in den Prozess unterschiedlich sein kann. Iterationen können sich durch zusätzliche Erkenntnisse auf Basis der Evaluation und Kommunikation der Ergebnisse ergeben. Die einzelnen Kapitel der Arbeit lassen sich den Aktivitäten der DSRM zuordnen. Bild 1-1 visualisiert den Standardprozess nach PEFFERS et al. und ordnet diesen den Kapiteln der vorliegenden Arbeit zu.

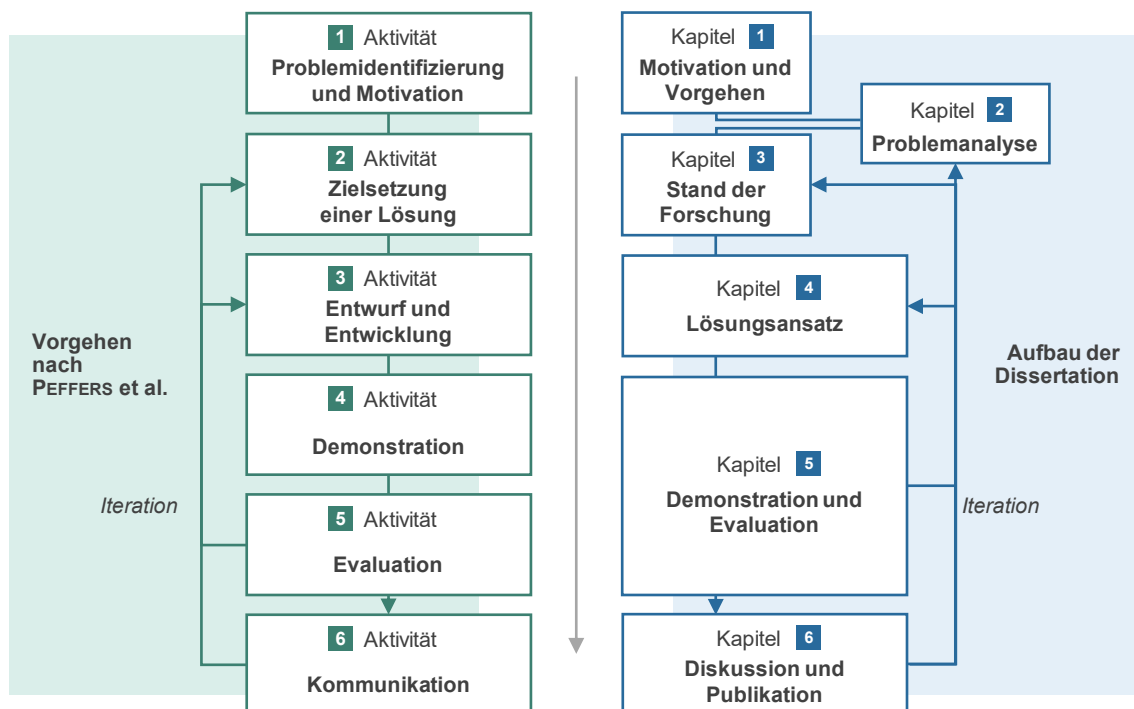


Bild 1-1: Zusammenhang zwischen dem DSRM-Vorgehen nach PEFFERS et al. und dem Aufbau der Arbeit [PTG+06, S. 93]

In **Kapitel 1** wird die Arbeit eingeleitet und die Motivation und das Vorgehen der Arbeit skizziert. Die in Kapitel 1 kurz umrissene Problematik und Zielsetzung wird in **Kapitel 2** vertieft. Die detaillierte Problemanalyse beschreibt die zugrundeliegende Thematik und die identifizierten Probleme. Anhand der betrachteten Handlungsfelder werden Anforderungen abgeleitet, die als Zielsetzung für die Entwicklung der Lösung gelten. Das **Kapitel 3** geht auf den Stand der Forschung ein und analysiert etablierte Lösungsansätze aus dem Forschungsbereich. Durch die Bewertung anhand der Anforderungen wird der Handlungsbedarf aufgezeigt und somit die Zielsetzung bestätigt. Der Kern der Arbeit ist der Lösungsansatz. Dieser wird in **Kapitel 4** beschrieben und umfasst mehrere zusammenhängende Artefakte, die im Einzelnen vorgestellt werden. Die Demonstration und Evaluation erfolgt in **Kapitel 5** anhand des „Automatisierten Überladevorgangs“ sowie der „Maschinenübergreifenden Fehlerkommunikation“. Abschließend werden in **Kapitel 6**

2 Problemanalyse

Ziel der Problemanalyse ist die Identifikation von Anforderungen an ein *Rahmenwerk zur musterbasierten Gestaltung von System of Systems*. Hierfür werden zunächst in **Kapitel 2.1** die für die Arbeit wesentlichen Begriffe definiert. **Kapitel 2.2** führt in das Themenfeld der technischen Systeme ein und erläutert die Herausforderungen aufgrund der technologischen Veränderungen der letzten Jahre. Dabei wird auf die Komplexität der technischen Systeme von heute eingegangen und welche Folgen sich aus der steigenden Vernetzung der Systeme ergeben. In **Kapitel 2.3** werden Herausforderungen beschrieben, die sich aus den Veränderungen der technischen Systeme für die Gestaltung ergeben. Neben etablierten Entwicklungsansätzen, wie dem Ansatz des Systems Engineering, wird auch auf die Besonderheiten für das Engineering durch die Entwicklung von Systemverbünden eingegangen. Im Fokus von **Kapitel 2.4** stehen wissensbasierte Ansätze, die bei der Lösung von komplexen Aufgaben unterstützen. **Kapitel 2.5** dient der Abgrenzung der zuvor skizzierten Problemstellungen, aus denen in **Kapitel 2.6** die Anforderungen an das Rahmenwerk abgeleitet werden.

2.1 Verständnis und Begriffsdefinitionen

Gegenstand dieser Arbeit ist ein *Rahmenwerk zur musterbasierten Gestaltung von System of Systems*. Ein **Rahmenwerk**¹ wird als eine strukturierte Grundlage verstanden, die den äußeren Rahmen für weitere Ausgestaltungen bildet. In diesem sind einzelne Elemente zur Lösung bestimmter Aufgaben geordnet. Es legt klare Richtlinien und Handlungsgrenzen fest und dient einer bestimmten Zielgruppe als Orientierungshilfe.

Der Begriff **Gestaltung** wird im Kontext dieser Arbeit synonym zu dem Begriff „Engineering“ verwendet. Das Engineering bezeichnet die Tätigkeit, bei der Methoden und Werkzeuge in einer strukturierten Art und Weise genutzt werden, um ein Produkt zu entwickeln [Wei06, S. 8]. Inzwischen geht das Begriffsverständnis über die reine Entwicklung hinaus und umfasst die gesamte Produktentstehung [DAR+21, S. 25]. Auch im Kontext von System of Systems umfasst das Engineering die Planung, Analyse, Entwicklung und Integration [ISO21841, S. 2]. Da der Begriff Engineering nicht eindeutig im Deutschen definiert ist, wird im Kontext dieser Arbeit der Begriff Gestaltung synonym verwendet [DAR+21, S. 25].

Muster sind Vorlagen für wiederkehrende Problemstellungen und erfassen bewährtes Lösungswissen. Als etabliertes Konzept in der Produktentstehung werden Lösungsmuster angewandt, um das Verständnis und die Effizienz bei der Lösung von Aufgaben zu erhöhen [CV07, S. 144]. Ein Lösungsmuster beschreibt ein „[...] immer wieder auftretendes Problem, beschreibt dann den Kern der Lösung für dieses Problem, und zwar so, dass

¹ Ein Rahmenwerk im Kontext dieser Arbeit ist von dem englischen Begriff „Framework“ zu unterscheiden, welches u. a. im Kontext der Architekturbeschreibung die Konventionen, Prinzipien und Praktiken für die Beschreibung von Architekturen definiert [ISO42010, S. 2].

man diese Lösung millionenfach anwenden kann, ohne sich je zu wiederholen" [AIS+95, S. X].

Als zentraler Betrachtungsgegenstand des Lösungsansatzes werden **System of Systems (SoS)** adressiert. Ein SoS bezeichnet einen Verbund aus mehreren Komponenten, die selbst als Systeme bezeichnet werden können [Mai98, S. 271]. Ein SoS zeichnet sich unter anderem dadurch aus, dass die interagierenden Systeme sowohl eigenständig Aufgaben erfüllen können als auch von unterschiedlichen Unternehmen gemanagt werden. Im Fokus dieser Arbeit stehen technische SoS, die sich von organisationalen oder auch soziotechnischen SoS unterscheiden. Eine detaillierte Beschreibung von SoS erfolgt in Kapitel 2.2.2.2.

2.2 Von mechanischen Systemen zu komplexen SoS

Der Wandel hin zu komplexen SoS basiert auf einer Vielzahl von Treibern im Rahmen des Megatrends der Digitalisierung. Fortschritte in der Informations- und Kommunikationstechnologie, die Elektrifizierung der technischen Systeme sowie neue Geschäftsmodelle sorgen für immer neue Möglichkeiten. Es können Daten und Informationen zwischen Systemen geteilt werden. Neue Software wird über Updates freigeschaltet und stellt den Kunden bedarfsgerechte Funktionen bereit. Zudem können neue Nutzungsmodelle, wie Sharing Economy, etabliert werden.

Diese neuen Möglichkeiten werden nicht zuletzt mit dem weiteren Megatrend hin zu mehr Nachhaltigkeit verbunden. Dieser Trend erfordert, dass technische Systeme immer effizienter genutzt werden, um Ressourcen zu schonen und Umweltauswirkungen zu reduzieren. Die Digitalisierung bietet hierbei eine Reihe von Möglichkeiten. Durch die Vernetzung von Systemen können Synergien geschaffen werden, beispielsweise durch die Kooperation von Systemen oder das Teilen von Ressourcen.

In der INCOSE SE Vision 2035 werden neben der Digitalisierung und Nachhaltigkeit weitere Megatrends aufgeführt, die durch die technischen Möglichkeiten der heutigen Zeit vorangetrieben werden. Die Trends hin zu einer „Vernetzten Welt“, „Industrie 4.0“ als auch „Smart Systems“ vereinen das Bild voneinander abhängiger Systeme, welche miteinander interagieren, um ein gemeinsames Ziel zu erreichen [INC21-ol, S. 3ff.].

Die beschriebenen Treiber führen zu immer komplexeren Systemen und Sachverhalten. Vor diesem Hintergrund erfolgt in den folgenden Abschnitten eine Beschreibung des Wandels hin zu intelligenten vernetzten Systemen. Dafür wird zunächst der Systembegriff eingeführt und anschließend die Charakteristika der heutigen Systeme sowohl aus Produkt- als auch Produktionssicht beschrieben.

2.2.1 Der Systembegriff

Der Begriff System bedeutet in seinem Wortursprung „aus mehreren Teilen zusammengesetztes und gegliedertes Ganzes“ [Dud23-ol]. Er wurde insbesondere durch die Arbeiten des Biologen LUDWIG VON BERTALANFFY und seinen Grundlagen zur allgemeinen Systemtheorie geprägt [Ber50], [Ber72, S. 411]. Die Theorie betont das Verständnis von Systemen unabhängig von ihrer spezifischen Disziplin und definiert allgemeine Konzepte, die auf verschiedene Systeme anwendbar sind. Der Begriff System und das damit verbundene Systemdenken ist eines der zentralen Konzepte des Systems Engineering. Demzufolge kann ein System als eine Gruppe von Elementen angesehen werden, die miteinander interagieren und zusammenarbeiten, um ein bestimmtes Ziel zu erreichen. INCOSE¹ definiert ein System wie folgt:

“A system is an arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not” [WSR+23, S. 2].

Entsprechend der Systemdefinition besteht ein System aus mehreren Elementen. Durch das Dekomponieren eines Systems in seine Systemelemente wird eine **Hierarchie** gebildet (Bild 2-1). Je nach Größe und Komplexität des Systems können sich mehrere Hierarchieebenen ergeben, sodass ein Systemelement wiederum als System mit Systemelementen betrachtet werden kann [ISO15288, S. 11]. Aus diesem Grund ist die Definition eines System of Interest (SoI) erforderlich, welches die Grenze des betrachteten Systems festlegt. Durch eine **Systemgrenze** ist dieses klar definiert und von seinem Umfeld abgegrenzt (Bild 2-1).

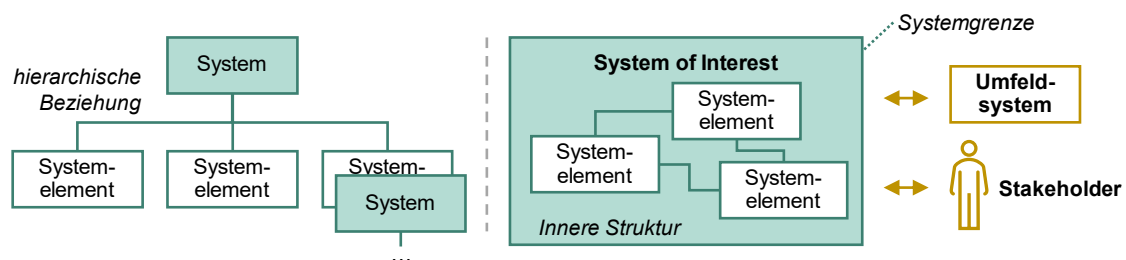


Bild 2-1: Hierarchiekonzept (links) und Konzept der Systemgrenze (rechts) in Anlehnung an ISO 15288 [ISO15288, S. 11ff.]

Die Elemente außerhalb der Systemgrenze stellen den Systemkontext dar. Dieser umfasst sowohl Umfeldsysteme als auch Stakeholder, wie den Nutzer eines Systems. Über Schnittstellen können diese mit dem SoI interagieren oder Anforderungen stellen [ISO15288, S. 13].

¹ INCOSE (International Council on Systems Engineering) ist eine gemeinnützige Organisation, die maßgeblich für die Entwicklung und Verbreitung der Prinzipien und Praktiken des Systems Engineering steht [INC23-ol].

Ein System kann physisch sein, wie beispielsweise ein mechanisches Gerät, oder abstrakt, wie zum Beispiel eine Organisation [WSR+23, S. 2]. Je nach Betrachtungsgegenstand lassen sich technische, organisationale oder auch soziotechnische Systeme unterscheiden, bei denen neben den technischen Systemen auch die Interaktion mit den Menschen berücksichtigt wird [Rop09, S. 47]. Im Folgenden werden technische Systeme als zentraler Betrachtungsgegenstand dieser Arbeit fokussiert.

2.2.2 Wandel hin zu vernetzten intelligenten technischen Systemen

Der Übergang von rein mechanischen zu mechatronischen Systemen markiert einen bedeutenden Fortschritt der technischen Systeme. Vor der Entwicklung der Elektrizität wurden technische Geräte ausschließlich durch die Bewegung von mechanischen Komponenten gesteuert. Mit dem Aufkommen der Elektronik und der Informationstechnik eröffneten sich jedoch neue Möglichkeiten und leistungsfähigere Systeme entstanden [Ise08, S. 1f.]. Inzwischen treiben Vernetzungstechnologien und die intelligente Nutzung von Daten die Entwicklung technischer Systeme [PH14]. Im folgenden Kapitel wird der Übergang von mechatronischen Systemen hin zu intelligenten technischen Systemen beschrieben und ihre charakteristischen Merkmale hervorgehoben.

2.2.2.1 Von mechatronischen Systemen zu intelligenten technischen Systemen

Mechatronische Systeme sind technische Systeme, die durch das Zusammenspiel von Mechanik, Elektronik und Informationstechnik entstehen. Sie bestehen aus einem Grundsystem, Sensorik, Aktorik und Informationsverarbeitung, wobei der Mensch und die Umgebung mit dem System interagieren können [VDI2206a, S. 14]. Dieser generischen Grundstruktur folgend, können eine Vielzahl an Systemen beschrieben werden [GTS+14, S. 27]. Das Grundsystem bilden die mechanischen, elektromechanischen, hydraulischen und pneumatischen Strukturen. Die Sensorik erfasst Zustandsgrößen des Grundsystems und der Umgebung. Über die Informationsverarbeitung werden alle relevanten Informationen ausgewertet und die Stellgrößen der Aktorik bestimmt. Die Aktorik setzt diese um und beeinflusst damit die Zustandsgrößen des Grundsystems [VDI2206a, S. 14f.].

Durch das Zusammenwirken der verschiedenen Disziplinen in einem mechatronischen System werden neue und verbesserte Funktionen realisierbar [Ise08, S. 1]. Ein Beispiel ist die Weiterentwicklung des Bremssystems im Auto. Wurden früher die Kräfte zum Bremsen über Hydraulikschläuche übertragen, realisieren heutzutage elektromechanische Komponenten und Kabelleitungen die Bremsfunktion. Gewicht und Wartungen können reduziert werden. Zusätzlich bieten ergänzende Sensoriken eine flexiblere und auf mehreren Informationen basierende Bremsfunktion, wie bspw. ein Antiblockier-System [VDI2206a, S. 7], [Ise08, S. 2].

Um die Leistungsfähigkeit mechatronischer Systeme kontinuierlich zu verbessern, werden zunehmend anspruchsvollere regelungstechnische, informationstechnische und mathematische Verfahren zur Erfassung und Verarbeitung von Informationen eingesetzt. Dadurch werden die Systeme flexibler und ihr Verhalten gewinnt zunehmend an Interaktivität. Die gleichzeitigen Weiterentwicklungen in der Kommunikations- und Informationstechnologie, u. a. das Internet der Dinge, führen zusätzlich zu einer steigenden Vernetzung [GDE+19, S. 87], [KWH13, S. 17]. Technische Systeme mit gewachsener Kognition, Kommunikations- und Vernetzungsfähigkeit werden auch als **intelligente technische Systeme (ITS)** bezeichnet. Aufbauend auf der Grundstruktur mechatronischer Systeme visualisiert Bild 2-2 die Referenzstruktur von ITS mit einer weiterentwickelten Informationsverarbeitung und der Vernetzung mit weiteren Systemen. Die Informationsverarbeitung wird durch das Drei-Schichten-Modell aus der Kognitionswissenschaft dargestellt [Str98, S. 96]. Aufbauend auf einfachen Reglern mit direkten Zusammenhängen, die zu den nicht kognitiven Regulierungen gehören, werden assoziative Regulierungen bis hin zu kognitiven Regulierungen dargestellt. Die oberste Stufe der Kognition umfasst Fähigkeiten, wie das Planen, Ziele modifizieren und Lernen [GDE+19, S. 86], [Str98, S. 97ff.].

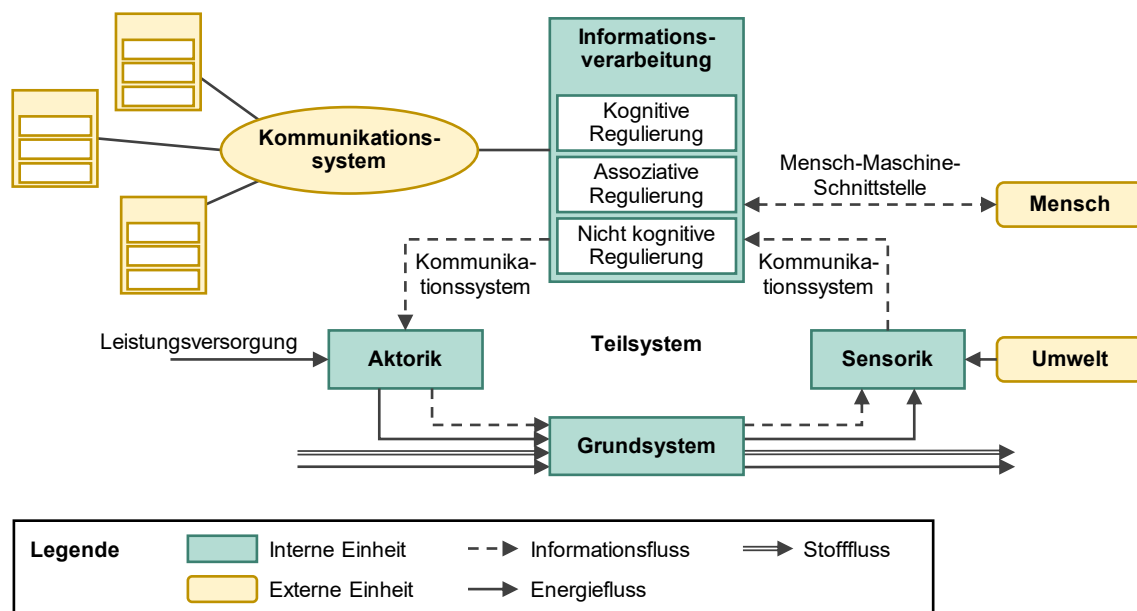


Bild 2-2: ITS auf Basis der Grundstruktur mechatronischer Systeme [GDE+19, S. 87]

ITS zeichnen insbesondere vier Merkmale aus [Dum10, S. 41], [GDE+19, S. 86]:

- **Adaptiv:** ITS interagieren mit ihrem Umfeld und können ihr Verhalten entsprechend dem Systemzweck autonom anpassen.
- **Robust:** ITS sind flexibel und autonom in einem dynamischen Umfeld und können unerwartete Situationen bewältigen und Unsicherheiten ausgleichen.

- **Effektiv:** ITS zeigen ein proaktives Verhalten, indem sie zukünftige Einflüsse antizipieren, Gefahren frühzeitig vermeiden und Ziele im Vergleich zu mechatronischen Systemen schneller und in besserer Qualität erreichen.
- **Benutzerfreundlich:** ITS berücksichtigen das Benutzerverhalten, passen sich an dieses an und verbessern dadurch die Bedienbarkeit. Zudem optimieren sie ihr eigenes Systemverhalten basierend auf historischen Benutzerdaten.

Die vier Merkmale von ITS beschreiben Systeme, die mit dem Umfeld und den Nutzern interagieren und sich diesem anpassen. Eine entscheidende Rolle spielt hierbei die Vernetzung zwischen ITS und Umgebung. Diese findet sowohl drahtgebunden als auch drahtlos auf lokaler und globaler Ebene statt. Die stetig steigende Vernetzung und der wachsende Datenaustausch zwischen Systemen treibt die Entwicklung von ITS voran und zeigt den Bedarf, den Betrachtungsfokus zu erweitern.

Der Begriff **Cyber-Physische Systeme (CPS)** erweitert den Grundgedanken von ITS und legt den Schwerpunkt auf die Vernetzung mit weiteren Systemen, Daten und Diensten. Insbesondere die Vernetzung mit dem Internet der Dinge und Dienste und die damit einhergehende Einbindung von neuen Technologien und Diensten zur Erweiterung des Kundennutzens kennzeichnen ein CPS [VDI2206b, S. 15]. Die Stoßrichtung der CPS geht einher mit dem Trend hin zum Internet der Dinge, Daten und Dienste und dem Begriff der Industrie 4.0 (Kapitel 2.2.3) [Kag14, S. 243], [KWH13, S. 17], [aca17-ol, S. 4ff.].

Auch der Begriff **System of Systems (SoS)** beschreibt Systeme, die sich miteinander vernetzen. Der Fokus liegt auf den Systemverbünden, die einen gewissen Autonomiegrad zwischen den interagierenden Systemen voraussetzen. Dies umfasst sowohl den Umstand, dass sie von unterschiedlichen Unternehmen entwickelt, hergestellt und gemanagt werden, als auch die Eigenständigkeit der Systeme in ihren Aufgaben [Mai98, S. 271], [WSR+23, S. 235]. Durch das Zusammenwirken in Systemverbünden werden Aufgaben und Mehrwerte erzielt, die ein einzelnes System nicht erzielen kann [Mai15, S. 20]. Die Interaktion von ITS in Systemverbünden hat jedoch zur Folge, dass ein Verhalten entstehen kann, welches nicht vollständig vorhersagbar ist. Der Begriff Emergenz bringt dies zum Ausdruck und ist ein weiteres Merkmal, welches die Forschungsrichtung von SoS prägt. Das emergente Verhalten hängt nicht zuletzt mit den Fähigkeiten der Adaptivität und Robustheit von ITS zusammen.

Sowohl CPS als auch SoS bauen auf den Grundkonzepten von ITS auf. Beide Stoßrichtungen erweitern den Betrachtungsgegenstand um den Aspekt der Vernetzung zwischen mehreren Einheiten (Bild 2-3). Dies hat zur Folge, dass sich die Systemgrenzen verschieben und der Betrachtungshorizont erweitert werden muss [HMG+91, S. 10], [FGD21, S. 2912]. SoS gehen dabei stärker auf die Autonomie der Systeme und die Herausforderungen in den Managementtätigkeiten ein, während der Fokus von CPS eher auf der Kombinationen von physischen Systemen mit der virtuellen Welt (Daten und Services) gelegt wird [Hen16, S. 52f.]. Im Folgenden wird der Fokus auf SoS und die damit einhergehenden Herausforderungen gelegt.

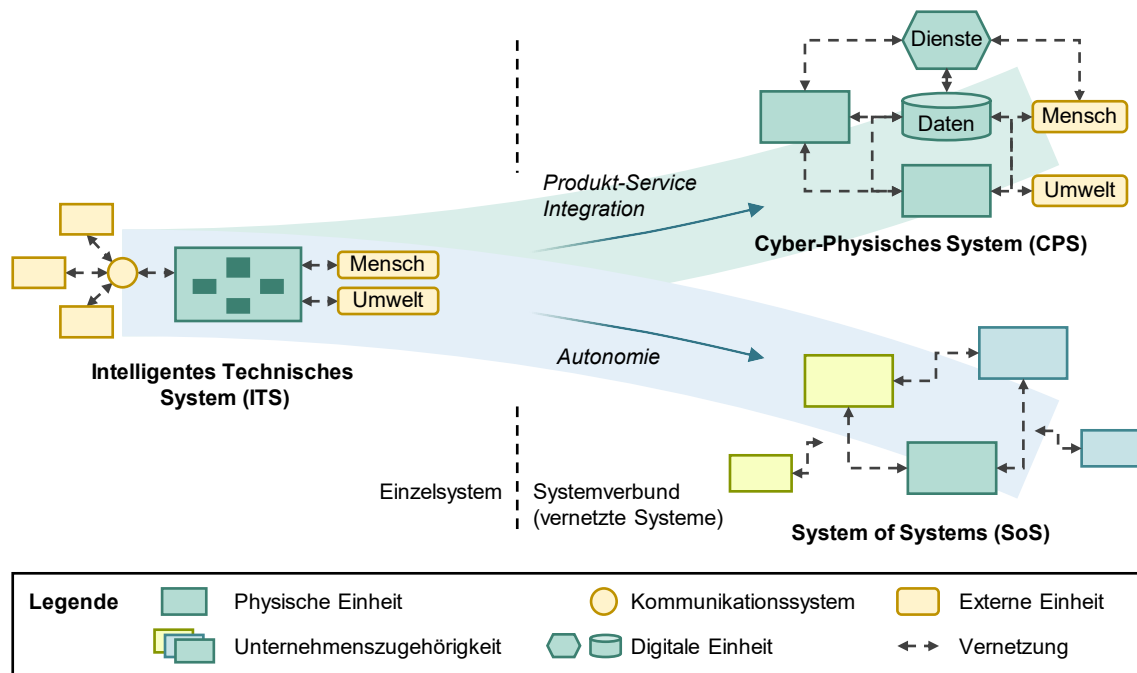


Bild 2-3: Cyber-Physische Systeme in Anlehnung an WESTERMANN [Wes17, S. 94] und System of Systems als Stoßrichtungen intelligenter technischer Systeme

2.2.2.2 Vernetzte intelligente technische Systeme (SoS)

Die zunehmende Digitalisierung sowie die Fähigkeiten von ITS sind die Basis für das Zusammenwirken von technischen Systemen in Systemverbünden. Diese werden durch Vernetzungstechnologien, wie Bluetooth oder 5G, aber auch die Zunahme von Sensorik in Systemen ermöglicht.

Die Interaktion von Systemen in Systemverbünden zur Erreichung eines übergeordneten Gesamtziels, welches durch ein einzelnes System nicht erreicht werden kann, wird auch **System of Systems (SoS)** genannt [ISO21841]. Ein einzelnes System, welches im SoS interagiert, wird auch als eigenständiges System¹ (ES) bezeichnet. Im Rahmen der allgemeinen Systemtheorie wurde das Zusammenspiel zwischen Systemen bereits früh in der Forschung betrachtet und spielt seitdem immer wieder eine Rolle [Bou56], [GSB08, S. 486]. Zurückgeführt werden kann der Begriff System of Systems im Zusammenhang mit technischen Systemen letztendlich auf die Strategic Defense Initiative der USA [Uni89, S. 16], [GSB08, S. 486]. Seitdem sind SoS neben dem Militärbereich auch im

¹ Im Englischen wird im Kontext von SoS der Begriff „constituent system“ genutzt. Der Begriff „Konstituierung“ wird im deutschen Sprachgebrauch jedoch eher im Rahmen von politischen und rechtlichen Prozessen eingesetzt, z. B. konstituierende Sitzung. Daher wird der Begriff „eigenständiges System“ genutzt, welcher die eigentliche Begriffsbedeutung widerspiegelt: „independent systems that forms part of a system of systems (SoS). Note: Constituent systems can be part of one or more SoS. Each constituent system is a useful system by itself, having its own development, management, utilization, goals, and resources, but interacts within the SoS to provide the unique capability of the SoS.“ [ISO21839, S. 2].

Kontext von Stromnetzen, Transportsystemen oder der Produktion immer relevanter geworden [NLF+15, S. 1].

Häufig wird im Zusammenhang mit Systemen, die über Grenzen hinweg kommunizieren und kooperieren, das Wort „Smart“ ergänzt [GDE+19, S. 86], [Mai98, S. 283]. So werden SoS-Beispiele häufig auch durch Smart X bezeichnet, wobei je nach Anwendungsgebiet das X ersetzt wird. Beispiele hierfür sind Smart Factory, Smart Grid oder auch Smart City [AFE19, S. 467], [DR16, S. 23], [AK18, S. 1]. Innerhalb der Anwendungsgebiete lassen sich zahlreiche Beispiele finden. Eines ist die Flottenfahrt, bei der sich beliebig viele Fahrzeuge miteinander vernetzen, um in einem geringen Abstand hintereinander herzufahren und Energie einzusparen [BBK+21, S. 2]. Je nach Anwendungsfall sind die Grenzen dabei fließend und Anwendungsgebiete überschneiden sich. Im Beispiel der Flottenfahrt ist die energiesparende Fahrt in engen Abständen sowohl im Kontext der Personenkraftfahrzeuge als auch im Transportbereich für LKW und Busse sowie deren Kombinationen zu betrachten. Bild 2-4 gibt einen Überblick über Anwendungsgebiete und deren Überschneidungen im Rahmen von Anwendungsfällen, wobei die Übersicht keinen Anspruch auf Vollständigkeit erhebt. Vielmehr soll es die Vielfalt der Anwendungsgebiete und die fließenden Grenzen zwischen SoS visualisieren.

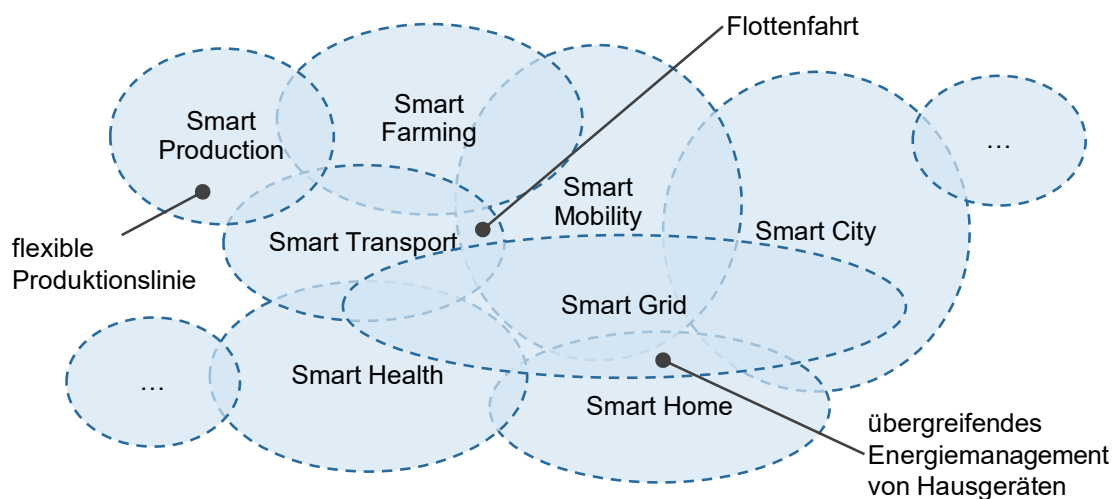


Bild 2-4: Übersicht von SoS-Anwendungsgebieten und -Anwendungsfällen

In der Literatur werden SoS in unterschiedlichen Abstraktionsgraden und Größen beschrieben. Von dem konkreten Anwendungsfall der Flottenfahrt bis hin zum sehr großen und abstrakten Smart Mobility Beispiel mit unzähligen Anwendungsmöglichkeiten und Teilnehmern, werden die unterschiedlichsten SoS genannt. ABBOT beschreibt diesen Zusammenhang damit, das SoS sowohl nach oben als auch nach unten keine Grenzen haben [Abb06, S. 44f.]. Ein Smart Home kann beispielsweise Teil einer übergeordneten Smart City sein oder sich auf gleicher Ebene mit dem Smart Mobility SoS vernetzen, um das Laden von E-Autos zu optimieren. Gleichzeitig kann ein Smart Grid Teil eines Smart

Homes sein oder vice versa. Auch können einzelne Systeme in mehreren SoS partizipieren [NL18, S. 113]. Eine hierarchische Betrachtung von SoS kommt so an seine Grenze. Daher sollten SoS eher als Netzwerk aufgefasst werden [Abb06, S. 41f.]

Die Beschreibung von SoS ist häufig nicht präzise [Axe19b, S. 31]. Um Klarheit in die Zusammenhänge im SoS zu bringen und SoS genauer zu beschreiben, wurden unterschiedliche Begriffe eingeführt. Beispiele sind *SoS-Konstellationen* nach AXELSSON oder *kollaborative Systemgruppen* nach BÖHM et al. [Axe19b, S. 32], [BBK+21, S. 20]. Beide Begriffe beschreiben konkrete Gruppen innerhalb eines dynamischen SoS, die sich durch die aktive Vernetzung und Interaktion von Systemen ergeben. Diese können auf konkrete Anwendungsfälle bezogen werden, wie im Beispiel der Flottenfahrt auf die Interaktion zwischen drei LKWs.

Neben den dynamischen Vernetzungen von Systemen im SoS finden weitere Veränderungen statt. Während der Laufzeit können neue Systeme hinzukommen, Systeme können ausgetauscht oder auch entfernt werden [KJT+13, S. 287f.], [MJ+19, S. 50f.]. Auch kann es Systeme geben, die sich zum Betrachtungszeitpunkt noch in der Entwicklung befinden [RHO12, S. 471f.]. Wie in Bild 2-5 visualisiert, sorgen sowohl die dynamischen Interaktionen, als auch die Weiterentwicklungen und Veränderungen einzelner Systeme im SoS dafür, dass keine klare Systemgrenze definiert werden kann [BBK16, S. 4]. Eine SoS-Grenze verändert sich [ISO21840, S. 43].

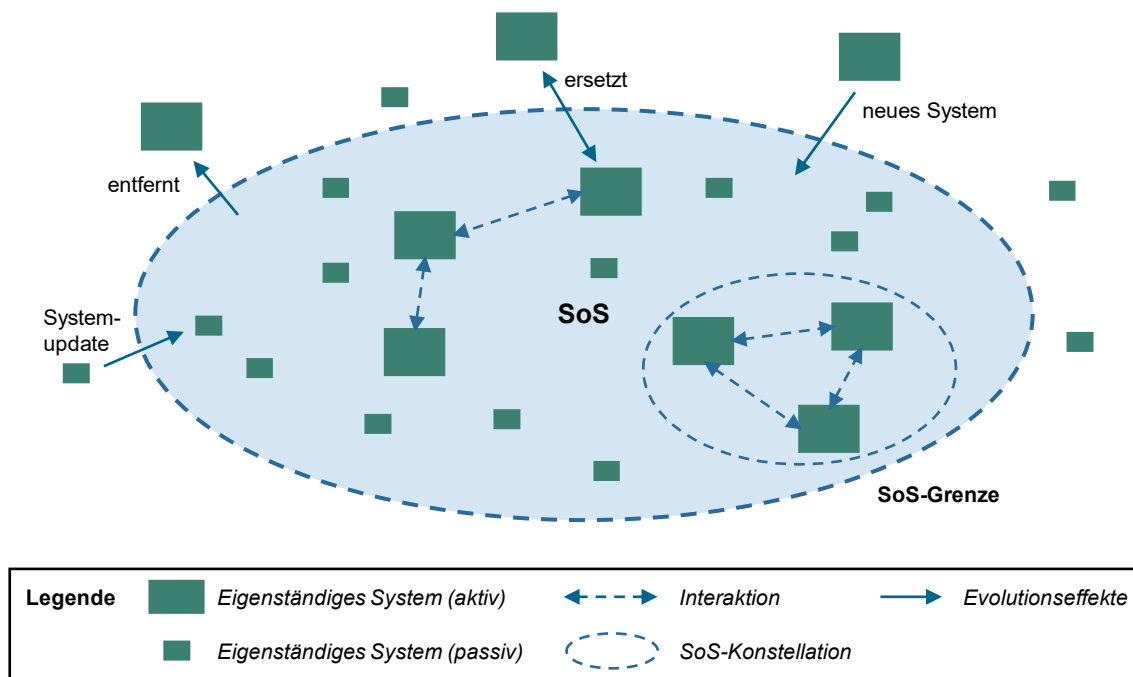


Bild 2-5: Dynamische und evolutionsbasierte Veränderungseffekte von SoS in Anlehnung an AXELSSON und KALAWSKY et al. [Axe19b, S. 32], [KJT+13, S. 287]

ITS sind adaptiv und robust (Kapitel 2.2.2.1). Durch ihre Interaktion in SoS können Phänomene auftreten, die nicht durch kausale Zusammenhänge auf Systemlevel erklärt wer-

den können [Axe22, S. 1070]. Ein Verhalten, welches nicht vollständig durch das Verhalten der einzelnen Systeme vorhersehbar ist und sowohl positive als auch negative Effekte hervorbringt, wird auch als emergent bezeichnet [HSM+23, S. 467f.], [KJT+13, S. 288]. Emergente Phänomene treten in unterschiedlichen Stärken auf. Je starrer die einzelnen Systeme miteinander verknüpft sind, desto geringer ist die Emergenz [HB07, S. 1815]. In dynamischen SoS mit stark unabhängigen Strukturen treten daher vermehrt Emergenzeffekte auf. MITTAL und RAINEY unterscheiden vier Typen von Emergenz: einfache, schwache, starke und unheimliche Emergenz [MR15, S. 3]. Die *einfache Emergenz* kann leicht durch Modelle mit geringer Komplexität vorhergesagt werden. *Schwache Emergenz* bezieht sich auf reproduzierbare Effekte, die durch Beobachtungen und Modelle mit reduzierter Komplexität verstanden, aber nicht konsistent im Voraus vorhergesagt werden können. Eigenschaften, die durch kein vereinfachtes Modell reproduzierbar sind und in Simulationen zu uneinheitlichen Ergebnissen führen, werden als *starke Emergenz* bezeichnet. Eine noch größere Komplexität sowie Eigenschaften, die nicht auf die Eigenschaften der einzelnen Systeme zurückzuführen sind, werden als *unheimliche Emergenz* bezeichnet [MR15, S. 3].

Bei starker Unabhängigkeit zwischen den eigenständigen Systemen im SoS, kann starke bis unheimliche Emergenz auftreten. In der größten Ausprägung von Unabhängigkeit wird auch von virtuellen SoS gesprochen [Mai98, S. 278]. Dieser Begriff bezeichnet SoS, die kein zentrales Management besitzen, willkürlich zusammenkommen und ein starkes emergentes Verhalten aufzeigen. Der Zusammenhang wird auch durch die Emergenz-Prinzipien nach HSU und BUTTERFIELD beschrieben [HB07, S. 1814f.]. Demnach verhält sich die Emergenz nicht-linear zu der Anzahl an beteiligten Systemen und antiproportional zu dem Grad der Bindung zwischen den Systemen. Dieser Effekt wird durch die Selbstorganisation von SoS noch verstärkt. Bild 2-6 visualisiert den Zusammenhang zwischen dem Unabhängigkeits- und Komplexitätsgrad von SoS und den Typen der Emergenz. Der Komplexitätsgrad¹ von SoS bezieht sich sowohl auf die Anzahl an beteiligten Systemen als auch die Dynamik der Vernetzung der Systeme untereinander.

In mehreren Beispielen zeigen sich die positiven Effekte der Interaktion in SoS. Beim vorausschauenden Fahren durch die Interaktion von Fahrzeugen mit der Verkehrsinfrastruktur oder bei automatisierten Übergabeprozessen in Lieferketten und zwischen Fertigungsanlagen in vernetzten Produktionen, werden positive Effekte wie Effizienzsteigerungen und Ressourcenschonung erzielt. Diese werden durch die Optimierung des Verkehrsflusses oder der Auslastungsrate der Systeme erreicht. Auch bei der Flottenfahrt wird durch die Interaktion zwischen LKWs ein positiver Effekt durch die Ausnutzung des Windschattens des vorausfahrenden Fahrzeuges erzielt. Dieser zeigt sich jedoch nur im Gesamtverbund. Das vorausfahrende Fahrzeug steht sowohl in Einzelfahrt als auch in der Flottenfahrt voll im Wind und kann keine Vorteile erzielen. Diesen Zielkonflikt muss ein SoS im Gesamtverbund lösen.

¹ Der Begriff Komplexität bezieht sich sowohl auf eine hohe Vielzahl und Vielfalt an Systemelemente als auch auf die hohe Veränderlichkeit und Dynamik der Interaktionen. [HWF+19, S. 11], [Sch05, S. 5f.]

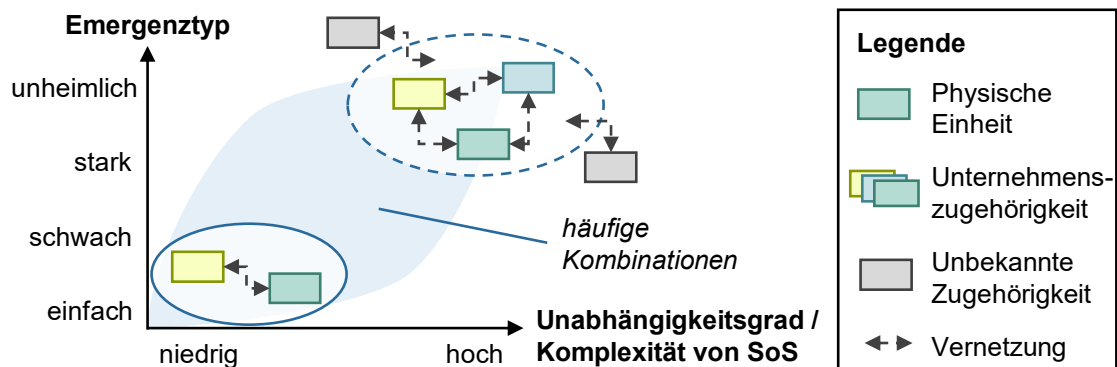


Bild 2-6: Emergenz von SoS in Anlehnung an HSU und BUTTERFIELD [HB07, S. 1814f.] und MITTAL und RAINEY [MR15, S. 3]

Neben den genannten Anwendungsbeispielen zeigen sich die Merkmale von SoS auch in der Produktion. Im Folgenden werden die Gemeinsamkeiten, aber auch die Besonderheiten von Produktionssystemen in Bezug auf SoS beschrieben.

2.2.3 Wandel hin zu vernetzten Produktionssystemen (Industrie 4.0)

Neben den Produkten sind auch die Produktionssysteme im Wandel. Von der ersten bis zur dritten industriellen Revolution wurden bereits kontinuierliche Produktivitätssteigerungen durch Mechanisierung, Elektrifizierung und den Einfluss von Informationstechnik erzielt [Kag14, S. 243]. Effiziente Arbeitsmaschinen, die durch Computerprogramme gesteuert werden, sind bereits seit einigen Jahren Stand der Forschung. Diese Produktionssysteme zeichnen sich durch eine hohe Produktivität und Flexibilität im Arbeitsprogramm aus [GP14, S. 3f.]. Seit einigen Jahren befindet sich die Industrie in der vierten industriellen Revolution. Diese steht für eine neue Ära der Produktion, in der Maschinen und Systeme miteinander kommunizieren und autonom Entscheidungen treffen können. Maschinen, Betriebsmittel, Produkte/Werkstücke sowie Lager und Transportsysteme werden zu leistungsfähigen Wertschöpfungsnetzwerken [GDE+19, S. 87]. Durch die Vernetzung und den Einsatz von Technologien wie dem Internet of Everything, künstlicher Intelligenz und Big Data werden Produktionsprozesse effizienter, flexibler und intelligenter gestaltet [Kag14, S. 243f.]. Das Paradigma der Industrie 4.0 ermöglicht es Unternehmen, ihre Produktionssysteme an die Anforderungen der globalen Märkte anzupassen und somit wettbewerbsfähig zu bleiben. Ein Beispiel ist die Factory 56 von Mercedes Benz. Diese Smart Factory setzt die Vision um und vernetzt sämtliche Maschinen und Automatisierungssysteme für eine flexible und effiziente Produktion von individualisierten Produkten [Moc21, S. 227].

Ähnlich zu ITS zeigen auch Produktionssysteme der Industrie 4.0 Merkmale von SoS auf (vgl. Kapitel 2.2.2). Neben der Vernetzung und der dynamischen Interaktion sind unterschiedliche Unternehmen an den Wertschöpfungsketten beteiligt. Produktionssysteme

kennzeichnen eine starke Heterogenität hinsichtlich der Technologien und Anbieter [KAG+16, S. 3]. Um Anwendungsszenarien wie die „Auftragsgesteuerte Produktion“ oder die „Selbstorganisierende adaptive Logistik“ umzusetzen, ist eine herstellernerneutrale Einbindung von Systemen in die Produktionsnetzwerke notwendig [ABD+16, S. 6ff.], [EKM17, S. 47f.]. Damit verschiedene Maschinen und Systeme von unterschiedlichen Herstellern Daten austauschen können, müssen sie interoperabel sein [Moc21, S. 121]. Interoperabilität wird unterschieden in:

- **Strukturelle Interoperabilität:** Die strukturelle Interoperabilität bezieht sich auf die Fähigkeit, Daten von einem System auf ein anderes zu übertragen. Dies wird durch die Verwendung von Anschlüssen, Bus-Systemen und Protokollen wie CAN, USB, http und TCP/IP ermöglicht [Moc21, S. 121].
- **Syntaktische Interoperabilität:** Die syntaktische Interoperabilität bezieht sich auf die Datenstrukturierung und ermöglicht die Identifikation von Informationen innerhalb von Daten. Dies wird durch die Verwendung von Standards und Formaten erreicht, bei denen definiert ist, wo im Datensatz welcher Wert steht [Moc21, S. 121].
- **Semantische Interoperabilität:** Die semantische Interoperabilität bezieht sich auf die Bedeutung von Informationen. Ein gemeinsames Verständnis der Informationen wird u. a. durch die Verwendung von Klassifikationen, Ordnungssystemen oder Nomenklaturen erreicht [Moc21, S. 121].

Standardisierung ist eine notwendige Voraussetzung für die Interoperabilität von Produktionssystemen der Industrie 4.0 [KAG+16, S. 4]. Insbesondere in der heterogenen und spezialisierten Unternehmenslandschaft in Deutschland ist die nahtlose Zusammenarbeit über Unternehmensgrenzen hinweg erforderlich. Die Plattform Industrie 4.0 stellt im Leitbild 2030 die Interoperabilität als eines von drei Handlungsfeldern in den Vordergrund [BMW19-ol, S. 1f.]. Zur Erreichung des Ziels organisieren sich die Unternehmen in unterschiedlichen Organisationen und Initiativen [KAG+16, S. 9]. Ein Beispiel ist die OPC Foundation. Mit mehr als 900 Mitgliedern aus der ganzen Welt entwickelt die Organisation eine plattformunabhängige Kommunikationsinfrastruktur, um die Interoperabilität zwischen Automatisierungslösungen zu ermöglichen [OPC23-ol], [EM11, S. 397]. Mit der OPC Unified Architecture (OPC UA) wurde ein Standard entwickelt, der den Datenaustausch unabhängiger Systeme in autarken Systemverbünden ermöglicht [EM11, S. 402]. Zur Realisierung müssen zusätzlich spezifische Informationsmodelle abgestimmt werden, welche für die jeweilige Anwendungsdomäne die semantische Interoperabilität sicherstellt. Diese wurde beispielsweise für die Verpackungsindustrie durch die OPC UA PackML definiert und ist dort inzwischen als Standard etabliert [BHA+20-ol].

Das Beispiel OPC UA zeigt die Relevanz, sich in Organisationen und Interessensgemeinschaften zu organisieren. Die Abstimmung über Unternehmensgrenzen hinweg ist die Basis für die Interaktion in Systemverbünden. Dafür müssen Unternehmen auch mit Zu-

lieferern und Wettbewerbern Kooperationen eingehen und die Standardisierung vorantreiben. Dabei bestehen jedoch häufig Vorbehalte gegenüber Kooperationen in größeren Gremien bezüglich der Datensicherheit und der Gefahr des Wissensabflusses [KAG+16, S. 9].

Auch für die Interaktionsfähigkeit von Systemen im Rahmen von SoS sind Kooperationen essenziell. DAHMANN et al. beschreiben Vereinbarungen zwischen Unternehmen als entscheidend für den Erfolg von SoS [DRL+11, S. 27]. Neben der Organisation in Interessensgemeinschaften (Gremien) können auch direkte Verträge zwischen Unternehmen vereinbart werden. Diese Art der Kooperation ermöglicht die Nutzung proprietärer Lösungen, beispielsweise in Form von geschlossenen Standards. Die Anpassung an spezifische technische Bedarfe und höhere Gewinnpotentiale sind mögliche Vorteile für Unternehmen in der Industrie 4.0 [KAG+16, S. 8].

2.2.4 Herausforderungen von System of Systems

Die in Kapitel 2.2.2 und 2.2.3 beschriebenen Systeme und Produktionssysteme befinden sich in einem Wandel. Der zunehmende Anteil an Hardware und immer leistungsfähigerer Informations- und Kommunikationstechnologie führt dazu, dass die Systeme inzwischen (teil-)autonom agieren und sich dynamisch in Systemverbünden vernetzen (Bild 2-7). Ein solcher Systemverbund von unabhängigen Systemen, die zeit- und ortsabhängig miteinander interagieren, bringt der Begriff System of Systems zum Ausdruck [PH14]. Das Konzept von SoS ist mit verschiedenen Herausforderungen verbunden, die im Folgenden zusammengefasst werden.

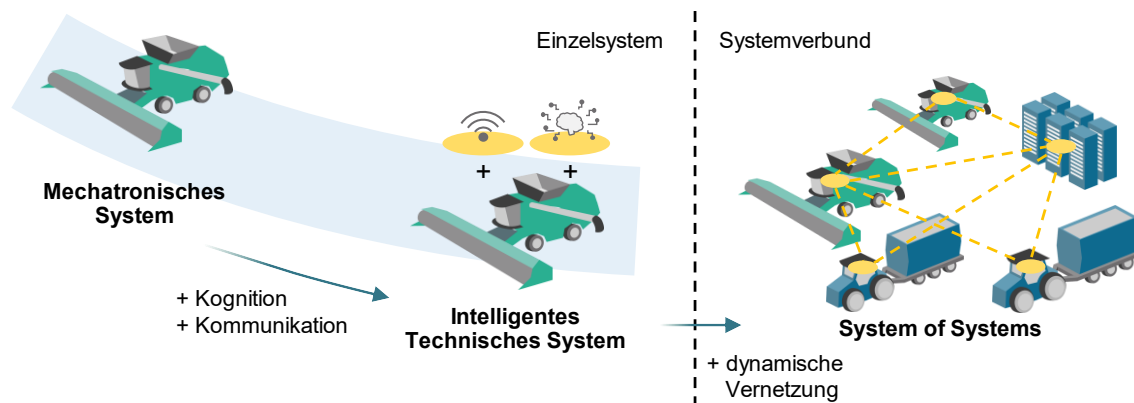


Bild 2-7: Wandel der technischen Systeme in Anlehnung an PORTER und HEPPELMANN [PH14]

Zunächst ist das Forschungsfeld von SoS sehr heterogen und eine *einheitliche Definition von SoS existiert nicht* [Jam08a], [NLF+15, S. 9], [AMY+18, S. 2800]. Dies sorgt dafür, dass Unternehmen die Herausforderungen nicht fokussiert angehen können. Nicht zuletzt führen die unterschiedlichen Sichtweisen und Ausprägungen der Anwendungsbeispiele hinsichtlich ihrer Komplexität und Abstraktionsgrade dazu, dass das Verständnis von SoS

in der Community unterschiedlich ist [Axe19b, S. 31]. Es besteht Bedarf an einer einheitlichen Definition und Charakterisierung von SoS.

Ausgehend von der Vernetzung zwischen unabhängigen Systemen werden mehrere Herausforderungen deutlich. Das *emergente Verhalten von SoS* entsteht durch die (teil-)autonome und dynamische Interaktion zwischen Systemen. Konnten mechatronische Systeme noch durch kausale Zusammenhänge innerhalb von klaren Grenzen beschrieben werden, stehen Ingenieure im Rahmen von SoS vor der Herausforderung, wie das Verhalten analysiert und vorausgesagt werden kann [HSM+23, S. 464]. Insbesondere bei der starken und unheimlichen Emergenz ist dies nicht mehr durch nachvollziehbare Zusammenhänge möglich. MOGUL beschreibt die Herausforderung der Emergenz als nicht lösbar:

“We will never be able to solve all emergent misbehavior problems, especially as system complexity increases.” [Mog06, S. 302]

Für die einfache und schwache Emergenz im Rahmen von weniger komplexen SoS, gilt dies nicht. Für diese Stufen der Emergenz gibt es erste Ansätze die Ingenieure nutzen können, um das SoS-Verhalten zu analysieren. Ein Beispiel ist die agentenbasierte Simulation [Axe22, S. 1081]. Dieser Ansatz abstrahiert Systeme zu Agenten, die bestimmte Fähigkeiten haben und Entscheidungen treffen können [Axe22, S. 1077]. Die unterschiedlichen Entscheidungen der Agenten und deren Abhängigkeiten untereinander werden dann simuliert und analysiert. Zur Analyse kombinieren bspw. BOYD et al. den agentenbasierten Ansatz mit Ansätzen aus der Spieltheorie, um die Entscheidungen der einzelnen Agenten nachzuvollziehen [BMS22, S. 114].

Dabei stellen auch *Zielkonflikte*, die aus der Interaktion unabhängiger Systeme resultieren, eine Herausforderung dar. Im Beispiel der Flottenfahrt würde sich keine Interaktion zwischen LKWs ergeben, wenn die eigenständigen Systeme nicht ein übergeordnetes Gesamtziel verfolgen würde. Das vorausfahrende Fahrzeug erzielt durch die Flottenfahrt keinen direkten Mehrwert. Ziele können sich in Einzelbetrachtung für ein System anders darstellen als im Rahmen der Interaktion im Systemverbund.

Dies gilt auch für die beteiligten Unternehmen. *Eigeninteressen*, wie ein wirtschaftlicher Profit, stehen häufig *im Konflikt mit den Interessen auf SoS-Ebene*. Wie in Kapitel 2.2.3 am Beispiel der offenen Standards im Rahmen der Industrie 4.0 erläutert, bietet ein offener Austausch zwischen Systemen häufig einen großen Vorteil für den Betrieb. Dieser steht jedoch im Konflikt mit dem Ziel einzelner Unternehmen Daten und Know-How zu schützen. Auch die Schaffung von Herstellerabhängigkeiten und Wechselbarrieren sind hier aufzuführen [DAR+21, S. 52]. Es ist wichtig, sowohl die Unternehmenssicht als auch die SoS-Sicht zu berücksichtigen.

Um die *Interoperabilität* zwischen unabhängigen Systemen zu gewährleisten, sind Abstimmungen zwischen Unternehmen notwendig. Neben den Industrie 4.0-Organisationen

sind auch in anderen Branchen Gremien und Interessensgemeinschaften zur Standardisierung von Schnittstellen gegründet worden. Beispiele sind die Agricultural Industry Electronics Foundation (AEF) für die Interaktion zwischen Traktor und Anbaugerät oder die CHAdeMO Association zur Standardisierung der Schnittstelle zwischen Ladeinfrastruktur und E-Fahrzeugen [AEF23-ol], [CHA21-ol]. Unternehmen partizipieren in den Gremien mit gemeinsamen Absichtserklärungen. Durch Verträge können Unternehmen auch direkte Kooperationen eingehen und bspw. geschlossene (proprietäre) Standards entwickeln. Solche proprietären Standards gehören einem Unternehmen und können gegen Lizenzvereinbarungen genutzt werden. Bild 2-8 visualisiert die unterschiedlichen Kooperationsformen.

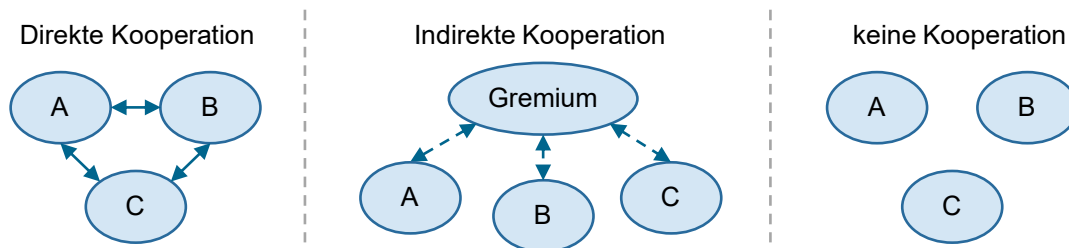


Bild 2-8: Kooperationsformen zwischen Unternehmen in SoS

Fazit: Die Erweiterung der Betrachtungsgrenze durch die Veränderung der Systeme von mechatronischen Systemen hin zu vernetzten ITS stellt Unternehmen vor große Herausforderungen [PH14]. Eine Betrachtung, in der Systeme klar von ihrer Umwelt abgegrenzt werden können, kommt durch die dynamische Vernetzung mit unabhängigen Systemen an seine Grenze. Während in klassischen Entwicklungsprojekten ein Entwicklungsteam die Auslegung eines Systems beeinflussen kann, ist diese Aufgabe im SoS verteilt und es sind mehrere Unternehmen beteiligt. Emergenz, die heterogene Systemlandschaft, die unterschiedlichen Unternehmen und die stetige Veränderung sind neben der wachsenden Komplexität weitere Herausforderungen, die Auswirkungen auf die Gestaltung der Systeme haben. Das nachfolgende Kapitel stellt den aktuellen Stand des Engineerings vor und geht auf die Herausforderungen der Gestaltung in Bezug auf SoS ein.

2.3 Das Engineering im Kontext von SoS

Technische Systeme funktionieren in großen Teilen durch das synergetische Zusammenwirken von Mechanik, Elektrotechnik und Informationstechnik. Insbesondere die in Kapitel 2.2 beschriebenen intelligenten technischen Systeme erfordern eine gute interdisziplinäre Zusammenarbeit in der Entwicklung. Um die Komplexität dieser Systeme abzubilden und den Austausch zwischen den Disziplinen zu adressieren, bietet das Systems Engineering einen vielversprechenden Ansatz. Systems Engineering baut auf der Systemtheorie (vgl. Kapitel 2.2.1) auf und betrachtet ein System als Ganzes. Das Systems Engineering ist ein interdisziplinärer Ansatz und berücksichtigt den gesamten Lebenszyklus eines Systems. Die wesentlichen Grundlagen des Systems Engineerings werden im folgenden Kapitel 2.3.1 vorgestellt.

Etablierte Vorgehensmodelle im Engineering, wie das V-Modell, geben seit Jahren einen Rahmen für die Entwicklung von Systemen vor. Vorgehensmodelle sind häufig abstrakt und nicht vollständig. Ein grundlegendes Verständnis über unterschiedliche Vorgehensmodelle ist daher insbesondere im heterogenen Feld von SoS relevant. Kapitel 2.3.2 stellt etablierte Vorgehensmodelle des Engineerings vor und bewertet diese hinsichtlich der Relevanz für SoS.

Mit der wachsenden Komplexität sind dokumentenzentrierte Ansätze an die Grenze der Leistungsfähigkeit gekommen. Aus dem Bedarf nach effizienten Entwicklungsansätzen hat sich das Model-based Systems Engineering entwickelt. Dieser erweiterte Ansatz des Systems Engineerings stellt Modelle in den Mittelpunkt der Arbeit. Dies gilt als essenziell, um die Vielzahl an Abhängigkeiten sowohl im System als auch im Entwicklungsgeschehen abzubilden. Die Grundlagen des Model-based Systems Engineerings werden in Kapitel 2.3.3 beschrieben.

Aufbauend auf den Grundlagen des Systems Engineering, befasst sich das SoS-Engineering (SoSE) mit der Vernetzung und Interaktion von Systemen in Systemverbünden. Das SoSE ist im Vergleich zum Systems Engineering ein noch junges Themenfeld und noch nicht bis ins letzte Detail erforscht. Kapitel 2.3.4 fasst die Herausforderungen des SoS-Engineerings zusammen und beschreibt erste Ansätze.

2.3.1 Grundlagen des Systems Engineering

Systems Engineering (SE) ist ein Ansatz, der in den 1940er Jahren von den Bell Telephone Laboratories geprägt wurde. Seit den 1950er Jahren hat sich eine Reihe von bewährten Verfahren und formalen Methoden des Systems Engineerings entwickelt, die in Normen und Handbüchern formalisiert wurden [HWF+19, S. v], [WSR+23, S. 3]. Ein zentrales Buch ist das Systems Engineering Handbuch der INCOSE. Zunächst als National Council on Systems Engineering (NCOSE) gestartet, wurde durch das starke internationale Interesse 1995 die INCOSE gegründet. Mit dem 2002 veröffentlichten ISO/IEC/IEEE 15288 Standard wurde das SE letztlich formal anerkannt [WSR+23, S. 3].

Das Ziel des Systems Engineering besteht darin, komplexe Systeme zu entwerfen, zu entwickeln und zu verwalten, indem sie als Ganzes betrachtet werden. Die ISO 15288 definiert Systems Engineering als:

„transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems using systems principles and concepts and scientific, technological and management methods” [ISO15288, S. 8].

Ein zentrales Prinzip des Systems Engineering ist das **Systemdenken**. Wie in Kapitel 2.2.1 beschrieben bestehen Systeme aus Systemelementen, die interagieren, um ein gemeinsames Ziel zu erreichen. Systeme stehen in Beziehung zu ihrer Umwelt, die durch

eine Systemgrenze von dem System abgegrenzt wird. Diese Denkweise ermöglicht, komplexe Zusammenhänge zu verstehen, indem ein ganzheitlicher Ansatz für Ursache-Wirkungsbeziehungen verfolgt wird [HWF+19, S. 3], [WSR+23, S. 1].

HABERFELLNER et al. führen weitere Aspekte des Systems Engineerings in dem sogenannten „SE-Männchen“ zusammen (Bild 2-9). Demnach ist neben dem Systemdenken ein **Vorgehensmodell** als Leitfaden relevant. Dieses gibt den grundsätzlichen Rahmen des Entwicklungsgeschehens vor und strukturiert die Aktivitäten. Das Vorgehen sollte vier Logiken folgen. Zum einen soll es *vom Allgemeinen zum Detail* führen. Zum anderen soll im Entwicklungsvorgehen das *Denken in Alternativen* etabliert und für Probleme immer mehrere mögliche Lösungen analysiert werden. In der Makro-Logik sollte das Vorgehen in *überschaubare Teilschritte* strukturiert werden. In der Mikro-Logik soll es in jeder Projektphase dem *Problemlösungszyklus* folgen. Dieser schreibt einfache Teilschritte von der Zielsuche und Problemdefinition über die Suche nach Lösungen bis zur Auswahl der Lösungen vor [HWF+19, S. 27ff.].

Ein **Problemlösungsprozess** wird durch fachliche Aktivitäten der Systemgestaltung und organisatorische Aktivitäten aus dem Projektmanagement unterstützt. Die *Systemgestaltung* setzt sich mit dem Problem und der Lösung der Probleme auseinander und umfasst die Architekturgestaltung und die Konzeptentwicklung. In der *Architekturgestaltung* wird die Prinzipiellösung entwickelt, welche in der *Konzeptentwicklung* ausdetailliert und umgesetzt wird. Das *Projektmanagement* umfasst organisatorische Aufgaben, wie die Planung, Lenkung und Steuerung des Projekts unter Berücksichtigung von Zeit, Kosten und verfügbaren Ressourcen. Das Systems Engineering betrachtet ein Projekt ganzheitlich und führt die Aktivitäten der Systemgestaltung und des Projektmanagements zusammen. Sämtliche Aktivitäten werden durch Methoden und Tools gestützt [HWF+19, S. 101ff.].

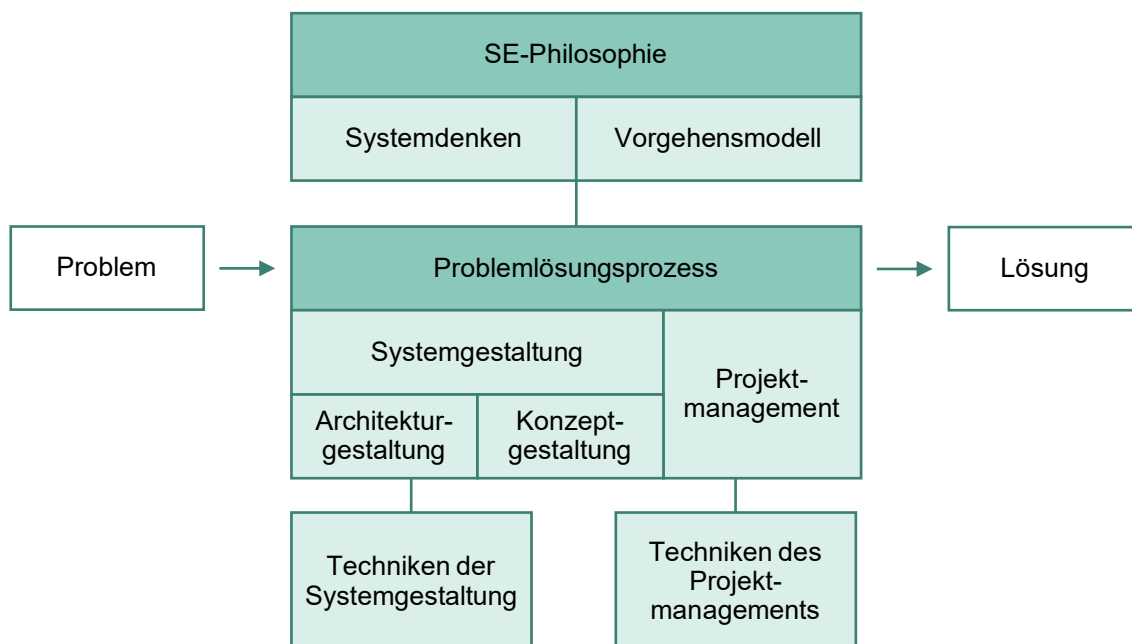


Bild 2-9: Das „SE-Männchen“ nach HABERFELLNER et al. [HWF+19, S. vi]

Darauf aufbauend müssen detaillierte Strukturen und Prozesse für ein ganzheitliches SE in Unternehmen implementiert werden. Die ISO 15288 gibt einen Prozessrahmen für das Systems Engineering vor und definiert vier Prozessbereiche, die der gleichzeitigen Betrachtung von Systemgestaltung und Projektmanagement folgen:

- 1) **Vertragsprozesse:** Diese Prozesse definieren Aktivitäten, die erforderlich sind, um eine Vereinbarung zwischen zwei Organisationen in Bezug auf die Lieferung und Beschaffung von Produkten oder Dienstleistungen zu erzielen.
- 2) **Organisatorische Unterstützungsprozesse:** Die projektunterstützenden Prozesse stellen auf strategischer Ebene die erforderlichen Ressourcen bereit, um die Bedürfnisse und Erwartungen von Stakeholdern der Organisation zu erfüllen.
- 3) **Technische Managementprozesse:** Die technischen Managementprozesse werden während des gesamten Lebenszyklus und auf allen Ebenen eingesetzt, um Pläne zu erstellen und zu entwickeln, diese auszuführen, die tatsächliche Leistung zu bewerten, den Fortschritt im Vergleich zu den Plänen zu überprüfen und die Ausführung bis zur Erfüllung zu kontrollieren.
- 4) **Technische Prozesse:** Diese Prozesse ermöglichen die Erstellung von Anforderungen und Systemlösungen, die die gewünschten Fähigkeiten unter Berücksichtigung von Leistung, Umgebung, externen Schnittstellen und Designeinschränkungen erfüllen [ISO15288].

Insgesamt werden 30 Prozesse im Detail beschrieben (Bild 2-10). Dabei werden je Prozess die Kerntätigkeiten aufgeführt sowie die Zusammenhänge zu vorgelagerten und nachgelagerten Prozessen erläutert.

Neben der ISO 15288 gibt es weitere Standards und Grundlagenliteratur, die unterschiedliche Schwerpunkte setzen [Cza19, S. 30ff.]. Eine Analyse nach HONOUR und VALERDI stellt die unterschiedlichen Standards gegenüber und konsolidiert die wesentlichen Kernthemen des Systems Engineering [HV06, S. 6ff.]:

- **Zieldefinition:** Im Zentrum dieser Tätigkeit steht das Planen und Klären der Aufgabe im Austausch mit relevanten Stakeholdern. Sie steht zu Beginn eines Projekts und ist der Ausgangspunkt für die Entwicklung eines Systems.
- **Anforderungsmanagement:** Das Anforderungsmanagement umfasst das Erfassen, Managen, Analysieren und Validieren von Anforderungen. Wesentliche Fähigkeiten, Charakteristiken und Qualitätsfaktoren des Systems werden hier definiert.
- **Architekturgestaltung:** Der Kern der Systemarchitekturgestaltung ist eine strukturierte Entwurfs- und Ausarbeitungsphase auf einem bestimmten Abstraktionslevel. Sie wird genutzt, um verschiedene Lösungsalternativen zu entwickeln und zu bewerten.

- **Implementierung:** Die Implementierung umfasst die technische Ausarbeitung der Systemarchitektur. Einzelne Systemkomponenten werden erstellt und zu einem Gesamtsystem integriert.
- **Technische Analyse:** Die Bewertung des Systems anhand der Anforderung und das Treffen von Trade-Off Entscheidungen ist Teil der technischen Analyse.
- **Verifikation und Validierung:** Die Verifikation weist nach, dass die Anforderungen erfüllt sind. Der Abgleich, ob das Ziel und die Stakeholderbedarfe erfüllt sind, wird durch die Validierung überprüft.
- **Technisches Management:** Das technische Management kümmert sich um die Planung, Überwachung und Steuerung aller Arbeiten, die erforderlich sind, um die Projektziele zu erreichen.
- **Beschaffungs- und Lieferantenmanagement:** Ergänzend zum technischen Management werden im Rahmen dieser Tätigkeit die Lieferanten- und Beschaffungsstrukturen gemanagt und Verträge verhandelt.

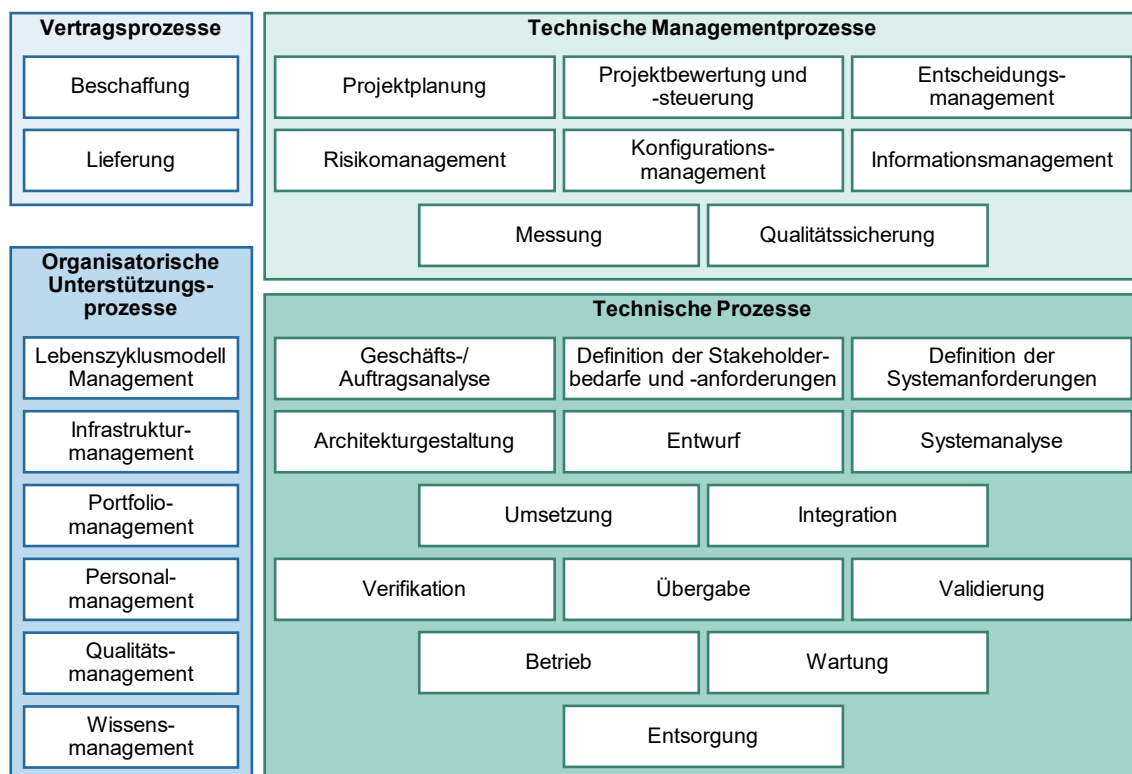


Bild 2-10: Systems Engineering Lebenszyklusprozesse nach ISO 15288 (Übersetzung nach GfSE) [ISO15288, S. 21], [GFS17]

Fazit: Die Konzepte des Systems Engineerings bieten ein Grundgerüst an Herangehensweisen für die Gestaltung von komplexen Systemen. Sie helfen Problemstellungen strukturiert und ganzheitlich anzugehen. Die in der ISO 15288 beschriebenen Prozesse sind detailliert und eine gute Orientierung zur ganzheitlichen Implementierung von Systems

Engineering in Unternehmen. Die Analyse der unterschiedlichen Grundlagen und Standards zeigt die Relevanz, sowohl technische Prozesse als auch Managementprozesse in Abhängigkeit zueinander zu adressieren. SoS-spezifische Managementaktivitäten, wie die unternehmensübergreifende Abstimmung über Entwicklungsvorgänge, werden jedoch nicht explizit berücksichtigt.

2.3.2 Vorgehensmodelle im Engineering

Systems Engineering ist ein interdisziplinärer Ansatz zur Entwicklung komplexer technischer Systeme. Im Rahmen des Systems Engineerings stellen HABERFELLNER et al. die Relevanz eines übergeordneten Vorgehens (Makro-Logik) heraus, welches Teilschritte innerhalb des Engineerings definiert (vgl. Kapitel 2.3.1) [HWF+19, S. 27ff.]. Gleichzeitig existieren eine Vielzahl an Vorgehen in der Produktentstehung, die sich je nach Zielstellung im jeweiligen Fall ausgeprägt haben. WYNN und CLARKSON haben in einer fundierten Analyse Vorgehensmodelle der Produktentstehung analysiert und drei Ebenen charakterisiert [WC18, S. 164]. Demnach konzentrieren sich Modelle der Makro-Ebene auf die gesamte Projektstruktur und den Entwurfsprozess. Modelle der Meso- und Mikro-Ebene gehen detaillierter auf die einzelnen Abläufe ein und stellen diese in einen unmittelbaren Zusammenhang. Ein Beispiel sind die in der ISO 15288 beschriebenen Prozesse inklusive der Beschreibung der In- und Outputs zwischen den Prozessen (Kapitel 2.3.1). Im Folgenden werden drei Modelle vorgestellt, die der Charakterisierung der Makro-Ebene nach HABERFELLNER et al. folgen: das V-Modell, iPeM und ein agiles Vorgehen [HWF+19, S. 27ff.], [WC18].

Das **V-Modell** nach VDI 2206 ist eine angepasste Version des V-Modells der Softwareentwicklung, die speziell auf mechatronische Systeme zugeschnitten ist. Es stellt einen strukturierten Entwicklungsprozess dar, der in Bild 2-11 visualisiert ist. Input für den Entwicklungsprozess ist ein Entwicklungsauftrag, der durch Anforderungen präzisiert wird und als Eingangsgröße in den linken Ast des V-Modells führt. Dieser repräsentiert den Systementwurf, in dem Anforderungen an das zu entwickelnde System definiert und daraus logische und physische Lösungen abgeleitet werden. Dazu gehört die Definition der Systemarchitektur und des Designs. Das Ergebnis des Systementwurfs ist ein interdisziplinäres Lösungskonzept. Am unteren Ende des V-Modells befindet sich der domänenspezifische Entwurf. Hier wird das interdisziplinäre Lösungskonzept weiter konkretisiert. Der rechte Ast des V-Modells repräsentiert die Systemintegration. Hier werden die einzelnen Ergebnisse der Fachdisziplinen zu einem Produkt zusammengeführt. Die Lösung wird kontinuierlich anhand der definierten Anforderungen überprüft und ihre Eigenschaften abgesichert. Die Modellbildung und -analyse ist ein unterstützender Aspekt des V-Modells, der in verschiedenen Phasen angewendet werden kann. Dabei werden Modelle genutzt, um das System zu beschreiben oder zu verifizieren. Durch mehrmaliges Durchlaufen des V-Modells wird das Produkt konkretisiert und unterschiedliche Reifegrade erreicht [VDI2206a].

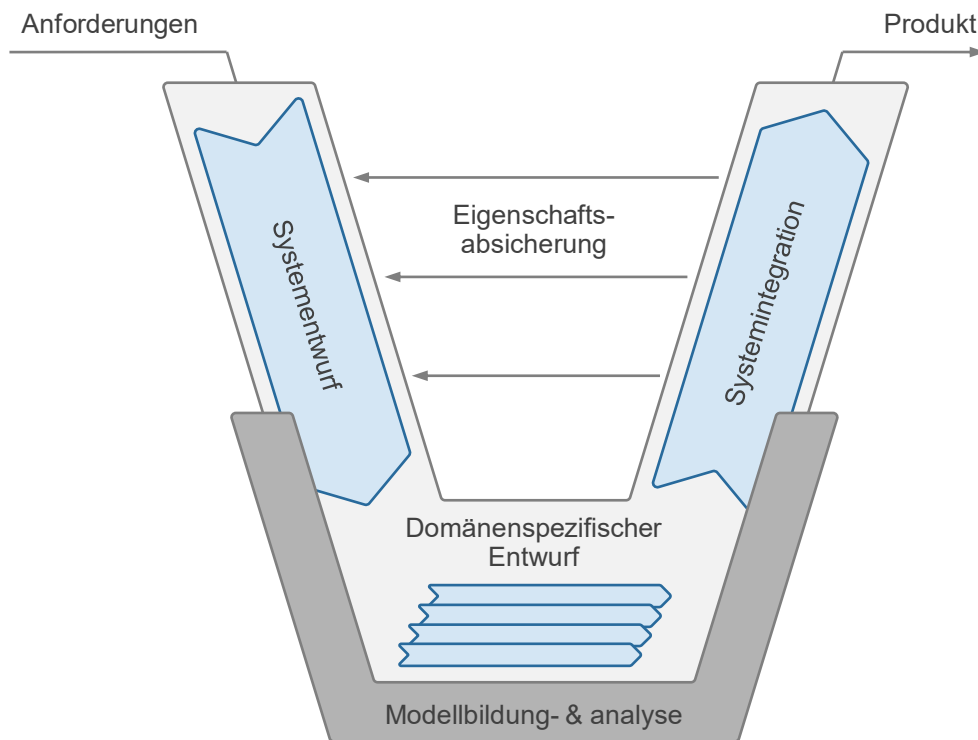


Bild 2-11: Das V-Modell nach [VDI2206a, S. 29]

Ein weiteres Makro-Modell ist das **Integrierte Produktentstehungsmodell (iPeM)** nach ALBERS et al. (Bild 2-12) [ARB+16, S. 101]. Es basiert auf der allgemeinen Systemtheorie nach ROPOHL und beschreibt die Produktentstehung als kontinuierliches Zusammenspiel von drei Systemen: dem Zielsystem, dem Handlungssystem und dem Objektsystem [Rop09], [ARB+16, S. 101]. Das Zielsystem stellt die Zielsetzung dar und wird durch das Handlungssystem in ein Objektsystem überführt. Dies geschieht anhand der Produktentstehungsaktivitäten mithilfe von Methoden, Prozessen und Ressourcen. Die Aktivitäten der Produktentstehung, wie bspw. Ideenfindung oder Validierung, geben den Handlungsrahmen vor.

Im Detail werden die einzelnen Aktivitäten anhand des Problemlösungszyklus SPALTEN durchgeführt. SPALTEN strukturiert sich ähnlich wie der Problemlösungszyklus nach HABERFELLNER et al. und umfasst sieben Schritte: (S) *Situationsanalyse*, (P) *Problemeingrenzung*, (A) *Alternative Lösungssuche*, (L) *Lösungsauswahl*, (T) *Tragweitenanalyse*, (E) *Entscheidung und Umsetzen* und (N) *Nachbereiten und Lernen*. Die insgesamt 70 Mikroaktivitäten werden als Tätigkeiten aufgefasst, die iterativ von Produktentwicklern durchgeführt werden [ARB+16, S. 101f.].

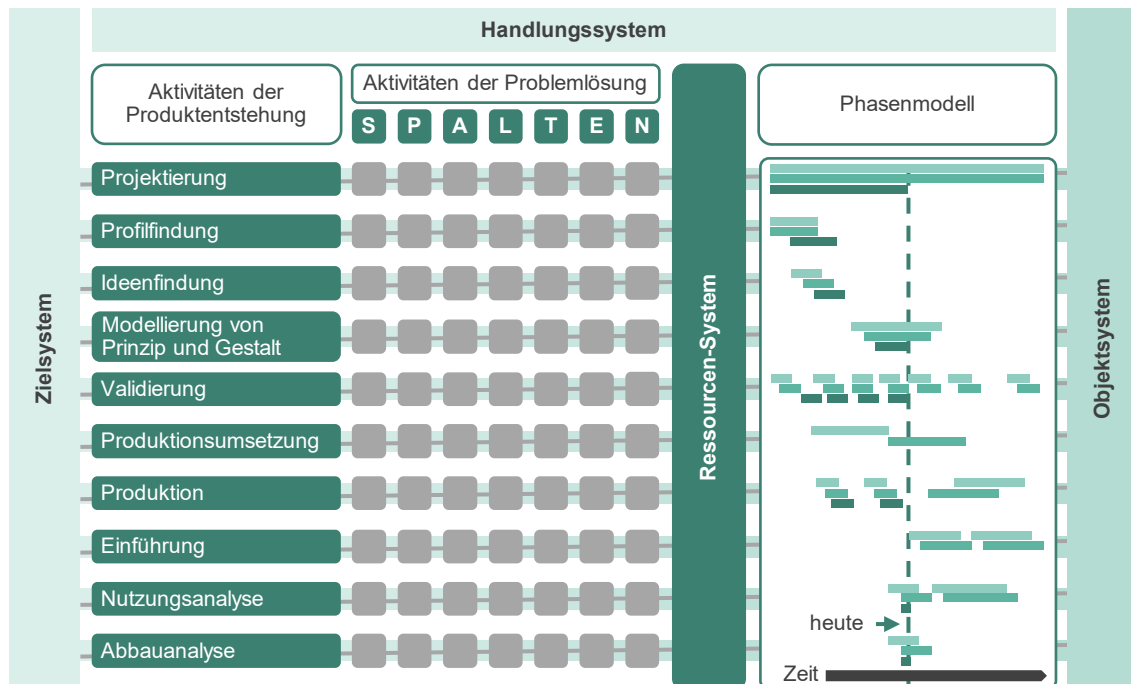


Bild 2-12: Das integrierte Produktentstehungsmodell (iPeM) nach ALBERS et al. [ARB+16, S. 101], Übersetzung nach ZINGEL [Zin13, S. 96]

In ihrer Arbeit stellen WYNN und CLARKSON weitere Vorgehensmodelle auf der Makro-Ebene vor. Im Fall von großen Unwägbarkeiten werden iterative Entwicklungsvorgehen hervorgehoben, die auf Lernzyklen aufbauen [WC18, S. 190]. Ein Beispiel ist der Agile Systems Engineering Ansatz nach TURNER [Tur07]. Aufbauend auf dem allgemeinen agilen Vorgehen sind einzelne Aktivitäten des Systems Engineerings als Lernzyklen zu verstehen (Bild 2-13). Im Kern werden je Iteration vorzeigbare Arbeitsprodukte erstellt, die relevanten Stakeholdern präsentiert werden. Im Anschluss wird in einer nächsten Iteration das Arbeitsprodukt angepasst und weiterentwickelt. Der Umfang solcher Arbeitsprodukte im Rahmen des Systems Engineering ist dabei noch Teil des wissenschaftlichen Diskurses und nicht genauer definiert.

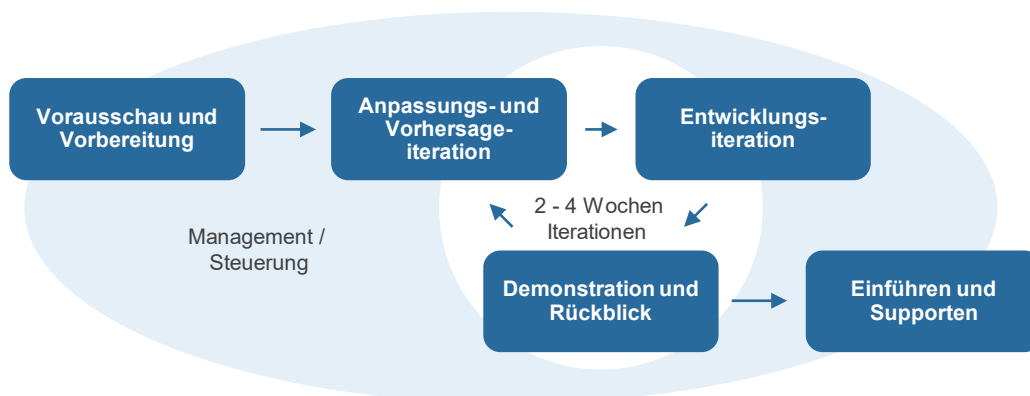


Bild 2-13: Allgemeines agiles Vorgehen nach TURNER [Tur07, S. 12]

Fazit: Die beschriebenen Vorgehen sind abstrakte Modelle der Makro-Ebene, die Unternehmen als Grundlage für die Entwicklung von komplexen mechatronischen Systemen heranziehen können. Alle drei Vorgehensmodelle adressieren eine interdisziplinäre Zusammenarbeit, die an unterschiedlichen Stellen stattfindet. Während iPeM und agile Ansätze die interdisziplinäre Zusammenarbeit in den aufgabenorientierten Teams vorsehen, sind im V-Modell insbesondere im Systementwurf sowie an den Übergängen ein interdisziplinärer Austausch vorgesehen. Alle Ansätze heben den iterativen Charakter des Vorgehens hervor. Dieser ist für komplexe Systeme essenziell, um anhand von neu erlangtem Wissen Entscheidungen anzupassen. Insbesondere der agile Ansatz nach TURNER bleibt sehr generisch und geht nicht auf konkrete Arbeitsinkremente im Kontext des Systems Engineerings ein. Demgegenüber detaillieren iPeM und das V-Modell mehrere Tätigkeiten, um ein System zu entwickeln. SoS-bezogene Tätigkeiten sind nicht Teil der Modelle. Vielmehr sorgen sie für eine Vielfalt an Entwicklungsvorgängen auf der Systemebene, die anschließend noch im heterogenen Gestaltungsfeld von SoS integriert werden müssen.

2.3.3 Model-based Systems Engineering

Die Tätigkeiten im Systems Engineering hängen miteinander zusammen und führen dazu, dass viele Informationen ausgetauscht werden müssen. Insbesondere für komplexe Systeme ist der effiziente Austausch von Informationen über sämtliche Aktivitäten hinweg essenziell. Während in der Vergangenheit das Vorgehen dokumentenbasiert stattgefunden hat, nimmt die Bedeutung der formalisierten Modellbildung im Rahmen des Systems Engineerings kontinuierlich zu [DAR+21, S. 29]. Modelle helfen ein zu entwickelndes System durchgängig zu beschreiben und zu analysieren, begonnen bei der frühen Phase der Konzipierung über den gesamten Produktlebenszyklus hinweg. Als erweiterten Ansatz des Systems Engineering wird daher im Model-based Systems Engineering (MBSE) das Engineering durch die formalisierte Anwendung der Modellierung unterstützt. Die INCOSE definiert MBSE als:

“formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the [concept stage] and continuing throughout development and later life cycle [stages].” [WSR+23, S. 219]

Das MBSE hat das Ziel, die interdisziplinären Entwicklungsprozesse und deren Ergebnisse mit einem ganzheitlichen Systemmodell zusammenzubringen und zu organisieren. Es gilt als Verständigungsmittel für die unterschiedlichen Entwicklungsbereiche und bietet die Grundlage zur Integration von speziellen Modellen der einzelnen Disziplinen [FMS12, S. 529]. In einer durchgeführten Literaturanalyse von HENDERSON und SALADO wurden die Vorteile des MBSE auf Basis der zu dem Themengebiet verfügbaren Literatur extrahiert. Die am häufigsten genannten Vorteile des MBSE in den zahlreichen ausge-

werteten Literaturquellen sind: bessere Kommunikation, verbesserte Systemqualität, bessere Rückverfolgbarkeit, reduzierte Kosten, Fehlerreduktion und bessere Wiederverwendung von Wissen [HS21, S. 60].

Die Basis sind Modelle. Modelle werden genutzt, um Sachverhalte abzubilden, die in ihrer Vollständigkeit zu komplex sind. Die allgemeine Modelltheorie nach STACHOWIAK beschreibt drei wesentliche Grundsätze, die ein Modell charakterisieren [Sta73, S. 131ff.]:

- **Abbildungsmerkmal:** Ein Modell bildet ein reales System oder ein Konzept mit dessen Eigenschaften ab.
- **Verkürzungsmerkmal:** Ein Modell erfasst nur die relevante Teilmenge, die zur Erfüllung des Modellierungszwecks notwendig ist.
- **Pragmatisches Merkmal:** Modelle ersetzen das Original, um eine Ersetzungsfunktion zu erfüllen. Diese sind abhängig vom Betrachter, dem Zeitpunkt und dem Zweck.

Um ein Systemmodell zu beschreiben, sind drei Aspekte nötig: eine Modellierungssprache, eine Methode und ein Software-Tool [Del14, S. 4]. Diese sind im Einklang zueinander zu betrachten (Bild 2-14). Eine **Methode** beschreibt, wie Aktivitäten durchzuführen sind, um ein Artefakt zu erstellen. Sie gibt vor, was spezifiziert werden muss und in welcher Reihenfolge Informationen entstehen. Eine **Modellierungssprache** ist ein standardisiertes Ausdrucksmittel, um bestimmte Zusammenhänge und Informationen zu beschreiben. Für die Erstellung des Modells wird ein **Tool** benötigt, welches die Modellierungssprache beherrscht und die einzelnen Aktivitäten der Methode unterstützt. [Del14, S. 4ff.], [WSR+23, S. 220].

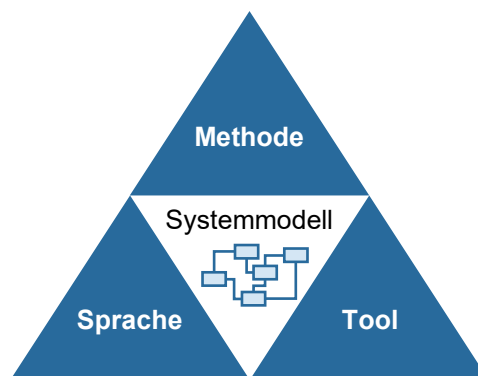


Bild 2-14: MBSE-Dreieck nach [Kai13, S. 26]

In den letzten Jahren wurden eine Vielzahl an Methoden entwickelt. Etablierte Methoden sind OOSEM, SYSMOD oder ISE&PPOOA [FMS12, S. 431ff.], [WLW+16], [FH19], [IFF+23, S. 52f.]. Sie orientieren sich häufig an dem Problemlösungszyklus nach HABERFELLNER et al., der in Kapitel 2.3.1 beschrieben wurde. Am Beispiel OOSEM sind fol-

gende Aktivitäten durchzuführen: (1) Stakeholderbedarfsanalyse, (2) Anforderungsanalyse, (3) Definition der logischen Architektur und (4) die Synthese der Kandidaten der physischen Architektur. Parallele Aktivitäten sind die Optimierung und Evaluation der Alternativen, die Unterstützung der Validierung und Verifikation sowie das Management der Rückverfolgbarkeit [FMS12, S. 436]. Die Methoden werden häufig unternehmensspezifisch angepasst und weiterentwickelt, um die jeweiligen Systeme unter den gegebenen Rahmenbedingungen gestalten zu können [DAR+21, S. 13]. Als Sprache hat sich die SysML als de-facto Standard im MBSE entwickelt [WSR+23, S. 13]. Diese muss in der Regel um unternehmensspezifische Aspekte erweitert werden, um die jeweiligen Aktivitäten zu unterstützen.

Modelle im MBSE können von sehr abstrakten bis hin zu detaillierten Darstellungen reichen. Sie können deterministisch oder stochastisch sein und sie können sich im Präzisionsgrad unterscheiden [MAS23, S. 5]. Je nach Anwendungsbereich und Zweck sind daher unterschiedliche Ausprägungen von MBSE möglich und geeignet. TSCHIRNER identifiziert insgesamt vier Konzepte, die die unterschiedlichen Ausprägungen von MBSE typisieren [Tsc16, S. 107f.].

Konzept 1 - Innovationswerkstatt: Das Konzept zeichnet sich durch seine Unterstützung bei der Ideenfindung und beim Planen und Klären von Aufgaben in Systementwicklungsprojekten aus. Es werden klassische Hilfsmittel wie Brown Paper und eine informale Sprache genutzt, um die Kreativität zu fördern. Das Vorgehen ist iterativ und Top-Down – beginnend mit der Idee. Es eignet sich insbesondere für kleine Vorentwicklungsprojekte und fördert die Zusammenarbeit durch ein informales "papierbasiertes" MBSE.

Konzept 2 - Mechatronic Sketch: Der Ansatz zeichnet sich durch einen klaren Fokus auf die mechatronische Struktur des Systems aus und verwendet eine semi-formale Sprache. Die Anzahl der Sprachkonstrukte ist überschaubar und ermöglicht die Verwendung von einfachen Tools. Der Ansatz eignet sich insbesondere für mittelgroße, kundenspezifische Systementwicklungsprojekte über verschiedene Wertschöpfungsstufen hinweg, erfordert jedoch einen hohen Pflegeaufwand für den Systemarchitekten.

Konzept 3 - Mechatronische Systemmodellierung: Dieses Konzept ist am weitesten mit der Idee von MBSE verknüpft und definiert ein ganzheitliches Systemmodell. Dieses wird Top-Down durch ein sequenzielles und formales Arbeiten definiert und umfasst die Systemstruktur, das Systemverhalten und Parameter. Es legt den Grundstein für die Integration von weiteren Datenmodellen durch die Nutzung von etablierten Tools mit einem großen Anwendungsbereich.

Konzept 4 – Simulationsbasierte Systemanalyse: Dieses MBSE-Konzept legt den Fokus auf das Verhalten eines Moduls und verwendet eine formale Sprache sowie Simulationswerkzeuge. Es wird in einem Bottom-Up Ansatz von Fachbereichsspezialisten erstellt und hauptsächlich in der Serienentwicklung großer Unternehmen eingesetzt. Es dient der Analyse und hat einen hohen Modellierungsaufwand.

Die Konzepte setzen unterschiedliche Schwerpunkte und unterscheiden sich in mehreren Dimensionen. U. a. nehmen der Formalisierungsgrad und der Aufwand mit jedem Konzept zu. Daher ist es wichtig, im Vorhinein den Zweck der Modellierung festzulegen. Eine Orientierung, welches Konzept für welches Ziel geeignet ist, gibt Tabelle 2-1.

Tabelle 2-1: MBSE-Konzepte und ihre prinzipielle Eignung für Modellierungszwecke nach Tschirner [Tsc16, S. 111]

	Modellierungszwecke																				
	Systemgrenzen definieren	Problemverständnis sicherstellen	Kommunikation befähigen	Kooperation unterstützen	Innovation befähigen	Interdisziplinäre Systembetrachtung	Architektur beschreiben	Lösungen kommunizieren	Produktkomplexität beherrschen	Dokumentation bereitstellen	Impactanalysen durchführen	Traceability ermöglichen	Übergang in die Konkretisierung	Verifikation und Validierung	Analysen unterstützen	Methoden unterstützen	Simulation unterstützen	Qualität verbessern	Risiko verringern	Entscheidungsunterstützung (Tech.)	Entscheidungsunterstützung (Techn. M.)
Konzept 1	●	●	●															●			
Konzept 2	●	●	●	●	●	●	●	●		●			●		●	●		●	●	●	
Konzept 3			●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Konzept 4						●								●	●		●	●	●	●	

● Prinzipiell gute Eignung eines Konzepts zur Unterstützung des Modellierungszwecks

Fazit: Systems Engineering bietet eine Herangehensweise zur Entwicklung komplexer Systeme. In Folge der durchgängigen Anwendung der Modellierung im Rahmen des MBSE werden Zusammenhänge über die Disziplinen hinweg deutlich und die wachsende Komplexität abgebildet. Je nach Ziel, gibt es unterschiedliche Herangehensweisen und Formalisierungsgrade von MBSE in Unternehmen. Dies erschwert den unternehmensübergreifenden Austausch im Kontext von SoS, da Modelle nicht in vergleichbaren Ausprägungen beschrieben sind. Für die Verbesserung der Kommunikation und des Verständnisses zwischen Unternehmen bietet sich ein semi-formales Konzept an.

2.3.4 SoS-Engineering

Fortschritte durch neue Technologien und die rasante Entwicklung der Informations- und Kommunikationstechnik verändern die Systeme (Kapitel 2.2). Gleichzeitig verändert sich auch das Engineering durch neue digitale Technologien und Konzepte (Bild 2-15). Mit Hilfe von Systems Engineering und insbesondere dem MBSE können komplexe Abhängigkeiten in einem System disziplinübergreifend abgebildet und die Entwicklung von komplexen technischen Systemen unterstützt werden [PJ20, S. 187]. Dennoch steht das

Engineering durch die stetig wachsende Vernetzung und der Integration in Systemverbünden vor neuen Herausforderungen, die nicht zuletzt in den charakteristischen Merkmalen von SoS begründet sind.

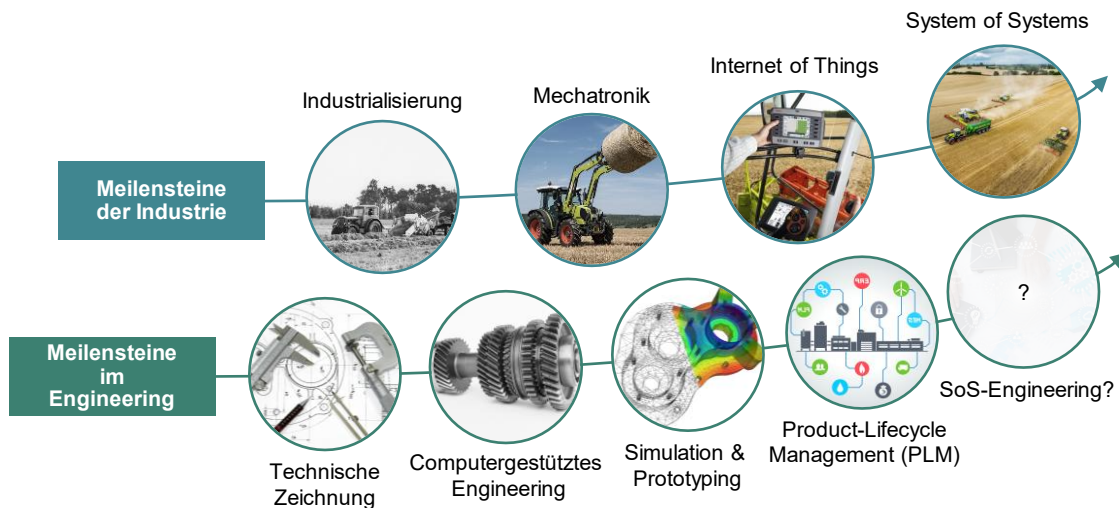


Bild 2-15: Der Wandel im Engineering

SoS-Engineering ist als Ergänzung des Systems Engineerings zu verstehen, welches den Fokus auf die Herausforderung von SoS sowohl technischer als auch managementzugehöriger Natur legt [NLF+15, S. 2]. Das **SoS-Engineering** umfasst die Planung, Analyse, Organisation, Entwicklung und Integration der Fähigkeiten von mehreren Systemen zu einer SoS-Fähigkeit [ISO21841, S. 2]. Eine Fähigkeit umschreibt in diesem Kontext die Eignung, ein Ziel zu erreichen [ISO21840, S. 1].

Ein SoS hat keine klaren Systemgrenzen und entsteht durch die dynamische Vernetzung eigenständiger Systeme. Dabei entwickelt es sich stetig weiter und Strukturen sowie Funktionen werden hinzugefügt, entfernt oder verändert. Die **unklare Systemgrenze und -hierarchie** sowie die Erkenntnis, dass ein SoS nie vollständig entwickelt ist, macht das Engineering von SoS zu einer großen Herausforderung [Jam08a, S. 6]. Während ein Kernkonzept des Systems Engineerings nach ISO 15288 die klare Definition von Systemgrenzen und die Möglichkeit zur Hierarchiebildung ist, braucht es im SoS-Engineering eine neue Denkweise und die Möglichkeit, ein SoS mit dynamischen Systemgrenzen zu beschreiben [ISO15288, S. 11].

ROGERS und MITCHELL sehen im MBSE einen Ansatz zur Bewältigung der SoS-Komplexität [RM21, S. 403]. MBSE wird allgemein als ein Ansatz angesehen, der Verbesserungen für die Systementwicklung bietet [CTE+23, S. 60]. Die Nutzung des MBSE-Ansatzes ist jedoch aktuell für die Analyse und Spezifikation eines abgegrenzten Systems bestimmt und adressiert nur unzureichend die besonderen Herausforderungen von SoS [MOD16, S. 637f.]. Auch der Ansatz des Model-based SoS-Engineerings ist noch jung und es bestehen offene Fragestellungen, wie die Definition einer einheitlichen Sprache zur Modellierung [NLF+15, S. 3ff.]. Gleichzeitig bestehen Herausforderungen bei der

Adaptierung des MBSE-Ansatzes in den Unternehmen und unterschiedliche Ausprägungen über die Unternehmen hinweg sind entstanden [CB18, S. 2]. Dies sorgt dafür, dass für eine **unternehmensübergreifende Kommunikation und Kooperation** auf Modellbasis zunächst Angleichungen nötig sind. Interoperabilität sowohl auf technischer Ebene als auch auf organisationaler Ebene ist eine Voraussetzung, um eine unternehmensübergreifende Gestaltung von SoS zu unterstützen [Tek22, S. 6].

Zusätzlich verfolgen Unternehmen unterschiedliche Entwicklungsansätze (vgl. Kapitel 2.3.2). Die vielschichtige Heterogenität des SoS aus unterschiedlichen Systemen die von unterschiedlichen Unternehmen entwickelt werden, führen zu **asynchronen Entwicklungs- und Lebenszyklusphasen** [Tek22, S. 6], [KSO+20, S. 370]. Dieser Zusammenhang hat zur Folge, dass sich das SoS zur Entwicklungszeit eines eigenständigen Systems verändert. Bild 2-16 visualisiert die unterschiedlichen Vorgehen zur Entwicklung eigenständiger Systeme und die damit einhergehenden Veränderungen im SoS in Abhängigkeit von der Zeit.

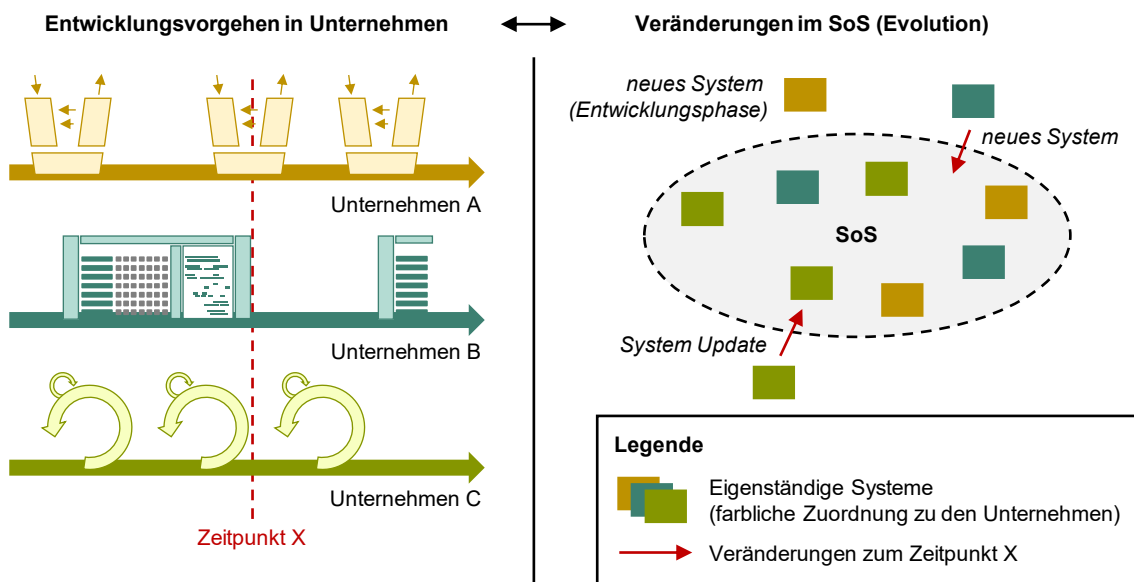


Bild 2-16: Auswirkungen unterschiedlicher Entwicklungsvorgehen auf das SoS

Aus der Heterogenität und der Notwendigkeit, die Interoperabilität zwischen Systemen zu gewährleisten, entstehen neue Aufwände für Unternehmen. Sie müssen über Unternehmensgrenzen hinweg zusammenarbeiten und innerhalb von Wertschöpfungsnetzwerken kooperieren. Während in einem Systementwicklungsprojekt die Verantwortung einem einzelnen Unternehmen zugeordnet ist und die Subsysteme eines Systems im Einklang zueinander die Meilensteine durchlaufen, herrschen im SoS häufig **verteilte Management- und Entwicklungsstrukturen** in dem kein Unternehmen die vollständige Verantwortung hat. Ist ein Traktor bspw. bereits auf dem Markt, während ein Mähdrescher noch in der Entwicklung steckt, ist die optimale Abstimmung dieser beiden Systeme aufeinander nicht mehr oder nur mit sehr großem Aufwand möglich. Gleichzeitig ist der

Entwicklungszyklus einer Service-App um ein Vielfaches kürzer als bspw. eine Mähdre-scher-Neuentwicklung. Im Rahmen der SoS-Gestaltung müssen daher kontinuierlich die Elemente und das Verhalten des SoS antizipiert werden und dynamische Anpassungen an die veränderten Bedarfe im SoS erfolgen [VRR07, S. 29].

Zusätzlich entstehen neue Aufwände durch die Veränderung zur Laufzeit der Systeme. Während ein klassischer Systemlebenszyklus linear von der Konzeptphase und Entwicklung über die Produktion und Nutzung bis zum Recycling verfolgt wurde, müssen Unternehmen zukünftig die Funktionsfähigkeit und -sicherheit in der Nutzungsphase gewährleisten [WSR+23, S. 26]. Dies ist u. a. auf die Nachhaltigkeitsmaßnahmen und neue Regulatorik zurückzuführen, wie den Cyber-Resilience Act der Europäischen Kommission [Eur22-ol]. Damit einher geht der Bedarf auf Veränderungen im Systemumfeld einzugehen, die sich auch in der **Evolution** von SoS widerspiegeln. Sowohl Upgrades der Software als auch der Hardware werden notwendig, um die Funktionsfähigkeit und -sicherheit im Zusammenspiel mit weiteren Systemen zu gewährleisten. Dies führt zu einem veränderten Verständnis über den Lebenszyklus von einzelnen Systemen, die im SoS partizipieren (Bild 2-17).

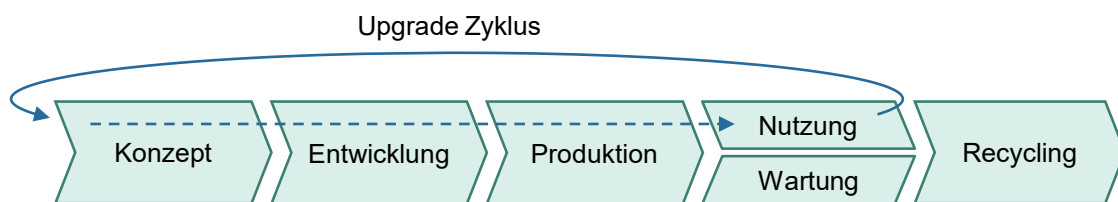


Bild 2-17: Lebenszyklus von Systemen im SoS in Anlehnung an WALDEN et al. [WSR+23, S. 26]

Fazit: Die SoS-Gestaltung folgt keiner normalen Systementwicklung [ADK13, S. 11]. Derzeitige Ansätze und Standards, wie die ISO 15288, fokussieren die Entwicklung von klar abgegrenzten Systemen und sind nicht für die Entwicklung von SoS ausgelegt. Das Ziel der SoS-Gestaltung muss in dem Verständnis der Systeme und deren Interaktion liegen und dabei die Weiterentwicklung des SoS fördern [DRL+10, S. 13]. Nachträgliche Updatefähigkeit oder die frühzeitige Standardisierung von Schnittstellen im SoS-Umfeld bieten hier erste Ansätze, um den Herausforderungen zu begegnen. Weitere Lösungen sind notwendig, um die Kommunikation und Kooperation von Unternehmen im Rahmen der SoS-Gestaltung zu fördern. Zusammenfassend ergeben sich folgende zentrale Herausforderungen im SoS-Engineering:

- unklare Grenzen des SoS,
- Notwendigkeit zur unternehmensübergreifenden Kommunikation und Kooperation,
- asynchrone Entwicklungs- und Lebenszyklusphasen und
- stetige Weiterentwicklung (Evolution).

2.4 Lösungswissen im Gestaltungsprozess

Bei der Gestaltung von SoS stehen Organisationen und Entwicklerinnen und Entwickler vor zahlreichen Herausforderungen. Diese umfassen u. a. die Bewältigung einer komplexen Struktur mit dynamischen Elementen sowie die Zusammenarbeit mit einer Vielzahl von Akteuren aus verschiedenen Unternehmen und Disziplinen (vgl. Kapitel 2.2 und 2.3). Gleichzeitig müssen sie immer schneller auf Veränderungen im SoS reagieren, bedingt durch neue Regularien und verkürzte Produktlebenszyklen. Ein effektives Wissensmanagement kann helfen, diesen Herausforderungen zu begegnen. Es ermöglicht den Wissensaustausch über Individuen hinweg und fördert die effiziente Wiederverwendung von Lösungswissen. In den Ingenieurwissenschaften wird ein solches Wissensmanagement als wesentlicher Erfolgsfaktor angesehen, um mit der zunehmenden Komplexität und den aktuellen Herausforderungen umzugehen [VDI5610, S. 2].

Gegenüber dem klassischen Wissensmanagement innerhalb von Organisationen muss im Kontext von SoS das Wissensmanagement organisationsübergreifend betrachtet werden. Dabei werden Barrieren, wie das Schützen von individuellem Know-how, noch relevanter. Das Wissen eines Unternehmens ist Teil des immateriellen Vermögens und somit schützenswert [Nor11, S. 54]. Im Rahmen von SoS muss Wissen daher in einer geeigneten Form zugänglich gemacht werden.

2.4.1 Wissensmanagement

Der Begriff Wissen wird im Alltag häufig verwendet und viele Assoziationen zu dem Begriff bestehen. Ein wesentlicher Konsens besteht darin, dass Wissen zur Lösung von Aufgaben benötigt wird. Auch in der Wissenschaft ist Wissen ein wichtiges Thema und wird in vielen Veröffentlichungen diskutiert. Diese Arbeit stützt sich auf das Verständnis nach PROBST et al., die den Begriff wie folgt definieren:

„Wissen bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen.“ [PRR10, S. 23]

Im Rahmen von Aufgaben, Problemen und Entscheidungsprozessen wird Wissen angewendet, um zu einer Lösung zu gelangen [VDI5610, S. 9]. Grundlage hierfür ist eine geeignete Wissensbasis, die sich in unterschiedlichen Formen in Organisationen wiederfindet. NONAKA und TAKEUCHI unterscheiden implizites und explizites Wissen. Die Unterscheidung stützt sich auf Überlegungen zu implizitem Wissen nach POLANYI und wird wie folgt charakterisiert [NT12, S. 23], [Pol66]:

- **Implizites Wissen** ist individuell gebundenes Wissen und repräsentiert das persönliche Wissen eines Individuums. Dieses Wissen ist nicht formalisiert und

schwierig weiterzugeben. Ein Beispiel ist Erfahrungswissen, das meist von Personen gesammelt wird. Es ist schwierig zu kommunizieren und von subjektiven Wahrnehmungen geprägt.

- **Explizites Wissen** ist demgegenüber formales Wissen, das in klar definierter Form ausgedrückt und weitergegeben werden kann. Es kann in unterschiedlichen Formen, wie Daten, wissenschaftlichen Formeln oder universellen Prinzipien festgehalten werden. Im Gegensatz zu implizitem Wissen ist explizites Wissen objektiv und leichter verständlich, da es in konkreter Form festgehalten ist und so weitergegeben werden kann [NT12, S. 23].

Ein wesentliches Problem des Wissensmanagement ist die Überführung von implizitem in explizites Wissen [Nor11, S. 47]. Dieser Übergang wird auch als **Externalisierung** bezeichnet und kann durch Methoden, wie Metaphern, Analogien und Modellen unterstützt werden [PRR10, S. 124].

Neben dem Explizieren von Wissen sind im Rahmen des Wissensmanagement weitere Aspekte relevant. Nach PROBST et al. muss ein geeignetes Wissensmanagement insgesamt sechs Kernprozesse vereinen, die enge Verbindungen zueinander aufweisen und sich gegenseitig beeinflussen (vgl. Bild 2-18) [PRR10, S. 28].

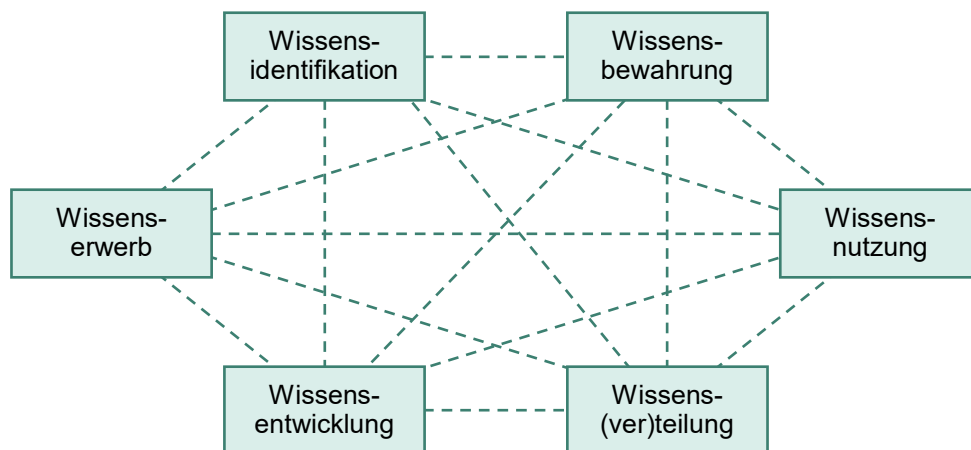


Bild 2-18: Kernprozesse des Wissensmanagements nach PROBST et al. [PRR10, S. 28]

Die wesentlichen Aspekte der Kernprozesse sind im Folgenden kurz zusammengefasst.

- **Wissensidentifikation:** Wissensmanagement soll die interne als auch externe Transparenz gewährleisten und die Individuen bei ihrer Suche nach Wissen unterstützen. Externe Wissensidentifikationsmaßnahmen helfen Unternehmen dabei, ihr Wissensumfeld zu analysieren und zu beschreiben.
- **Wissenserwerb:** Das Potenzial des Wissenserwerbs wird oft nicht vollständig genutzt. Insbesondere der Wissenserwerb aus externen Quellen, wie Kunden, Lieferanten, Konkurrenten oder Kooperationspartner, wird sehr oft nicht ausgeschöpft. Bei der Umsetzung von Wissensmanagement sollten Möglichkeiten zur Nutzung dieses Potenzials berücksichtigt werden.

- **Wissensentwicklung:** Neben dem Wissenserwerb ist die Wissensentwicklung ein zweiter Baustein zum Aufbau von neuem Wissen, bspw. durch Forschung und Entwicklung. Es umfasst die Schaffung neuer Fähigkeiten, Produkte, Ideen und Prozesse.
- **Wissensverteilung:** Die Verteilung von Wissen ist wichtig, um Informationen für die gesamte Organisation nutzbar zu machen. Es ist nicht notwendig, dass jeder alles weiß. Das Wissen sollte sinnvoll verteilt werden und genau da verfügbar sein, wo es zur Erfüllung von Aufgaben benötigt wird.
- **Wissensnutzung:** Der produktive Einsatz von Wissen ist der Kern des Wissensmanagements. Hierzu müssen Wissensbestände nutzbar sein und Barrieren für den Einsatz verhindert werden.
- **Wissensbewahrung:** Der Prozess der Wissensbewahrung sorgt dafür, dass wertvolle Erfahrungen und Fähigkeiten auch für zukünftige Aufgaben zur Verfügung stehen. Dafür ist die effiziente Nutzung verschiedenster Medien zur Speicherung des Wissens notwendig [PRR10, S. 29ff.].

Fazit: Ein geeignetes Wissensmanagement ist notwendig, um Aufgaben erledigen zu können. Im Rahmen von SoS muss das Wissensmanagement auf einer unternehmensübergreifenden Ebene funktionieren, um den Herausforderungen zu begegnen. Es werden Medien benötigt, die den Austausch und die Weiterentwicklung von Wissen über Unternehmensgrenzen hinweg ermöglichen. Der Austausch von Informationen durch Interaktion und Kommunikation über die Unternehmensgrenzen ist dabei ein Schlüssel zur Wissensentstehung. Durch die Verknüpfung von verteilten Informationen wird Wissen erst möglich. Sollten Kommunikationsbarrieren zwischen beteiligten Akteuren bestehen, wird die Lösungsfindung verhindert [PRR10, S. 125].

2.4.2 Wissen in Form von Lösungsmustern

Wissen ist notwendig, um Aufgaben und Problemstellungen zu lösen. Die Nutzung von vorhandenem Wissen ist ein entscheidender Faktor für ein effizientes Engineering. Ein in der Literatur häufig beschriebener Ansatz zur Operationalisierung von Wissensmanagement ist der Einsatz von Mustern¹. Im Folgenden wird die Frage geklärt, ob Muster ein geeigneter Ansatz für das Wissensmanagement von SoS sein kann.

Eine etablierte Definition von Mustern im Zusammenhang mit der Produktentwicklung ist die Definition nach ALEXANDER et al.:

„Jedes Muster beschreibt zunächst ein in unserer Umwelt immer wieder auftretendes Problem, beschreibt dann den Kern der Lösung dieses

¹ Im Englischen ist der Begriff „pattern“ weit verbreitet, der im Deutschen mit „Muster“ übersetzt wird. Gleichzeitig konkretisieren viele Wortergänzungen (solution pattern, design pattern, ...) den Kontext, in denen Muster genutzt werden. Im Folgenden wird der Begriff Muster synonym zum Begriff Lösungsmuster verwendet. Diese stehen im Fokus dieser Arbeit.

Problems, und zwar so, dass man diese Lösung millionenfach anwenden kann, ohne sich je zu wiederholen. “[AIS+95, S. 10]

In einer weiteren etwas vereinfachteren Definition beschreibt BARTER ein Muster als „*Lösung für ein bestimmtes Problem in einem bestimmten Kontext*“ [Bar98, 350]. Beide Definitionen stellen den Kern von Lösungsmustern heraus. Demnach ist ein Lösungsmuster eine verallgemeinerte Problem-Lösungsbeschreibung. Diese Problem-Lösungspaare können in allen Aktivitäten innerhalb eines Kontextes beschrieben werden und finden sich seit Jahren in unterschiedlichen Disziplinen wieder.

Aufbauend auf den Grundlagen nach ALEXANDER et al. wurde die Nutzung von Lösungsmustern zunächst stark in der Softwaretechnik vorangetrieben [AIS+77]. Dies wird zurückgeführt auf die Arbeiten von GAMMA et al., der sogenannten „Gang of Four“. In ihrer Arbeit stellen sie einen initialen Satz von 23 Software-Entwurfsmustern vor, die von strukturellen Mustern, wie einen *Adapter*, bis zu Verhaltensmustern, wie die *Kette der Verantwortlichkeiten*, reichen [GHJ+93, S. 409]. Die ersten disziplinübergreifenden Muster wurden von GASKA und GAUSE beschrieben, die die Idee der Anforderungsmuster aus der Software Community erweiterten und 19 disziplinübergreifende Anforderungsprozessmuster beschreiben, die als Ausgangsbasis für Anforderungsmuster dienen [GG98, S. 187]. Im weiteren Verlauf wurde der Musteransatz durch die kontinuierliche Verwendung von Mustern und die in der Community gesammelten Erfahrungen weiterentwickelt. Es wurde erkannt, dass Muster in verschiedenen Bereichen angewendet werden können. Ein Beispiel sind die Organisationsmuster von KOLP et al., die generische Strukturen von Unternehmensorganisationen beschreiben [KGM03]. Auch im Rahmen der Mechatronik und dem Systementwurf intelligenter technischer Systeme wurde das Potential erkannt und Muster entwickelt [Pet03], [GDZ+07], [SZ03].

Der Schluss liegt nahe, dass die Nutzung von Lösungsmustern auch im Kontext von SoS sinnvoll erscheint. Um dies genauer zu untersuchen, wurde im Rahmen einer systematischen Literaturanalyse die Eignung des Lösungsmustereinsatzes im Kontext von SoS analysiert [AGW+22]. Hierzu wurden zunächst Literaturquellen dahingehend untersucht, welche Vorteile Lösungsmuster im Engineering bieten. Auf Basis der Arbeiten von RISING, SANZ und ZALEWSKI sowie CLOUTIER und VERMA konnten folgende Vorteile identifiziert werden [Ris99], [Ris07], [SZ03], [CV07]:

- **Übertragbarkeit:** Die einheitliche Struktur von Mustern erleichtert ihre Anwendung in sämtlichen Gestaltungsbereichen der Produktentstehung. Dies bietet die Möglichkeit, Wissen sprachunabhängig wiederzuverwenden und zu repräsentieren.
- **Verbesserung der Kommunikation:** Innerhalb des Wissensmanagements dienen Muster als Grundlage für den Austausch und die Weiterentwicklung von Wissen. Durch den Einsatz von Mustern wird die Kommunikation verbessert sowohl für individuelle Entwickler als auch innerhalb und zwischen Teams.

- **Langfristige Dokumentation:** Wie bereits von NONAKA und TAKEUCHI festgestellt, ist das Wissen in der Produktgestaltung oft nur implizit vorhanden. Im Rahmen der Externalisierung können Muster als Medium zum Explizieren von Wissen genutzt werden und damit Dritten zugänglich gemacht werden. Ein Muster kann sich weiterentwickeln und durch weitere Informationen ergänzt werden.
- **Komplexitätsreduktion:** Muster helfen dabei, komplexe Probleme in elementare Probleme zu zerlegen und so die Komplexität auf lösbare Teilprobleme zu reduzieren. Durch die Kombination von Lösungsmustern können komplexe Zusammenhänge gelöst werden. Dabei stehen Muster in Beziehung zueinander und können sich bedingen oder gegenseitig ausschließen.
- **Effizienzsteigerung:** Durch den Einsatz von Mustern kann die Effizienz in Problemlösungsprozessen erheblich gesteigert werden. Sie ermöglichen es, kollektives Erfahrungswissen in neue Entwicklungsprojekte einzubringen. Im Sinne von Best-Practice- und Lessons-Learned-Ansätzen tragen Muster dazu bei, Erfolge hervorzuheben und Misserfolge in zukünftigen Projekten zu vermeiden.
- **Förderung der Kreativität:** Im Vergleich zu konkreten Lösungen werden in Mustern abstrahierte Problemlösungsinformationen festgehalten. Diese ermöglichen es durch Muster die gleiche Art von Problemen auf ähnliche Art und Weise zu lösen, ohne dass sich die konkrete Lösung wiederholt. Dabei können sich Muster im Abstrahierungsgrad unterscheiden. Je abstrakter das Muster beschrieben ist, desto größer ist auch der mögliche Anwendungsbereich. Muster dienen als Anregung, um Lösungen für Problemstellungen zu erarbeiten. Eine große Wissensbasis aus Mustern kann so die Kreativität eines Individuums unterstützen.

Für die Gestaltung von SoS sind die genannten Vorteile von Lösungsmustern in vielfacher Hinsicht relevant. Durch die Managementunabhängigkeit ist der Austausch von Wissen und die Kommunikation über Unternehmensgrenzen hinweg eine wesentliche Herausforderung (vgl. Kapitel 2.3.4). Durch die klare Struktur und die Abstraktion von konkreten Lösungen, sind Muster ein Medium, die den Wissensaustausch in diesem herausfordernden Umfeld ermöglichen können. Auch die Komplexität von SoS, die sich durch die dynamische Interaktion ergibt, kann durch Muster auf wesentliche Teilprobleme reduziert werden. So können Muster in Teilen zur Lösung der Probleme beitragen.

Erste Arbeiten untersuchen bereits den Einsatz von Mustern im Rahmen von vernetzten Systemen (Bild 2-19). Um die Dynamik in SoS zu analysieren, beschreiben KALAWSKY et al. Architekturmuster als Hilfsmittel zum Verständnis von Kommunikation und Verhaltensweisen in komplexen SoS [KJT+13, S. 289f.]. Diese helfen dabei, alternative Architekturen zu verstehen und die Effizienz der Kommunikation in einem dynamischen SoS zu analysieren. Ein weiteres Beispiel sind die Interoperabilitätsmuster nach BRÖRING et al., die generalisierte Interaktionsstrukturen in einem IoT-Ökosystem beschreiben

[BSS+17, S. 56f.]. Sie helfen dabei, die grundlegenden Zusammenhänge zwischen Akteuren zu verstehen und deren Interoperabilität zu fördern. Dabei gehen sie konkret auf das Zusammenspiel zwischen Plattformen und Applikationen bzw. Services ein.

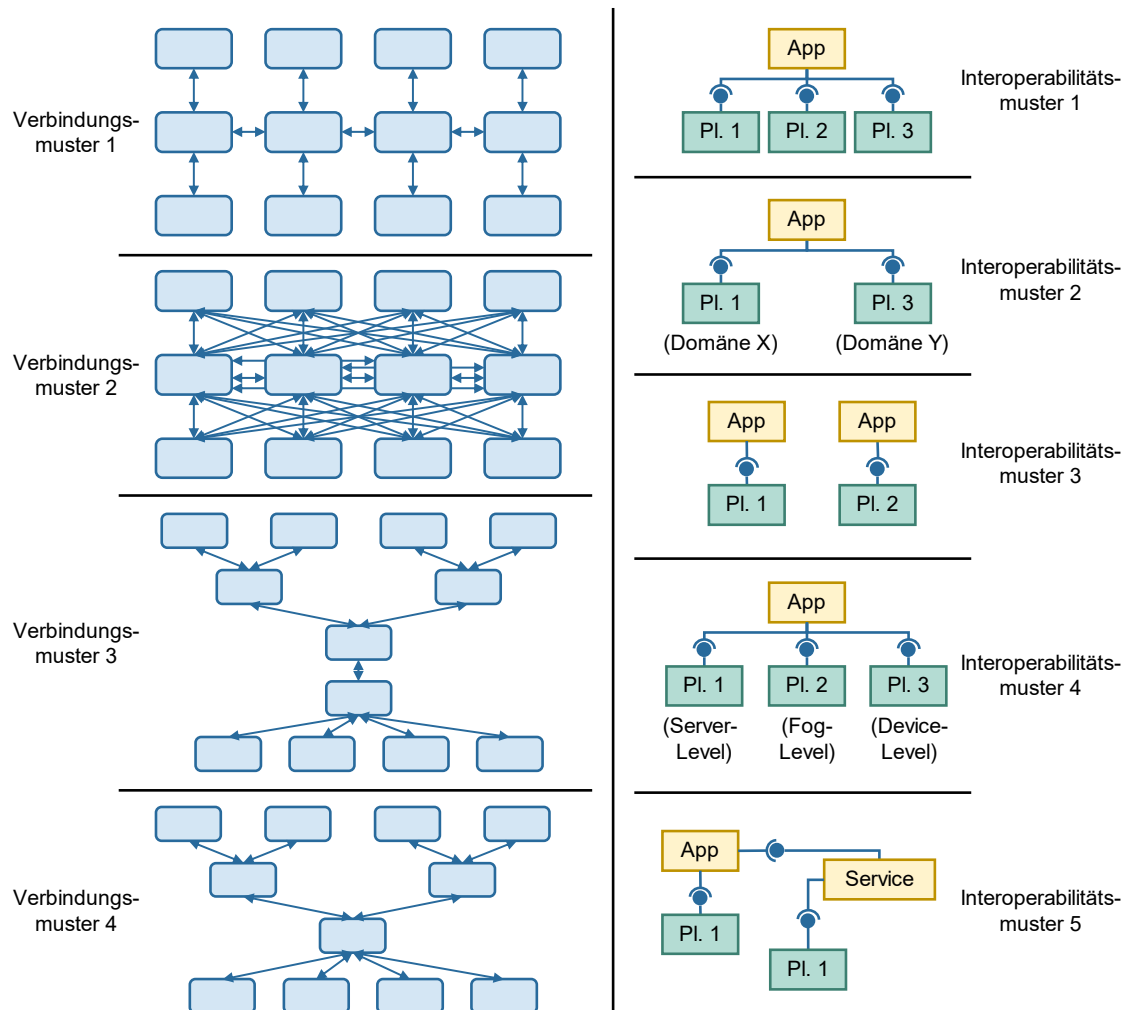


Bild 2-19: Gegenüberstellung der Verbindungsmuster nach KALAWSKY et al. (links) und Interoperabilitätsmuster nach BRÖRING et al. (rechts) [KJT+13, S. 290], [BSS+17, S. 57]

Fazit: Während der Musteransatz für die Gestaltung von SoS grundsätzlich sinnvoll erscheint, bestehen nach wie vor Hemmnisse bei der Übertragung. Die vorgestellten Muster nach KALAWSKY et al. und BRÖRING et al. sind unabhängig voneinander beschrieben und folgen beispielsweise keiner einheitlichen Strukturierung. Obwohl die Grundidee der Lösungsmuster ähnlich ist und beide auf das Zusammenspiel zwischen Systemen eingehen, unterscheiden sie sich in der Beschreibungsstruktur. Zusätzlich sind sie in unterschiedlichen Abstraktionsgraden beschrieben und somit nicht direkt vergleichbar. Dies ist jedoch erforderlich, um als Wissensbasis für die Gestaltung von SoS nutzbar zu sein und Muster auf unterschiedliche Problemstellungen anwenden zu können.

2.4.3 Wissen in Form von Designprinzipien

Neben theoretischen Grundkenntnissen und expliziten Wissen in Form von Lösungsmustern sind auch Handlungsweisen eine Form von Wissen, die zur Lösung von Problemen benötigt werden [PRR10, S. 23]. Diese grundsätzlichen Handlungsweisen werden häufig in Form von Designprinzipien beschrieben. Designprinzipien sind in vielen Ingenieursdisziplinen seit vielen Jahren etabliert, um einfache oder auch komplexe Aufgaben zu unterstützen. Beispiele sind die 15 Prinzipien des Software Engineerings nach DAVIS oder auch die Gestaltungsprinzipien für den Maschinenbau nach PAHL und BEITZ [Dav94], [PB21]. Designprinzipien legen bestimmte Handlungsweisen und Richtlinien dem Handeln zugrunde und sollen Hilfestellung bei der Gestaltung liefern. Ein Beispiel ist das Prinzip der gleichen Gestaltfestigkeit aus dem Maschinenbau. Geht es im Rahmen der Konstruktion um die Fragestellung, wie die Kräfte im Bauteil verteilt werden sollen, kann das Prinzip der gleichen Gestaltfestigkeit herangezogen werden [PB21, S. 527f.]. Dies dient dem Konstrukteur als Orientierung für die Gestaltung eines Bauteils, welches die Kräfte gleichmäßig aufnimmt und weiterleitet, ohne an bestimmten Stellen zu versagen.

Allgemein ist ein Prinzip nach ROUSSEAU *ein grundlegender Gedanke oder eine Regel, die als Richtschnur für ein Urteil oder eine Handlung dienen kann* [Rou18, S. 3]. Im Kontext des Engineerings sind Prinzipien Regeln, Richtlinien, Konzepte oder Methoden für die Gestaltung von Produkten, Systemen oder Prozessen. Sie werden aus Erfahrung und Wissen abgeleitet und bestätigt. Prinzipien sind generisch und bilden eine theoretische Grundlage für das Design [Bal11, S. 29]. Dies spiegelt auch die Definition nach FU et al. wider. Demnach ist ein Prinzip:

„A fundamental rule or law, derived inductively from extensive experience and/or empirical evidence, which provides design process guidance to increase the chance of reaching a successful solution.”
[FYW15, S. 2]

Prinzipien¹ sind eine Form expliziten Wissens [FYW15, S. 1]. Sie sind insbesondere für Ingenieure ein geeignetes Mittel, um die Chance auf eine erfolgreiche Lösung zu erhöhen. Dies gilt auch für die Gestaltung von SoS. MEENTEMEYER et al. beschreiben in ihrer Arbeit bspw. neun Designprinzipien für das SoS-Engineering [MSB09, S. 341]. Diese umfassen sowohl allgemeine organisatorische Handlungsweisen als auch konkrete Maßnahmen für die Gestaltung. Beispiele für allgemeine Handlungsweisen sind das Prinzip der integrierten Produktteams oder das Prinzip der Modellierung und Simulation. Beim Prin-

¹ Prinzipien kommen in der Literatur in unterschiedlichen Granularitäten und Spezifitäten vor. FU et al. unterscheiden daher zwischen Prinzipien, Leitlinien und Heuristiken [FYW15]. Heuristiken basieren demnach eher auf Intuition und experimentellem Verständnis während Prinzipien auf umfassender Erfahrung und empirischer Evidenz basieren. Da das Engineering von komplexen Systemen noch ein junges Wissensgebiet ist, ist die Spezifität der Ansätze noch gering [BOM+23, S. 1]. Dennoch wird in der Literatur im Kontext von SoS häufig von Prinzipien gesprochen. Im Rahmen dieser Arbeit wird daher keine Unterscheidung zwischen den unterschiedlichen Begriffen gemacht und der Begriff Prinzip für Handlungsweisen mit unterschiedlichen Spezifitäten genutzt.

zip der integrierten Produktteams sollten sowohl Verantwortliche der eigenständigen Systeme als auch Verantwortliche für das gesamte SoS in einem Team zusammenarbeiten. Diese sollten miteinander kommunizieren und gemeinsam Entscheidungen treffen. Das Prinzip der Modellierung und Simulation beschreibt den Einsatz von Modellen und Simulationen als relevante Ergänzung für den Entwicklungs- und Bewertungsprozess, der u. a. Vorteile bei der Validierung bietet. Ein Beispiel für eine konkrete Maßnahme, die als Handlungsweise den Erfolg von SoS unterstützen kann, ist das Prinzip der Interoperabilität. Dieses empfiehlt die Etablierung von Kommunikationsprotokollen und Standards, um die Interoperabilität innerhalb eines SoS zu gewährleisten [MSB09, S. 341f.].

Fazit: Insgesamt stellt das Wissensmanagement einen wesentlichen Erfolgsfaktor in der Entwicklung von komplexen Systemen dar. Die effiziente Nutzung von Wissen in Form von Lösungsmustern und Designprinzipien kann sich als vorteilhaft in der organisationsübergreifenden Gestaltung von SoS erweisen. Hierfür müssen sie jedoch in geeigneter Struktur vorliegen und zielgerichtet für die passenden Problemstellungen eingesetzt werden.

2.5 Problemabgrenzung

Die Problemanalyse verdeutlicht den Wandel, den die Systeme aufgrund der Digitalisierung und zunehmenden Vernetzung von mechatronischen Systemen hin zu Intelligenten Technischen Systemen (ITS) durchlaufen. ITS sind unter bestimmten Voraussetzungen dazu in der Lage, in Systemverbünden als sog. **System of Systems** zusammenzuwirken, um ein übergeordnetes Ziel zu erreichen. Nur durch ihre Interaktion können sie Aufgaben erfüllen, die als einzelnes System nicht zu erfüllen sind. Die interagierenden Systeme sind eigenständig und werden von unterschiedlichen Unternehmen gemanagt, was Herausforderungen in Bezug auf die Interoperabilität und die Abstimmung über Verhaltensweisen mit sich bringt. Durch die Interaktion in Systemverbünden kommt es regelmäßig zu Verhalten, welches nicht vollständig im Vorhinein vorhergesagt werden kann. Dies unterscheidet SoS von eigenständigen Systemen, bei denen die Effekte innerhalb klarer Grenzen analysiert werden können. Insgesamt gibt es mehrere Merkmale, die SoS von eigenständigen Systemen unterscheiden. Verschiedene Ausprägungen dieser Merkmale sorgen für eine große Vielfalt an SoS und in vielerlei Hinsicht heterogene Strukturen (vgl. Kapitel 2.2).

Die Interaktion von Systemen innerhalb von Systemverbünden stellt das Engineering vor verschiedene Herausforderungen. Diese umfassen einerseits den zusätzlichen Komplexitätsfaktor aufgrund der Vielzahl an möglichen Interaktionen innerhalb eines SoS sowie andererseits die Notwendigkeit der Abstimmung und Zusammenarbeit über Unternehmensgrenzen hinweg. Für Unternehmen verändern sich die Geschäftstätigkeiten und die Unternehmensgrenzen verschieben sich. Daher ist es von großer Bedeutung, geeignete Gestaltungsansätze und -methoden für das **SoS-Engineering** zu entwickeln, um diesen

Herausforderungen erfolgreich zu begegnen und die Interaktion innerhalb von Systemverbünden effektiv zu gestalten (vgl. Kapitel 2.3).

Die gleichzeitig wachsenden Marktanforderungen und immer kürzer werdende Entwicklungszyklen, machen den effizienten und zielgerichteten Umgang mit Wissen relevanter. Ein geeignetes Management von Wissen spielt für jedes Unternehmen eine entscheidende Rolle als Basis für interdisziplinäre Zusammenarbeit und Innovationen. Ein vielversprechender Ansatz für die Operationalisierung von **Wissensmanagement** sind Lösungsmuster. Ein **Lösungsmuster** beschreibt ein wiederkehrendes Problem sowie die grundlegende Lösung für dieses Problem. Es kann beliebig oft eingesetzt werden, ohne dass sich die Lösung wiederholt. Eine Vielzahl von dokumentierten Mustern gibt Entwicklern Impulse, ihre Denkmuster mit dem Wissen des Kollektivs zu reflektieren und zu erweitern. Gleichzeitig bieten Lösungsmuster eine Kommunikationsbasis, da sie Zusammenhänge in abstrakter Weise explizieren und so für ein einheitliches Vokabular sorgen. Als Ergänzung können durch **Designprinzipien** weitere sinnvolle Handlungsweisen als Wissen bereitgestellt werden, die die unternehmensübergreifende Gestaltung von SoS unterstützen (vgl. Kapitel 2.4).

Insgesamt lässt sich feststellen, dass der Erfolg der heutigen Systeme maßgeblich von ihrer Integration und Interaktion innerhalb von Systemverbünden abhängt. Die komplexe Gestaltung dieser Systeme ist verbunden mit zahlreichen Herausforderungen in einem heterogenen Umfeld. Dies zeigt den Bedarf an einem „*Rahmenwerk zur musterbasierten Gestaltung von System of Systems*“. Das Rahmenwerk soll ein auf die Herausforderungen angepasstes SoS-Engineering in Kombination mit einem geeigneten Wissensmanagement bereitstellen, um die unternehmensübergreifende Gestaltung zu unterstützen und SoS zielorientiert zu gestalten. Der Bedarf an dem Rahmenwerk gliedert sich in die Handlungsfelder, die in Bild 2-20 dargestellt sind und im Folgenden näher beschrieben werden.

Handlungsfeld 1: Definition und Strukturierung von SoS

Das Forschungsfeld von SoS ist sehr heterogen und eine einheitliche Definition existiert nicht [NLF+15, S. 4]. Die vielfältigen Ausprägungen von SoS und die gleichzeitig unklare Grenze zwischen SoS und einem System sorgen dafür, dass die wesentlichen Fragestellungen und Herausforderungen nicht fokussiert angegangen werden können. Insbesondere durch die notwendige Kommunikation und Kooperation über Unternehmensgrenzen hinweg, ist ein einheitliches Verständnis für SoS essenziell. Es bedarf einer einheitlichen Definition und Strukturierung von SoS unter Berücksichtigung der charakteristischen Merkmale.

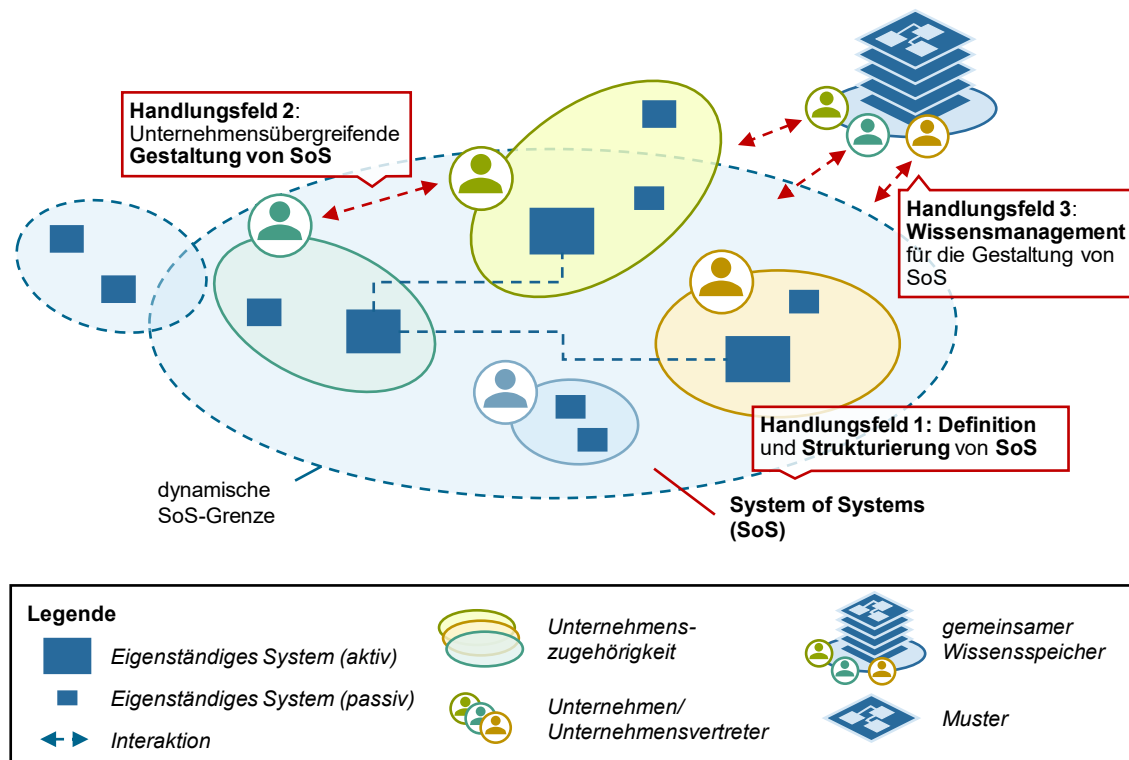


Bild 2-20: Übersicht der Handlungsfelder im Rahmen der Arbeit

Handlungsfeld 2: Unternehmensübergreifende Gestaltung von SoS

Die Herausforderung für Unternehmen liegt darin, über etablierte Grenzen hinwegzudenken und aktiv an der Gestaltung von SoS mitzuwirken. Dies beinhaltet sowohl zahlreiche technische Herausforderungen als auch die Frage, wie unternehmensübergreifend optimale Lösungen erzielt werden können. Um übergeordnete Ziele zu erreichen, die durch einzelne Systeme nicht gelöst werden können, müssen neue Gestaltungsansätze entwickelt werden. Diese Ansätze sollten die dynamischen Systemgrenzen und die stetige Weiterentwicklung von SoS berücksichtigen. Aufgrund des hohen Grades an Interdisziplinarität und Komplexität ist es wichtig, Systems Engineering und MBSE-Ansätze einzubeziehen. Die neuen Gestaltungsansätze müssen die unterschiedlichen Formen der Unternehmenszusammenarbeit berücksichtigen sowie auf die verschiedenen Arten von SoS eingehen. Wesentlich für die Anwendbarkeit in Unternehmen sind konkrete Beschreibungen von Tätigkeiten und Handlungsempfehlungen, um die unterschiedlichen Herausforderungen der SoS-Gestaltung zielorientiert angehen zu können.

Handlungsfeld 3: Wissensmanagement für die Gestaltung von SoS

Eine unternehmensübergreifende Gestaltung von SoS erfordert viel Wissen über technische Systeme und deren Verhaltensweisen. Unterschiedliche Ziele, IP-Schutz oder auch Sprachbarrieren sorgen dafür, dass die Zusammenarbeit und der Austausch von Wissen zwischen Unternehmen häufig schwierig sind. Es besteht Bedarf an einer einheitlichen Wissensbasis, die die Kommunikation und Zusammenarbeit zwischen Unternehmen fördert. Dazu gehören geeignete Medien, wie Lösungsmuster und Designprinzipien. Diese

erfordern eine geeignete Strukturierung und einheitliche Beschreibung, um sie zielgerichtet im Rahmen der Wissensnutzung einzusetzen.

2.6 Anforderungen an das Rahmenwerk

Aus der Problemanalyse und den drei identifizierten Handlungsfeldern resultieren insgesamt acht Anforderungen an ein *Rahmenwerk zur musterbasierten Gestaltung von System of Systems*. Diese werden im Folgenden beschrieben.

Handlungsfeld 1: Definition und Strukturierung von SoS

A1) Definition von SoS im Kontext von technischen Systemen: SoS werden unterschiedlich definiert, sodass es kein einheitliches Verständnis gibt. Die unterschiedlichen Ausprägungen von SoS erfordern eine Fokussierung auf einen Teilbereich. Um den Bedarf von Unternehmen zu entsprechen, muss das Verständnis von SoS auf technische SoS begrenzt und klar definiert werden.

A2) Berücksichtigung der charakteristischen Merkmale von SoS: SoS sind Systemverbünde, die anhand von charakteristischen Merkmalen beschrieben werden. Eine vollumfängliche Betrachtung der charakteristischen Merkmale ist relevant, um die daraus resultierenden Herausforderungen im Rahmen der Gestaltung zu adressieren.

A3) Klassifizierung von SoS: Die Vielzahl an unterschiedlichen Ausprägungen von SoS und Ausgangssituationen verhindern die Vereinheitlichung von Gestaltungsansätzen. Um bedarfsgerechte Methoden und Hilfsmittel bereitzustellen, bedarf es einer Klassifizierung von SoS. Die Klassen müssen sich in relevanten Kriterien für die Gestaltung unterscheiden und dabei prägnant und sparsam definiert werden.

Handlungsfeld 2: Unternehmensübergreifende Gestaltung von SoS

A4) Berücksichtigung von SoS- und Systemperspektiven: Systeme werden von einzelnen Unternehmen entwickelt und betrieben. Im SoS werden diese integriert und müssen zusammen mit anderen Systemen von unterschiedlichen Unternehmen zusammenwirken, um übergeordnete Aufgaben zu erledigen. Ein einzelnes Unternehmen kann daher nicht das gesamte SoS beeinflussen. Die Entscheidungen sowie die Auswirkungen dieser unterscheiden sich von der jeweiligen Perspektive. Ein Wechsel zwischen der SoS- und der Systemperspektive verdeutlicht den Einflussbereich und ermöglicht eine ganzheitliche Gestaltung von SoS unter Berücksichtigung der Entscheidungsebenen.

A5) Gestaltungsmethoden strukturieren und bereitstellen: Je nach Einfluss auf das SoS nimmt der Konkretisierungsgrad der Gestaltung zu oder ab. Auch die Entwicklungsaspekte unterscheiden sich. Dies erfordert angepasste Methoden, die im jeweiligen Gestaltungskontext angewendet werden können. Sie müssen strukturiert und bedarfsorientiert zur Verfügung gestellt werden.

A6) Anwendbarkeit sicherstellen: Das Rahmenwerk muss anwendbar sein und Unternehmen bei der Gestaltung von SoS unterstützen. Die Tätigkeiten müssen in einem angemessenen Aufwand-Nutzen-Verhältnis stehen, um die jeweiligen Ziele zu erreichen.

Handlungsfeld 3: Wissensmanagement für die Gestaltung von SoS

A7) Wissenstransfer ermöglichen: Um die Gestaltung von SoS über Unternehmensgrenzen hinweg zu ermöglichen, muss eine einheitliche Grundlage geschaffen werden, die die Kommunikation und den Austausch von Wissen zwischen unabhängigen Unternehmen ermöglicht. Diese muss die Weiterentwicklung von SoS über Unternehmensgrenzen hinweg fördern und eine Ausgangsbasis zur Standardisierung von Lösungen sein.

A8) Bereitstellung von SoS-Lösungswissen: Bei der Gestaltung von SoS stehen Unternehmen vor Herausforderungen, die sich in den charakteristischen Merkmalen von SoS zeigen. Das kollektive Wissen wird benötigt, um die Gestaltung hinsichtlich der Merkmale zu unterstützen. Dieses muss in Form von explizitem Wissen aufbereitet werden, welches wiederverwendet werden kann und die Barrieren der Kooperation (u. a. Know-How-Schutz) berücksichtigt.

3 Stand der Forschung

Das Ziel des Kapitels *Stand der Forschung* ist es, einen Überblick über den aktuellen Forschungsstand zu den drei identifizierten Handlungsfeldern zu geben. Dies umfasst eine Zusammenfassung der wichtigsten Ergebnisse bereits veröffentlichter Arbeiten und eine Beschreibung der offengebliebenen Fragestellungen. Entsprechend der Phase *Zielsetzung einer Lösung* nach der DSRM bietet dieses Kapitel einen Überblick über die bestehende Literatur und stellt gleichzeitig eine geeignete Ausgangsbasis für die Erarbeitung einer neuen Lösung dar, die auf dem Wissen über die Theorie aufbaut. Vor dem Hintergrund der Problemanalyse werden zunächst in **Kapitel 3.1** Ansätze zur Beschreibung von SoS untersucht. Diese geben einen Überblick über das Verständnis und zeigen die Vielfalt von SoS in der Literatur. **Kapitel 3.2** untersucht Ansätze zur Gestaltung von SoS und analysiert sowohl Ansätze aus dem Bereich der SoS als auch angrenzender Themengebiete, wie aus dem Bereich der CPS. Zur Unterstützung der Gestaltung werden wissensbasierte Ansätze in verschiedenen Arbeiten aufgegriffen. **Kapitel 3.3** gibt einen Überblick über wissensbasierte Ansätze zur Unterstützung der SoS Gestaltung.

3.1 Ansätze zur Beschreibung von SoS

Im Folgenden werden verschiedene Ansätze beschrieben, die die Beschreibung und Charakterisierung von SoS zum Ziel haben. In Kapitel 3.1.1 werden zunächst Ansätze zur Charakterisierung von SoS beschrieben. Charakteristische Merkmale werden in unterschiedlichen Ansätzen als Ausgangsbasis genutzt, um SoS zu klassifizieren. Bestehende Ansätze zur Klassifikation werden in Kapitel 3.1.2 beleuchtet.

3.1.1 Charakterisierungen von SoS

Zahlreiche Literaturquellen befassen sich mit dem Thema SoS und beschreiben SoS, wie in der viel zitierten Arbeit von MAIER, anhand von Charakteristika [Mai98]. Die Arbeit wurde in mehreren Ansätzen aufgegriffen und weiterentwickelt. Eine Zusammenfassung der Charakterisierung erfolgt in einer nachfolgenden Arbeit von MAIER, die im Folgenden vorgestellt wird. Im Anschluss wird die Charakterisierung nach BOARDMAN und SAUSER sowie der Ansatz nach ABBOTT beschrieben, in denen ergänzende charakteristische Merkmale von SoS beleuchtet werden.

3.1.1.1 Charakterisierung nach MAIER

Viele Forschungsarbeiten im Kontext von SoS sind zurückzuführen auf die Arbeit nach MAIER [Mai98]. Die Arbeit stützt sich auf die Analyse von zwei Anwendungsbereichen, in denen verteilte Systeme in unterschiedlichen Managementstrukturen vorkommen. Zum einen Kommunikationsnetzwerke, wo zwischen dem unabhängigen Internet und dem

eher zentral gemanagten Telefonnetz Unterschiede deutlich werden. Zum anderen das integrierte Luftverteidigungssystem mit unterschiedlichen Kontrollstrukturen.

In der Arbeit definiert MAIER ein SoS als eine Sammlung unabhängiger Systeme, die zusammenarbeiten, um ein übergeordnetes Ziel zu erreichen. Diese Systeme haben einen bestimmten Bedarf hinsichtlich des Designs, der Entwicklung und des Betriebs. Der Bedarf ist zurückzuführen auf bestimmte Merkmale, die MAIER in der 1998 veröffentlichten Arbeit beschreibt. Im Gegensatz zu der bis dahin akzeptierten Beschreibung von SoS als *große Systeme*, die eine große Komplexität aufzeigen und über gewisse Distanzen verteilt sind, stellt MAIER konkrete Merkmale hervor, die SoS von Systemen unterscheiden. Diese sind ein Grund, warum die Entwicklung von SoS sich anders verhält als die Entwicklung von eigenständigen Systemen.

Aufbauend auf seiner ursprünglichen Veröffentlichung fasst MAIER in einem nachfolgenden Beitrag die folgenden fünf charakteristischen Merkmale zusammen, die Systeme von SoS unterscheiden [Mai15, S. 13]:

Operative Unabhängigkeit (Operational independence): Ein SoS besteht aus einzelnen Systemen, die unabhängig voneinander existieren und jeweils einen eigenen Nutzen haben. Die einzelnen Systeme eines SoS haben einen eigenen Zweck und sind in der Lage, eigenständig nützliche Operationen auszuführen.

Managementunabhängigkeit (Managerial independence): Die einzelnen Systeme innerhalb eines SoS können unabhängig voneinander arbeiten, um sowohl ihre eigene Funktion als auch das übergeordnete Ziel des integrierten Gesamtsystems zu erreichen. Diese einzelnen Systeme sind unabhängig und werden von unterschiedlichen Organisationen betrieben. Für die Zusammenarbeit im SoS werden sie angeworben und in das SoS integriert.

Geografische Verteilung (Geographic distribution): Die einzelnen Systeme sind oft geografisch weit verteilt. In der Regel werden Informationen und Wissen miteinander ausgetauscht. Ein Austausch von physischer Masse oder Energie findet seltener statt.

Emergentes Verhalten (Emergent behavior): Das SoS erfüllt Funktionen und Zwecke, die keines der eigenständigen Systeme selbst erfüllen kann. Durch die Interaktion entwickeln sich Verhaltensweisen, die erst durch das Zusammenspiel im SoS entstehen können. Diese Verhaltensweisen werden auch als emergente Eigenschaften bezeichnet.

Evolutionäre Entwicklung (Evolutionary development): Ein SoS ist nie vollständig oder abgeschlossen. Die Entwicklung dieser Systeme erfolgt evolutionär über die Zeit hinweg. Dabei werden Erfahrungswerte genutzt sowie Struktur, Funktion und Zweck hinzugefügt, entfernt oder verändert. Ein SoS entwickelt sich im Laufe der Zeit immer weiter [Mai15].

Als Ergänzung argumentiert MAIER, dass insbesondere die operative Unabhängigkeit und die Managementunabhängigkeit sowie das Merkmal der Emergenz essenzielle Charakteristika von SoS sind. Die geografische Verteilung und evolutionäre Entwicklung sind typisch, aber nicht zwingend [Mai15, S. 13].

Bewertung: Die Grundlagenarbeit von MAIER fasst die Besonderheiten von SoS zusammen und stellt die Unabhängigkeit der einzelnen Systeme eines SoS heraus. MAIER argumentiert, dass aus den charakteristischen Merkmalen von SoS ein bestimmter Bedarf an die Entwicklung abgeleitet werden kann, der sich von Systementwicklungen unterscheidet. Insbesondere die operative Unabhängigkeit und Managementunabhängigkeit sowie die Emergenz werden hervorgehoben, wobei eine Begründung für diese Aussage fehlt. Auch auf die Ausprägung oder Messbarkeit dieser Merkmale wird nicht konkret eingegangen.

3.1.1.2 Charakterisierung nach BOARDMAN und SAUSER

Die Charakterisierung nach BOARDMAN und SAUSER stützt sich auf etablierte Literatur zu komplexen Systemen und der intensiven Analyse von komplexen Netzwerken, wie dem World Wide Web [BS06, S. 121]. Diese Netzwerke sind in verschiedenen Domänen zu finden und weisen Gemeinsamkeiten auf, die in der Arbeit in fünf Merkmalen zusammengefasst werden:

Autonomie (Autonomy): Ein System of Systems existiert, um eine Aufgabe zu erfüllen. Obwohl es gewisse Zwänge gibt, sollten diese nicht die Natur oder Leistungsfähigkeit der Teilsysteme verletzen. Wenn dies der Fall wäre, würde das System die Kooperation im SoS aufgeben und durch ein anderes ersetzt werden müssen. Die Autonomie beschreibt daher Systeme, die ihre eigenen Zwecke erfüllen können, ohne im Rahmen der Interaktion im SoS beeinträchtigt zu werden.

Zugehörigkeit (Belonging): Jedes System bildet im SoS neue Beziehungen zu anderen Systemen und muss sich zum Teil für die Interaktion im SoS verändern. Sie müssen überzeugt werden, den Wert des SoS anzunehmen, ohne ihre eigene Identität oder Leistungsfähigkeit zu verlieren. Die Zugehörigkeit zum SoS erhöht den Wert des Systems und ermöglicht es, den Zweck effektiver zu erreichen. Dies macht die Erreichung des übergeordneten Ziels wahrscheinlicher.

Konnektivität (Connectivity): In einem SoS mit eigenständigen Systemen bedarf es einer dynamischen Bestimmung der Konnektivität. Die Verbindungen entstehen und verschwinden je nach Bedarf, um den autonomen Charakter der einzelnen Systeme aufrecht zu erhalten. Im Gegensatz zu Systemen, in denen die Konnektivität zwischen Teilsystemen gleichzeitig erstellt wird, ist dies in SoS nicht möglich. Dies hängt u. a. damit zusammen, dass im SoS bereits Systeme vorhanden sind (legacy Systeme).

Vielfältigkeit (Diversity): Ein SoS ist vielfältig. Es muss je nach Bedarf auf vielfältige Ungewissheit reagieren und sich anpassen können. Im Gegensatz zu eigenständigen Systemen mit begrenzter Funktionalität, welche durch den Anwendungsbereich und das Design vordefiniert sind, weist ein SoS eine Vielfalt an Funktionen auf.

Emergenz (Emergence): Ein SoS muss eine hohe Emergenz aufweisen. Diese ist erforderlich, um eine umfassende Funktionsfähigkeit für Bedarfe aufzuweisen, die nicht immer offensichtlich sind. Die Emergenz ist inhärent durch die bereits aufgeführten Merkmale von SoS vorhanden: die Autonomie der Systeme, die Entscheidung über Zugehörigkeit, erweiterte Konnektivität und die Vielfältigkeit im Verhalten des SoS [BS06].

Bewertung: Die Charakterisierung nach BOARDMAN und SAUSER ergänzt die Merkmale von MAIER um weitere Aspekte, die ein SoS von Systemen unterscheiden. Die Ausführungen stützen sich auf die Analyse von sowohl technischen SoS als auch soziotechnischen SoS und organisationalen Netzwerken, wie die Staaten von Indien. Dies hat zur Folge, dass die Ausführungen sehr abstrakt erfolgen und wenig zum Verständnis für Anwender im Rahmen von technischen SoS beitragen.

3.1.1.3 Merkmale nach ABBOTT

Einen Schwerpunkt auf die strukturellen und qualitativen Unterschiede zwischen Systemen und einem SoS legt ABBOTT in seiner Arbeit [Abb06]. Ein zentraler struktureller Unterschied zwischen Systemen und einem SoS besteht darin, dass keine klassische Hierarchie existiert, die die Struktur und Organisation vorgibt. Vielmehr beschreibt ein SoS ein Netzwerk, in dem einzelne Systeme interagieren. Es gibt eine Umgebung vor, die sich weiterentwickelt und offen für neue Verbindungen ist. Diese Art der Betrachtung erklärt ABBOTT anhand von drei Merkmalen von SoS:

Offen nach oben (Open at the top): Ein SoS unterscheidet sich von Systemen dadurch, dass es kein festes oberstes Element gibt. Ein SoS fördert die fortlaufende Integration von neuen Anwendungen, sodass es keine maximale Ausprägung nach oben gibt.

Offen nach unten (Open at the bottom): Aufgrund der kontinuierlichen Weiterentwicklung von SoS ist auch die unterste Ebene nicht festgesetzt. Jederzeit können durch Technologieveränderungen auch Systeme auf unterster Ebene ausgetauscht oder verändert werden. Ein Beispiel ist die Umstellung der Telekommunikation von kabelgebundenen Systemen auf drahtlos Netze. Die Weiterentwicklung des Telekommunikations-SoS wäre nicht möglich, wenn die kabelgebundene Technik festgelegt wäre.

Kontinuierlich, aber langsam sich weiterentwickelnd (Continually but slowly evolving): Ein System von Systemen ist immer im Wandel aufgrund von Veränderungen im Umfeld, in dem es arbeitet. Dieser Wandel kann durch technologische Fortschritte, neue Nutzungsmöglichkeiten, aktualisierte Standards oder veränderte Schnittstellen verursacht werden. Um relevant zu bleiben, muss ein System von Systemen in der Lage sein, sich an neue Anforderungen anzupassen und seine Funktionen entsprechend zu erweitern oder

zu verändern. Diese Änderungen müssen langsam erfolgen, damit ein System sich entsprechend der Veränderungen anpassen kann.

Die genannten Merkmale führen dazu, ein SoS als Umgebung zu verstehen, die sich kontinuierlich verändert und bestimmte Rahmenbedingungen vorgibt. Im weiteren Verlauf erläutert ABBOTT die genannten Merkmale anhand eines Beispiels. Hierzu wird der United States Postal Service (USPS) herangezogen. Demnach ist das USPS offen für die Integration von mehreren Aktivitäten, die sich durch die Weiterentwicklung der Umgebung gezeigt haben. Zum einen wird das SoS für die Auslieferung von Katalogen für unterschiedliche Industrien geöffnet. Zum anderen wird auch die Auslieferung von Paketen für den Versandhandel (bspw. Amazon) unterstützt. Beide Aktivitäten waren im Ursprung nicht vordefiniert. Gleichzeitig ist das SoS nach unten offen und neue Technologien, wie ein optischer Sensor im Sortiersystem, können eingeführt werden. Das Merkmal der Evolution wird anhand der langsam aber kontinuierlich eingeführten neuen Semantik von Postleitzahlen argumentiert, die nach und nach erfolgte, ohne die beteiligten Systeme zu überfordern.

Bewertung: ABBOTT geht in seiner Arbeit vor allem auf die strukturellen Merkmale von SoS ein. Dabei wird der Aspekt der fehlenden Hierarchie ausführlich erläutert. Andere Aspekte von SoS, wie die Managementunabhängigkeit, werden nicht berücksichtigt. Während die Merkmale detailliert beschrieben sind, ist das genannte Beispiel weniger hilfreich. Wird das Merkmal der Managementabhängigkeit zugrunde gelegt, bspw. aus der Definition nach MAIER, wäre das genannte Beispiel von USPS kein SoS, da es von einer einzelnen Organisation gemanagt wird. Wie in Kapitel 2.2.2 erläutert, sind auch die Veränderungen von SoS immer dynamischer, sodass das Merkmal der langsamen Weiterentwicklung nicht vollumfänglich zutrifft.

3.1.2 Klassifikationen von SoS

Klassifikationen sind sinnvoll, wenn Gemeinsamkeiten innerhalb von Klassen bestehen und diese Klassen bestimmte Bedarfe an Handlungsweisen in der Gestaltung aufweisen [Mai98, S. 267]. In der Literatur werden unterschiedliche Begriffe für Klassifikationen oder Gruppierungen von Objekten genutzt. Je nachdem, ob die Klassen empirisch oder konzeptuell abgeleitet wurden, werden Begriffe wie Taxonomie oder Typologie verwendet. Wie NICKERSON et al. in einer fundierten Literaturanalyse über Taxonomien festgestellt haben, wird dies in der Literatur jedoch nicht immer einheitlich gemacht [NVM13, S. 338]. Daher werden im Rahmen dieses Kapitels allgemein Klassifikationen untersucht, die sowohl empirisch abgeleitete Taxonomien als auch konzeptuell erarbeitete Klassifikationen sein können.

Im Folgenden werden drei Klassifikationen von SoS vorgestellt. Zunächst wird die Klassifikation von SoS nach MAIER beschrieben. Diese ist die Grundlage für weitere Forschungsarbeiten im Kontext von SoS und wurde vielfach aufgegriffen. Eine Ergänzung zur Taxonomie nach MAIER stellt die Arbeit nach DAHMANN und BALDWIN dar, die im

Anschluss vorgestellt wird. Die ganze Vielfalt von SoS wird im dritten Ansatz in der Arbeit von TEKINERDOGAN abgebildet. Der Ansatz der multidimensionalen Klassifikation greift eine Vielzahl an Merkmalen auf und bringt diese in einen Zusammenhang.

3.1.2.1 Taxonomie nach MAIER

Die Arbeit nach MAIER hat zum Ziel, die wesentlichen Merkmale von SoS zu identifizieren und diese als Ausgangsbasis für die Typisierung von SoS zu nutzen. Wie in Kapitel 3.1.1.1 bereits beschrieben, stellt MAIER insbesondere die Unabhängigkeit von SoS als Merkmal heraus. Dazu gehört die Managementunabhängigkeit als auch die operative Unabhängigkeit der einzelnen Systeme im SoS. Diese Merkmale nutzt Maier, um drei Klassen von SoS zu beschreiben. Die SoS der jeweiligen Klasse weisen gewisse Ähnlichkeiten auf und werden wie folgt charakterisiert:

Gesteuerte SoS (directed SoS): Gesteuerte SoS sind integrierte SoS, die bestimmte Zwecke verfolgen. Während des langfristigen Betriebs werden sie zentral verwaltet, um diese Zwecke und potenzielle neue Zwecke zu erreichen. Obwohl die einzelnen Systeme in der Lage sind, unabhängig zu arbeiten, unterliegen sie der zentralen Steuerung, um das übergeordnete Ziel zu erreichen.

Kollaborative SoS (collaborative SoS): Im Gegensatz zu gesteuerten SoS, gibt es in kollaborativen SoS keine zentrale Autorität, die das SoS leitet. Stattdessen sind die einzelnen Systeme operativ unabhängig und arbeiten mehr oder weniger freiwillig zusammen, um die vereinbarten gemeinsamen Ziele zu erreichen.

Virtuelle SoS (virtual SoS): In virtuellen SoS gibt es weder eine zentrale Verwaltung noch vorab festgelegte Ziele. Stattdessen interagieren die Systeme willkürlich und es entsteht ein stark emergentes Verhalten. Dieses Verhalten kann wünschenswert, aber auch unerwünscht sein. Die Aufrechterhaltung dieses übergeordneten Verhaltens erfolgt durch unsichtbare Mechanismen, die nicht gesteuert werden können.

Die drei Klassen von SoS werden stark anhand der Managementunabhängigkeit von gesteuerten bis zu gänzlich ungesteuerten (virtuellen) SoS differenziert. Im weiteren Verlauf evaluiert MAIER den Einfluss der Klassifikation auf einzelne Beispiele. Eines der Beispiele ist die Internationale Raumfahrtstation (ISS). Diese wird von unterschiedlichen Organisationen aus unterschiedlichen Nationen betrieben, die gemeinsame Ziele verfolgen. Die Zusammenarbeit im SoS kann daher als kollaborative Zusammenarbeit unterschiedlicher Organisationen beschrieben werden. Dennoch sind die einzelnen Module stark abhängig voneinander und der Wegfall eines Moduls könnte die gesamte Funktionsfähigkeit des SoS beeinflussen. Die Architektur der ISS gleicht eher der Architektur eines eigenständigen Systems, welches integrativ entwickelt wurde. Daher trifft die Klassifikation als kollaboratives SoS nicht vollständig zu, da die operative Unabhängigkeit nicht gegeben ist und es daher eher als System einzustufen ist.

In einem weiteren Beispiel wird ein Verteidigungssystem analysiert, welches klassischerweise unter der zentralen Kontrolle einer Nation steht. In dem SoS mit zentraler Kontrolle können einzelne Systeme dazu gezwungen werden, sich entgegen ihrem eigenen Ziel zu verhalten, um das Gesamtziel des SoS zu unterstützen. Dies ist nach MAIER in kollaborativen SoS eher unwahrscheinlich, da die Systeme nur freiwillig in dem SoS interagieren.

Als Beispiel für ein virtuelles SoS wird das Internet genannt. Dieses ist ursprünglich als kollaboratives Netzwerk entstanden, hat sich jedoch im Laufe der Zeit zu einem weitverbreiteten Netzwerk entwickelt. Dieses verfolgt nicht mehr ausschließlich den ursprünglichen Zweck der Wissenschaftskommunikation und ruft teilweise negative Effekte hervor [Mai98].

Bewertung: Die Klassifikation nach MAIER basiert auf relevanten Merkmalen von SoS, die Auswirkungen auf das Design haben. Am Beispiel der Managementunabhängigkeit werden die Einflussmöglichkeiten im Rahmen der Gestaltung beschrieben, die sich je nach Ausprägung unterscheiden. Die Klassifikation strukturiert die Vielfalt von SoS in drei Klassen und ist somit verständlich und anwendbar. Insbesondere die operative Unabhängigkeit wird jedoch als schwierig zu bewertendes Merkmal hervorgehoben und sorgt für Herausforderungen in der Einordnung. Zusätzlich können sinnvolle Gestaltungsrichtlinien für eine Klasse mit der passenden Managementunabhängigkeit unbeachtet bleiben, wenn die operative Unabhängigkeit des betrachteten Systems mit einbezogen wird und somit die Einordnung in die Klasse nicht vollends zutrifft.

3.1.2.2 Erweiterte Klassifikation nach DAHMANN und BALDWIN

Aufbauend auf der Klassifikation nach MAIER beschreiben DAHMANN und BALDWIN eine weitere Klasse von SoS, die als Ergänzung der Klassen *gesteuerter*, *kollaborativer* und *virtueller SoS* gilt. Der Bedarf für eine neue Klasse resultiert aus den Analysen weiterer SoS Beispiele. Demnach existieren vermehrt SoS, die aus eigenständigen Systemen bestehen, aber andere Organisationsstrukturen aufweisen als *kollaborative SoS*. Diese SoS organisieren sich in Interessensgemeinschaften und weisen dennoch unabhängige Systeme auf. Diese Klasse von SoS wird wie folgt charakterisiert:

Föderiertes SoS (acknowledged SoS): Ein *föderiertes SoS* hat anerkannte Ziele und einen designierten Manager, der das SoS leitet. Die einzelnen Systeme behalten ihre Unabhängigkeit und haben ihre eigenen Ziele und Finanzierungen. Zudem werden sie eigenständig entwickelt. Weiterentwicklungen der einzelnen Systeme basieren auf Bedarfen, die sich aus dem Zusammenspiel im SoS ergeben.

Die drei Klassen nach MAIER sowie die Erweiterung um die Klasse der *föderierten SoS* wurde als Taxonomie in den ISO-Standard 21841 überführt. Die Taxonomie führt die vier Klassen von SoS zusammen, die sich in den Managementstrukturen und der operativen Unabhängigkeit der einzelnen Systeme unterscheiden [ISO21841]. Bild 3-1 visuali-

siert die Unterschiede der jeweiligen Klassen von SoS anhand der unterschiedlichen Beziehungsstrukturen zwischen Organisationen und den interagierenden Systemen in Anlehnung an NCUBE und LIM [NL18, S. 113].

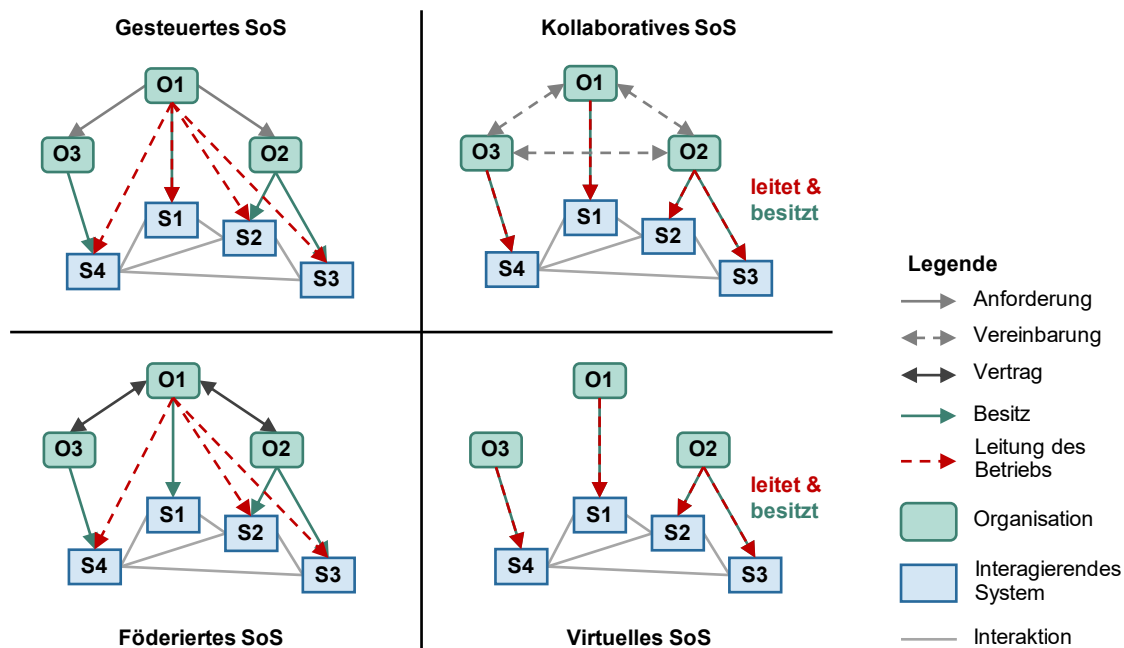


Bild 3-1: Taxonomie nach DAHMANN und BALDWIN [DB08], Darstellung in Anlehnung an NCUBE und LIM [NL18, S. 113]

Bewertung: Die Taxonomie gibt einen guten Überblick über unterschiedliche Klassen von SoS, die anhand der Unabhängigkeitsstrukturen unterschieden werden. Während die Einordnung in *virtuelle SoS* klar erscheint, ist insbesondere die Unterscheidung zwischen *gesteuerten*, *kollaborativen* und *föderierten SoS* nicht ganz trennscharf. Wie u. a. AXELSSON feststellt ist in realen Beispielen daher nicht ganz klar, welcher Typ jeweils vorliegt [Axe19b, S. 32]. Dies hängt nicht zuletzt mit der ungenauen Bewertung der operativen Unabhängigkeit zusammen.

3.1.2.3 Multidimensionale Klassifikation nach TEKINERDOGAN

Aufbauend auf einer Analyse der vorhandenen Literatur und von Beispielen, kommt TEKINERDOGAN in seiner Arbeit zu dem Schluss, dass es keinen klaren Konsens über eine Klassifizierung von SoS gibt [Tek19]. Dies hängt nicht zuletzt damit zusammen, dass die bestehenden Klassifikationen einen bestimmten Betrachtungswinkel einnehmen und daher keine vollständige SoS-Klassifikation darstellen. Aus diesem Grund wendet er die Methode einer umfassenden Domänenanalyse an, um die wesentlichen Quellen zum Thema SoS-Klassifikationen zu identifizieren und relevante Merkmale herauszuarbeiten.

TEKINERDOGAN identifiziert insgesamt sieben Perspektiven, die es ermöglichen, SoS zu unterscheiden. Diese umfassen sowohl die Klassifikation von SoS nach DAHMANN und BALDWIN (vgl. Kapitel 3.1.2.2) als auch weitere Perspektiven, wie die Unterscheidung

auf Basis der Anwendungsdomäne. Dabei wird zwischen technischen, sozialen, sozio-technischen und natürlichen SoS unterschieden. Die Vielzahl an Klassifikationen wurden schließlich zusammengeführt und als Feature-Modell dargestellt (Bild 3-2). Die einzelnen Ausprägungen werden dabei als Alternativen dargestellt.

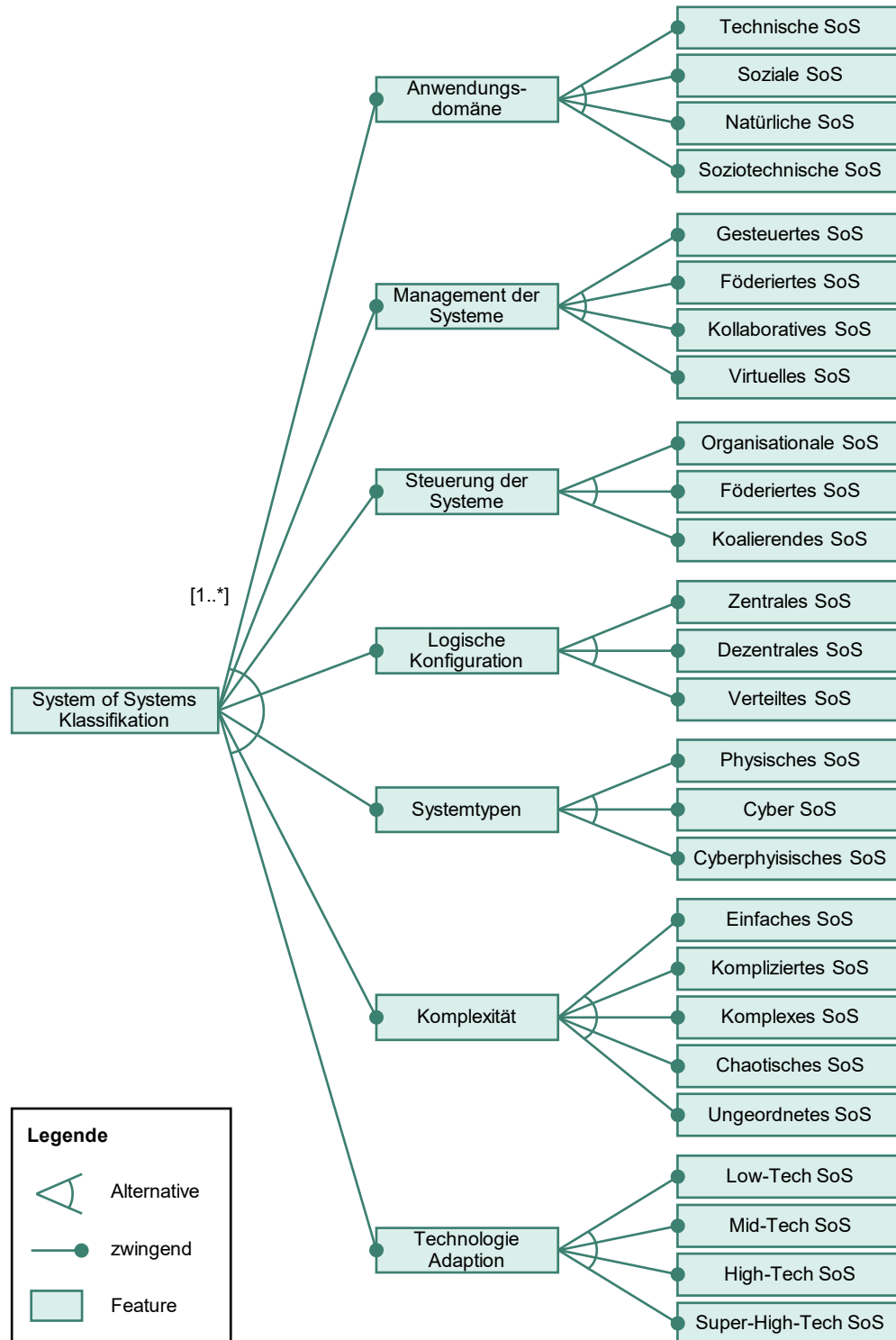


Bild 3-2: Multidimensionale Klassifizierung von SoS nach TEKINERDOGAN [Tek19, S. 283]

Das Modell kann genutzt werden, um SoS in mehreren Dimensionen zu klassifizieren. Durch die Kombination der einzelnen Ausprägungen können insgesamt 8640 Klassen von SoS unterschieden werden [Tek19].

Bewertung: NICKERSON et al. beschreiben die Eigenschaften von nützlichen Taxonomien als prägnant und sparsam. Eine Taxonomie mit zu vielen Dimensionen und Merkmalen in jeder Dimension erfüllt diese Eigenschaft nicht und ist demnach schwierig zu verstehen und anzuwenden [NVM13, S. 341]. Dies trifft auch auf die Taxonomie nach TEKINERDOGAN zu. Die theoretische Anzahl von 8640 Klassen resultiert in einer nicht überschaubaren Anzahl an unterschiedlichen SoS. Selbst eine Einschränkung von Kombinationsmöglichkeiten durch Ausschlussbeziehungen zwischen bestimmten Ausprägungen führt nicht zu einer nützlichen Anzahl an Klassen. Damit kann die Anforderung an eine zielgerichtete Auswahl von Entwicklungsvorgehen für bestimmte Klassen von SoS nicht erfüllt werden.

3.2 Ansätze zur Gestaltung von SoS

Gestaltungsansätze für vernetzte Systeme müssen mehrere Aspekte berücksichtigen, die über die Betrachtung von abgegrenzten Systemen hinausgehen (vgl. Kapitel 2.3). Die in diesem Kapitel beschriebenen Ansätze sind vor diesem Hintergrund entwickelt worden und beschreiben Entwicklungsvorgehen zur Entwicklung von vernetzten Systemen. Zunächst werden in Kapitel 3.2.1 die Entwicklungsansätze vorgestellt, die insbesondere für SoS erarbeitet wurden. Im Anschluss werden in Kapitel 3.2.2 weitere Ansätze aus angrenzenden Forschungsgebieten, wie CPS, vorgestellt.

3.2.1 Ansätze aus dem Forschungsbereich SoS

Das Forschungsgebiet SoS beschäftigt sich seit mehreren Jahren mit Entwicklungsvorgehen, die für SoS geeignet sind. Es wurden sowohl gänzlich neue Ansätze entwickelt, wie bspw. das Wellenmodell nach DAHMANN et al. als auch bestehende Ansätze durch SoS-spezifische Aspekte ergänzt, wie in der ISO 21840. Im Folgenden werden ausgewählte Ansätze vorgestellt, die jeweils auf unterschiedliche Aspekte im Rahmen der Gestaltung von SoS eingehen.

3.2.1.1 Wellenmodell der SoS-Entwicklung nach DAHMANN et al.

Ein Ansatz, der insbesondere die Aspekte der Veränderbarkeit und Evolution von SoS aufgreift ist das Wellenmodell von DAHMANN et al. [DRL+11]. Das Modell beschreibt kontinuierliche Evolutionszyklen von SoS, die durch Updates von einzelnen Aspekten, wie bspw. Systemerweiterungen oder neue Systementwicklungen, erreicht werden. Die Evolutionszyklen sind als Wellen visualisiert und in Bild 3-3 dargestellt.

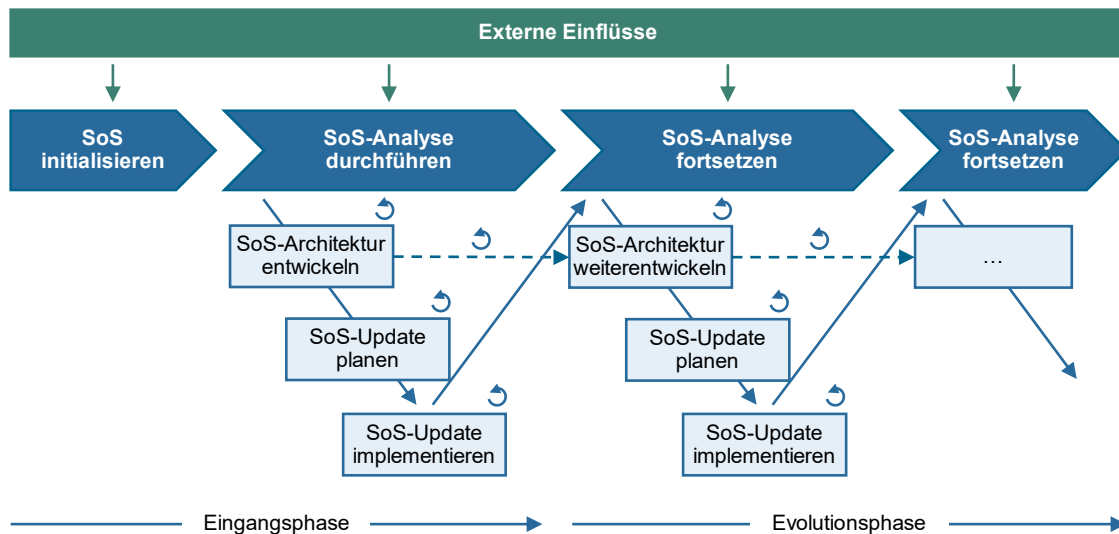


Bild 3-3: Wellenmodell der SoS-Entwicklung nach DAHMANN et al. [DRL+11, S. 215]

Das Wellenmodell beschreibt mehrere Schritte, die durch ein SoS-SE-Team durchzuführen sind. Das Modell kann in zwei Phasen unterschieden werden, die Eingangsphase und die Evolutionsphase. In der Eingangsphase wird das SoS initialisiert, ein erstes Konzept entwickelt und ein erstes Artefakt erstellt. Im Folgenden werden die Schritte der Eingangsphase beschrieben:

- **SoS initialisieren:** In diesem Schritt werden die grundlegenden Informationen für den Start des Gestaltungsprozesses festgelegt. Dazu gehört das Verständnis über die Ziele, Hauptnutzer, Nutzerrollen und -erwartungen sowie die Kernsysteme des SoS.
- **SoS-Analyse durchführen:** Im Rahmen der ersten Analyse wird zunächst der Ist-Stand des SoS bewertet und eine Baseline gezogen. Auf dieser Basis wird ein erster Plan für die Gestaltungsmaßnahmen entwickelt.
- **SoS-Architektur entwickeln:** Die initiale Entwicklung der SoS-Architektur stellt den technischen Rahmen des SoS dar und steht im Zentrum für die Entwicklung und Weiterentwicklung der Funktionen, Beziehungen und Abhängigkeiten im SoS. Zusätzlich wird festgehalten, wie die einzelnen Systeme miteinander interagieren.
- **SoS-Update planen:** In diesem Schritt werden die Updates priorisiert und geplant. In Vorbereitung für das anstehende Update werden Vereinbarungen getroffen und Implementierungs-, Integrations- und Testpläne erstellt.
- **SoS-Update implementieren:** Die Implementierung des SoS-Updates umfasst sowohl die Implementierung und Testdurchführung auf System-Level als auch auf SoS-Level. Dies wird durch das SoS-SE-Team begleitet und überwacht. Am Ende steht eine neue SoS-Baseline, die vom SoS-SE-Team dokumentiert wird.

Die zweite Phase umfasst Schritte, die begleitend zur Evolution des SoS immer wieder durchlaufen werden. Dazu gehören:

- **SoS-Analyse fortführen:** Kontinuierlich werden die wichtigsten Informationen über das SoS analysiert. Dies umfasst den SoS-Zustand sowie die Pläne für das SoS als Grundlage für die künftige SoS-Entwicklung.
- **SoS-Architektur weiterentwickeln:** Auf Basis der initialen Architektur werden Entscheidungen über Veränderungen getroffen, dokumentiert und gesteuert.
- **SoS-Update planen:** Wie in der Eingangsphase wird in der Evolutionsphase jedes Update begleitet und die entsprechenden technischen Pläne erstellt und angepasst.
- **SoS-Update implementieren:** Jede Implementierung wird sowohl auf System-Level als auch auf SoS-Level begleitet und Tests durchgeführt. Dies stellt die Funktionsfähigkeit sicher und berücksichtigt alle unvorhergesehenen Faktoren, die auftreten können.

Bewertung: Das Wellenmodell stellt ein detailliertes Vorgehen dar, welches insbesondere auf die Aspekte der Veränderbarkeit und Evolution von SoS eingeht. Dazu beschreiben DAHMANN et al. ausführlich, welche technischen Prozesse notwendig sind und wie sie im Rahmen der Evolution wiederkehrend anzuwenden sind. Das Vorgehen ist für ein SoS beschrieben, in dem es ein zentrales Management oder ein SoS-Team gibt. Dieses führt die Schritte durch und ist beispielsweise für die Tests auf SoS-Ebene verantwortlich. Die Anwendbarkeit des Vorgehens ist daher auf SoS mit zentralem Management beschränkt. SoS ohne zentrale Managementstrukturen werden nicht berücksichtigt.

3.2.1.2 SoSE-Prozess nach SAGE und BIEMER

SAGE und BIEMER beschreiben in ihrer Arbeit einen SoSE-Prozess, der auf unterschiedlichen Prozessmodellen der Literatur aufbaut. Dazu wurde der aktuelle Stand des Engineerings von Systemverbünden analysiert und in einem ersten Schritt die charakteristischen Merkmale gesammelt. Insgesamt wurden sieben Merkmale von SoS herangezogen, bestehend aus den fünf Merkmalen nach MAIER (vgl. Kapitel 3.1.1.1) sowie den Merkmalen der Selbstorganisation und Adaptierfähigkeit. Diese stellen die Ausgangsbasis zur Ableitung von Anforderungen an den SoSE-Prozess dar. Beispielsweise wurde aus dem Evolutionsmerkmal die Anforderung abgeleitet, dass der Prozess eine inkrementelle Entwicklung ermöglichen muss. Anschließend wurden sieben etablierte Standards und Architekturentwicklungsprozesse, wie die ISO 15288, analysiert und auf die Anforderungen hin überprüft. Da keiner der Standards die Anforderungen an das Design von SoS vollständig erfüllt, wurde der in Bild 3-4 dargestellte SoSE-Prozess entwickelt. Dieser baut auf den bestehenden Prozessen auf und kombiniert diese zu einem Gesamtprozess für das Engineering von SoS.

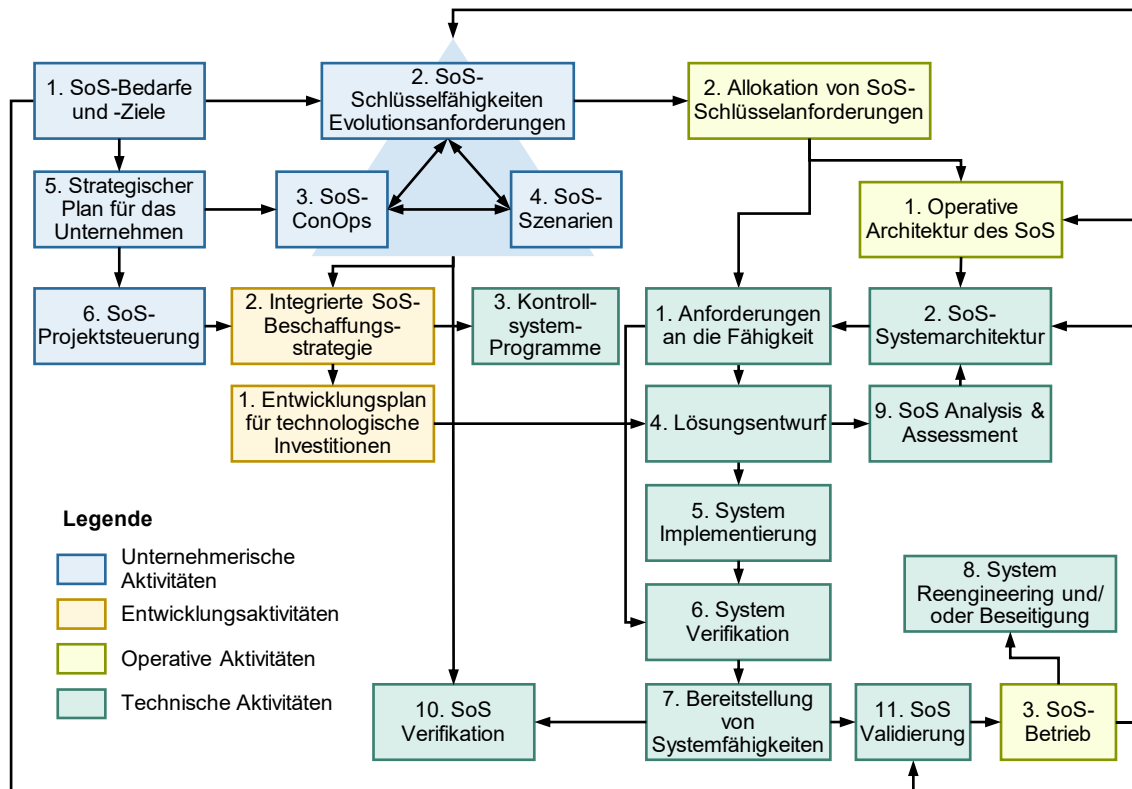


Bild 3-4: SoSE-Prozess nach Sage und Biemer [SB07, S. 15]

Insgesamt umfasst der Prozess vier Teilbereiche: unternehmerische Aktivitäten, Entwicklungsaktivitäten, operative Aktivitäten und technische Aktivitäten. Die **unternehmerischen Aktivitäten** resultieren aus dem Bedarf, die Managementunabhängigkeit zu berücksichtigen und über Unternehmensgrenzen hinweg zu agieren, zu kollaborieren sowie Entscheidungen zu treffen. Dazu gehört, den *strategischen Plan für das Unternehmen* festzulegen. Dies erfolgt unter Berücksichtigung der eigenen Unternehmensrichtlinien und finanzieller, organisatorischer und verwaltungstechnischer Aspekte.

Die **Entwicklungsaktivitäten** sind als ergänzende Aktivitäten im Prozess integriert, um die Evolution der eigenständigen Systeme zu unterstützen. Dabei wird u. a. der *Entwicklungsplan für technologische Investitionen* festgelegt. Dies umfasst die Analyse von neuen Technologien und die Bewertung, ob eine neue Fähigkeit entwickelt werden sollte.

Aufgrund der stetigen Weiterentwicklung (Evolution) von SoS, finden Entwicklung und Betrieb parallel statt. Daher wird der Betrieb durch die **operativen Aktivitäten** sichergestellt. Als eine von drei Aktivitäten wird die *operative Architektur* und die wesentlichen Fähigkeiten des SoS auf oberster Ebene definiert. Dies betrifft als eine der wenigen Aktivitäten das gesamte SoS, ohne den Fokus auf ein eigenständiges System zu legen.

Aufbauend auf den Systems Engineering Prozessen der ISO 15288, werden im Rahmen der **technischen Aktivitäten** die wesentlichen technischen Prozesse für das SoS aufgeführt. Neben der Architekturentwicklung, der Implementierung und der Verifikation des

eigenständigen Systems, werden diese Aktivitäten auch auf das SoS bezogen und bspw. das *Reengineering von Systemen im SoS* mit integriert [SB07].

Bewertung: Der Prozess wurde systematisch auf Basis der charakteristischen Merkmale von SoS erarbeitet. Er adressiert die unterschiedlichen Herausforderungen und unterstützt bspw. ein iteratives Vorgehen durch einen geschlossenen Kreislauf an Aktivitäten. Die hohe Anzahl an Prozessschritten und deren Zusammenhänge verdeutlichen die Vielfältigkeit von SoS-Entwicklungsprojekten. Insbesondere die integrierte Betrachtung von unternehmerischen Aktivitäten und technischen Aktivitäten erscheint vor dem Hintergrund der Managementunabhängigkeit von SoS als sinnvoll. Offen bleibt die Unterscheidung zwischen unterschiedlichen SoS-Klassen und welche Auswirkungen diese auf die einzelnen Aktivitäten haben.

3.2.1.3 SoSE-Methodik nach ADAMS und KEATING

Die SoSE-Methodik nach ADAMS und KEATING hat das Ziel, für unterschiedliche Fragestellungen adaptiert werden zu können und für unterschiedliche Problemstellungen anwendbar zu sein. Dazu soll eine umfassende Problemanalyse durchgeführt werden, um diese für die spezifische Ausrichtung der Methodik zu nutzen [AK11].

Grundlagen der Methodik sind zum einen die grundlegenden Prinzipien der allgemeinen Systemtheorie und zum anderen sieben ergänzende Perspektiven, die jeweils durch mehrere Tätigkeiten ausgeprägt werden (Bild 3-5). Die sieben Perspektiven werden für ein ganzheitliches Verständnis der Problemstellung benötigt und werden flexibel und iterativ angewendet.

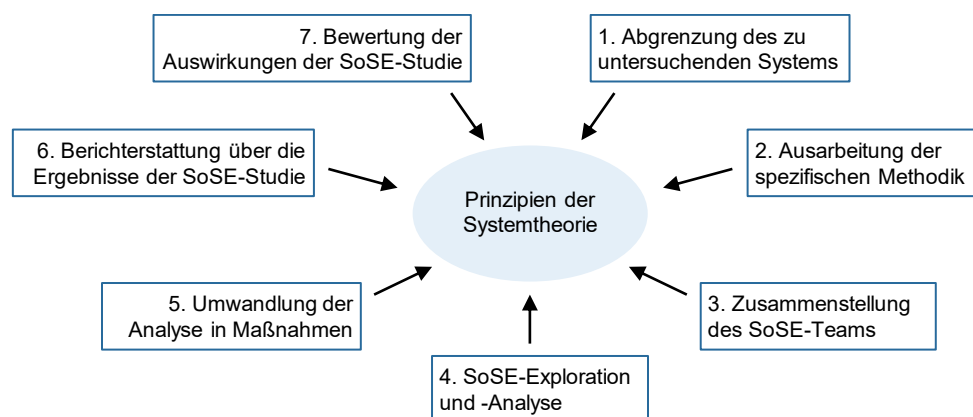


Bild 3-5: Die sieben Perspektiven der SoSE-Methodik nach Adams und Keating [AK11, S. 116]

Das Problem wird in **Perspektive 1** erfasst. Acht Tätigkeiten sind durchzuführen, u. a. die *Untersuchung des Kontextes*, die *Charakterisierung des zu untersuchenden Systems* und der *Domäne*. Zentral ist auch die *Definition des Untersuchungsziels* inklusive des High-Level-Ansatzes und der Annahmen für das Vorhaben.

Perspektive 2 umfasst die Ausarbeitung einer spezifischen Methodik, die auf Grundlage des Problems und des Problemkontextes entwickelt wird. Die Methodik wird auf die Untersuchungsziele abgestimmt und eine *Analysestrategie entwickelt*, die auf einem Konzept für die quantitative und qualitative Untersuchung aufbaut. Bewertungskriterien werden abgeleitet, um die Erreichung des Studienziels zu stützen.

In **Perspektive 3** wird ein SoSE-Team gebildet, welches aufgrund des Wissens und der Fähigkeiten optimal zu der Problemstellung passt. **Perspektive 4** dient der Analyse und Durchführung der Untersuchungsziele anhand eines definierten SoSE-Managementplans. Die entwickelten Studienobjekte werden analysiert und anhand der Ziele bewertet.

Die Resultate der Untersuchung werden in **Perspektive 5** umgesetzt. Im Rahmen der *Definition von Umsetzungszielen, -maßnahmen und -aktivitäten* wird die Art der Umsetzung geklärt. Dies erfordert häufig eine *Anpassung des SoSE-Managementplans*. Zuletzt umfasst die dritte Tätigkeit im Rahmen dieser Perspektive die *Umsetzung der Analyseempfehlungen* aus den Voruntersuchungen.

Perspektive 6 gilt der Dokumentation und Evaluation der Untersuchungen. In **Perspektive 7** wird schließlich die Auswirkung der getroffenen Maßnahmen analysiert und Maßnahmen zur Weiterverfolgung entwickelt [AK11].

Bewertung: Das Vorgehen beschreibt relevante Perspektiven, die für eine Auslegung einer passenden Methodik beachtet werden müssen. Diese machen deutlich, dass je nach Kontext und Problemstellung die Analyse und Bearbeitung des Problems unterschiedlich ist und Anpassungsbedarf besteht. Die Methodik setzt ein hohes Maß an Kenntnissen im Bereich der Systemtheorie und Grundlagen der Methodenentwicklung voraus und es werden wenig Anhaltspunkte zur Umsetzung der Methodik gegeben. Zusätzlich fehlen Beispiele, um das Verständnis für Anwender zu erhöhen.

3.2.1.4 SoS Lifecycle Engineering nach FORTE et al.

Aus dem Bedarf heraus, die zunehmende Interaktion von Produkten in Systemverbünden im Rahmen des Engineerings abzubilden, ist auch der SoSE-Lifecycle Ansatz nach FORTE et al. entstanden [FGD21]. Dieser vereint ein ganzheitliches SoS-Engineering Konzept mit zirkulären und agilen Ansätzen. Begleitend zu den unterschiedlichen Entwicklungsstufen und Veränderungen im SoS werden insgesamt vier SoS-spezifische Prozesse und ein systemspezifischer Prozess in dem Ansatz als wiederkehrende Prozessabfolgen vereint (Bild 3-6).

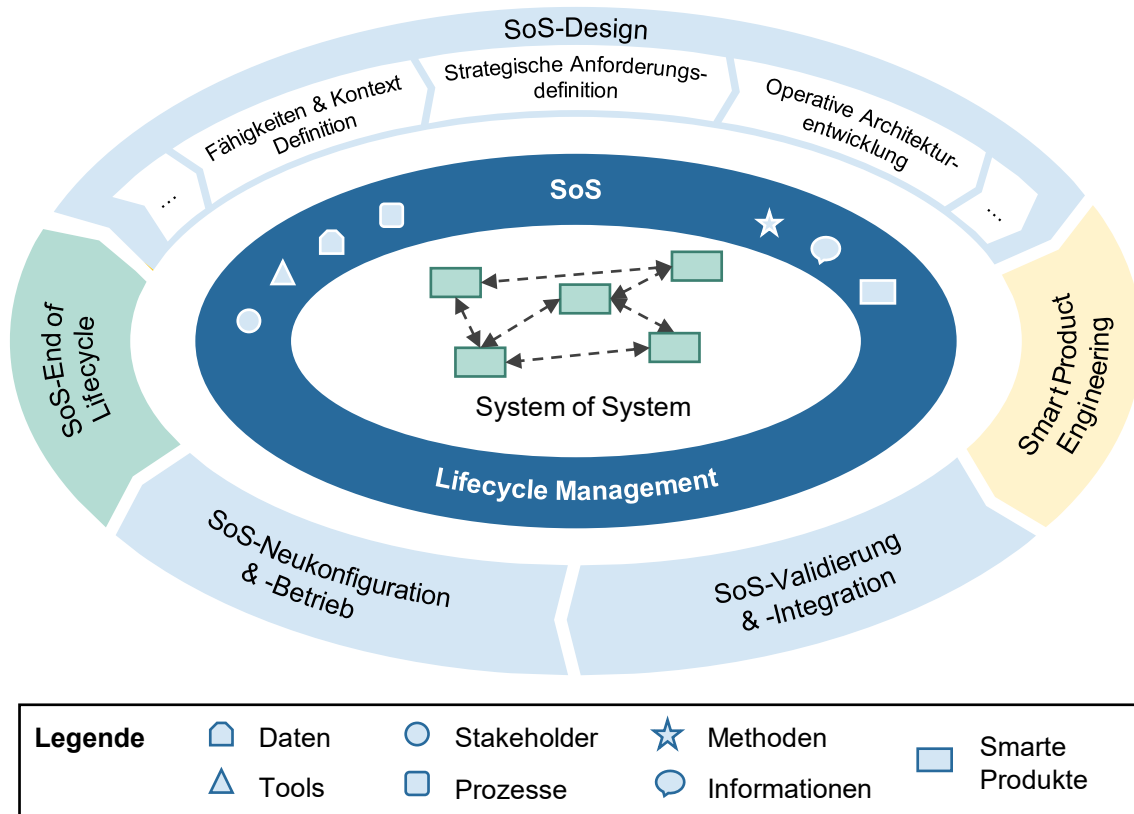


Bild 3-6: SoS-Engineering Lifecycle Ansatz nach FORTE et al. [FGD21, S. 2914]

Im Gegensatz zu Engineering-Ansätzen für eigenständige Systeme, wird im Prozess **SoS-Design** das gesamte SoS gestaltet und bspw. Stakeholder des Ökosystems analysiert. Das Ziel ist ein erstes Konzept, in dem die Fähigkeiten des SoS, aber auch die Schnittstellen zwischen den eigenständigen Systemen definiert werden. Auf Basis der Anforderungen des SoS-Designs werden dann die einzelnen Systeme (**Smart Product Engineering**) gestaltet.

Die **SoS-Validierung und -Integration** ist mit der Ungewissheit über Zusammenhänge durch die dynamische Vernetzung konfrontiert. Um die Absicherung gegenüber Stakeholderanforderungen auf SoS-Ebene zu unterstützen, sollen Model-in-the-loop-Ansätze genutzt werden. Diese greifen auf die Informationen der einzelnen Systeme und des SoS zurück und nutzen diese für Modellsimulationen.

SoS-Neukonfiguration und -Betrieb umfasst die relevanten Prozesse zur Begleitung des SoS im Betrieb. Durch die Vielzahl an verfügbaren Daten der vernetzten Systeme werden sowohl Analysen für die Optimierung des SoS erstellt als auch zukünftige Datengetriebene Systemverbesserungen initiiert. Auch die Planung für weitere Entwicklungsdurchläufe wird in diesem Abschnitt durchgeführt.

Getrieben durch Veränderungen im SoS werden Systemverbesserungen oder Optimierungen notwendig. Dies hat zur Folge, dass einzelne Aspekte ihr Lebensende erreichen. Im **SoS End-of-Lifecycle** Prozess werden diese Aspekte analysiert und entschieden, ob

sie im Rahmen eines weiteren Entwicklungszyklus angepasst und verbessert werden. Ist dies der Fall hat dies eine Rekonfiguration des gesamten SoS zur Folge und Verhaltensweisen müssen innerhalb eines neuen Entwicklungszyklus neu definiert werden.

Bewertung: Der Ansatz nach FORTE et al. betrachtet die SoS-Entwicklung als integratives Design von SoS und eigenständiger Systeme, die kontinuierlich fortgeführt werden. Die Basis sind Informationsmodelle, die über das SoS hinaus als SoS-Lifecycle Modelle für Entscheidungen und Prozesse, wie die Validierung, genutzt werden können. Unklar bleibt, wie die Informationen der einzelnen Systeme zusammengeführt werden und wie dies im Zusammenspiel zwischen unabhängigen Unternehmen umgesetzt werden soll. Auch werden Barrieren dieser integrativen Betrachtung, wie der Know-How-Schutz, nicht adressiert.

3.2.1.5 Rahmenwerk für SoS-Gestaltungsprozesse nach SHAKED und REICH

Das Modellierungsframework für SoS nach SHAKED und REICH stellt einen generischen Ansatz zur Ableitung von Gestaltungsprozessen dar. Es adressiert die zahlreichen Unterschiede von SoS-Entwicklungsprojekten, in dem es ein Vorgehen zur Entwicklung von spezifischen Prozessen für unterschiedliche Fragestellungen bereitstellt. Die Prozessbestandteile werden nach und nach anhand der In- und Outputs wie ein Puzzle zusammengesetzt. Ein Input für einen Prozess wird als Output von einem anderen Prozess bereitgestellt. Passen die In- und Outputs nicht zusammen, wird ein entsprechender Prozess ergänzt, der die benötigten Artefakte liefert. Bild 3-7 beschreibt einen Teil des Vorgehens und visualisiert einen Prozessbaustein.

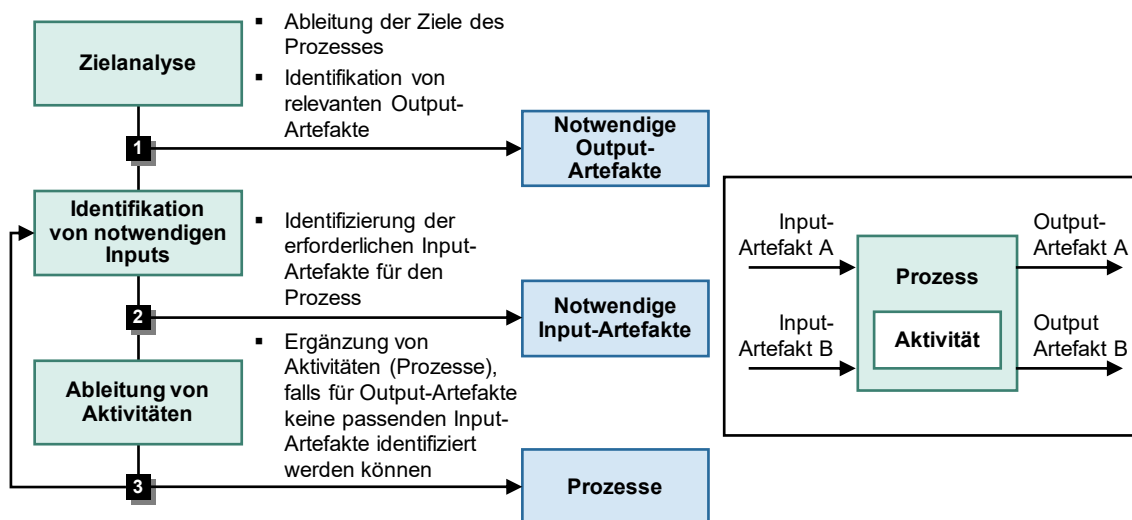


Bild 3-7: Generisches Vorgehensmodell zur Ableitung von Prozessen nach SHAKED und REICH [SR19, S. 564]

SHAKED und REICH wenden das Vorgehen im weiteren Verlauf auf verschiedene Fragestellungen an. Ein Beispiel ist die Zusammenarbeit von Vertragspartnern in SoS. Dazu wenden sie das Vorgehen auf den Austausch von Entwicklungsartefakten zwischen den

beteiligten Partnern an. Die jeweiligen Inputs werden innerhalb der einzelnen Prozessbausteine in Outputs gewandelt und wiederum für den nächsten Prozess genutzt. Es entsteht ein Zusammenspiel zwischen drei Vertragspartnern. Festgehalten ist, welche Artefakte in welchem Zustand ausgetauscht werden (Bild 3-8).

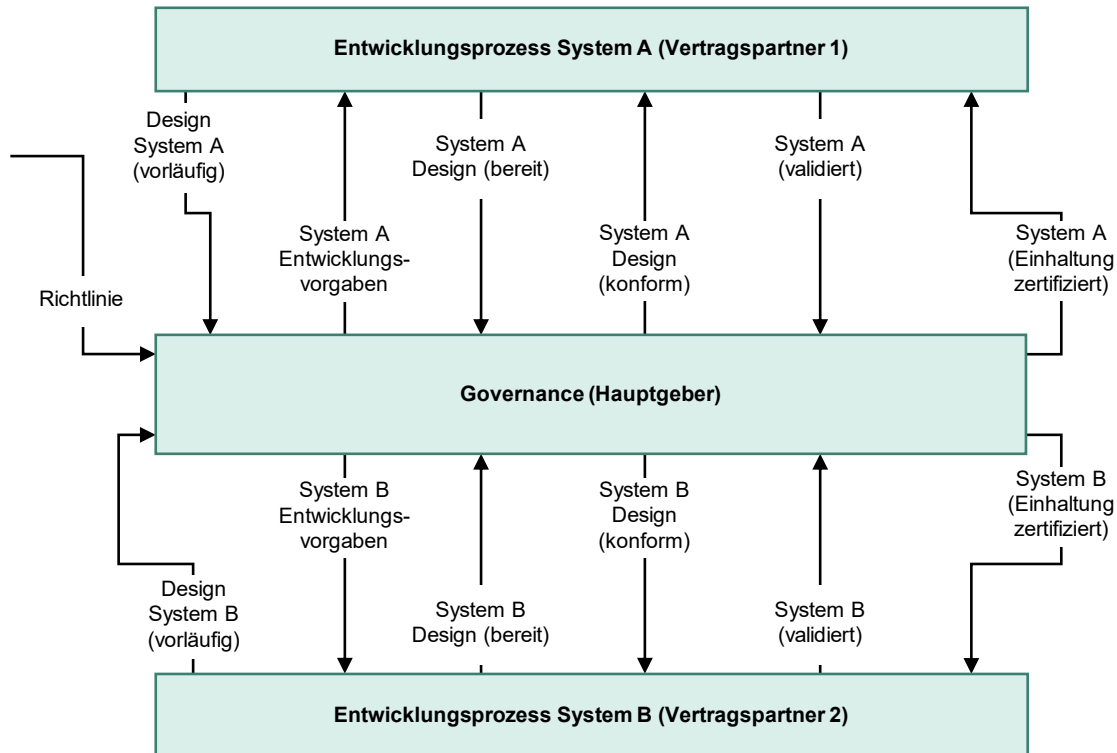


Bild 3-8: Beispielprozess für den Austausch von Entwicklungsartefakten zwischen Vertragspartnern im SoS nach SHAKED und REICH [SR19, S. 573]

Bewertung: Das Vorgehen liefert einen generischen Rahmen, um Prozesse auf bestimmte Bedarfe anzupassen und einzelne Prozessschritte miteinander zu kombinieren. Dies ist insbesondere für SoS sinnvoll, bei denen unterschiedliche Herausforderungen zu bewältigen sind und aufgrund der vielfältigen Unterschiede Anpassungen in den Gestaltungsvorgehen notwendig sind. Der Ansatz ist sehr generisch und daher auch wenig konkret für Unternehmen. SHAKED und REICH geben keine weiteren Anhaltspunkte für notwendige Aktivitäten und setzen somit viel Vorwissen bei den Anwendern für die Ausgestaltung des Entwicklungsvorgehen voraus. Auch auf den Aspekt der unternehmensübergreifenden Zusammenarbeit wird nicht eingegangen.

3.2.1.6 Modellbasierter SoSE-Ansatz nach BAEK et al.

Ein modellbasiertes Vorgehen bietet verschiedene Vorteile, um komplexe Sachverhalte abzubilden (vgl. Kapitel 2.3.3). Um die Vielzahl an Zusammenhängen im Rahmen des SoSE abzubilden, entwickeln BAEK et al. einen modellbasierten Ansatz, der die Entwicklung und Analyse von SoS unterstützen soll [BMS+20]. Der Ansatz baut auf einer generischen Sprache und Methode zur Beschreibung von SoS auf. Die Sprache ist in Form

eines Meta-Modells formalisiert und umfasst Sprachelemente sowie deren Beziehungen untereinander (Bild 3-9). Das formalisierte Sprachmodell unterstützt sowohl SoS-Entwickler als auch -Manager in ihren Aufgaben.

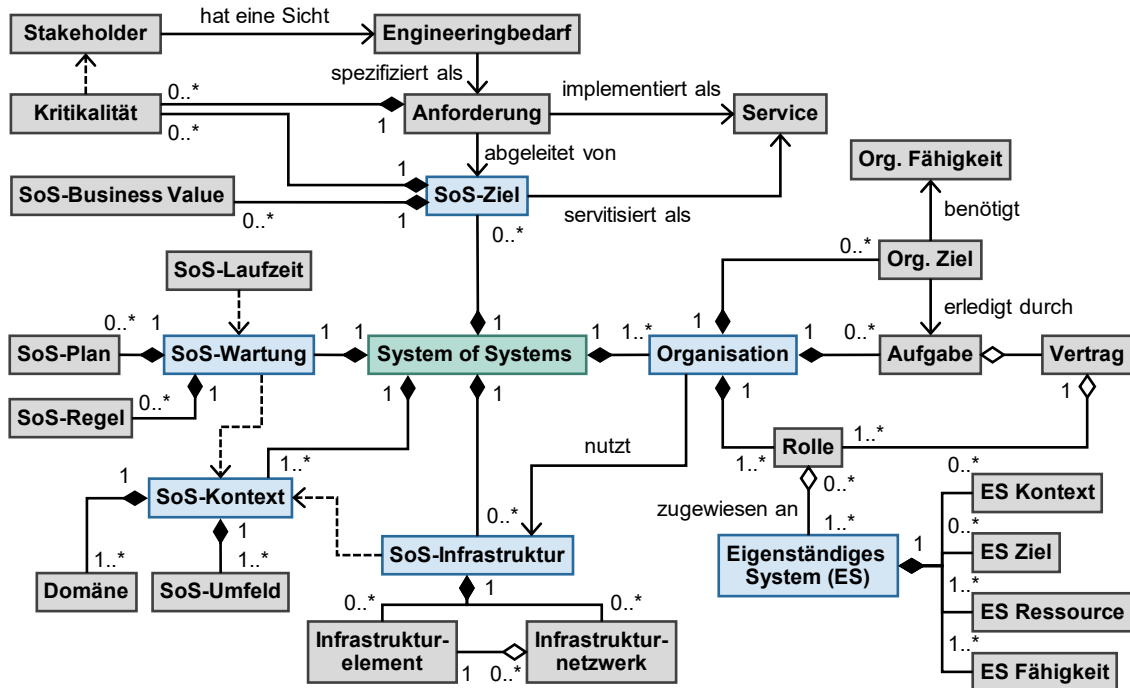


Bild 3-9: Meta-Modell für das SoS-Engineering nach BAEK et al. [BMS+20, S. 338]

Neben der Sprache wird auch eine passende Methode benötigt. Die Methode muss unterschiedliche SoS unterstützen können. Aus diesem Grund beschreiben BAEK et al. ein generisches Vorgehen mit fünf Schritten zur Entwicklung einer passenden SoS-Modellierungsmethode (Bild 3-10). Im **ersten Schritt** werden Anforderungen an die Methode aus Sicht der Analyse, Entwicklung, Simulation und Validierung von SoS abgeleitet. Die Anforderungen müssen ein zu entwickelndes Modell erfüllen, um die Anwender in ihren Tätigkeiten zu unterstützen. Im **zweiten Schritt** wird ein High-Level Architekturmodell aufgebaut und Modelltypen abgeleitet, die die Anforderungen unterstützen. Diese werden im **dritten Schritt** in eine Methode überführt, indem die einzelnen Modelltypen in einen Zusammenhang gesetzt werden. In **Schritt vier** wird die Modellierungsmethode und Sprache in einem Tool umgesetzt und schließlich in **Schritt fünf** getestet und freigegeben.

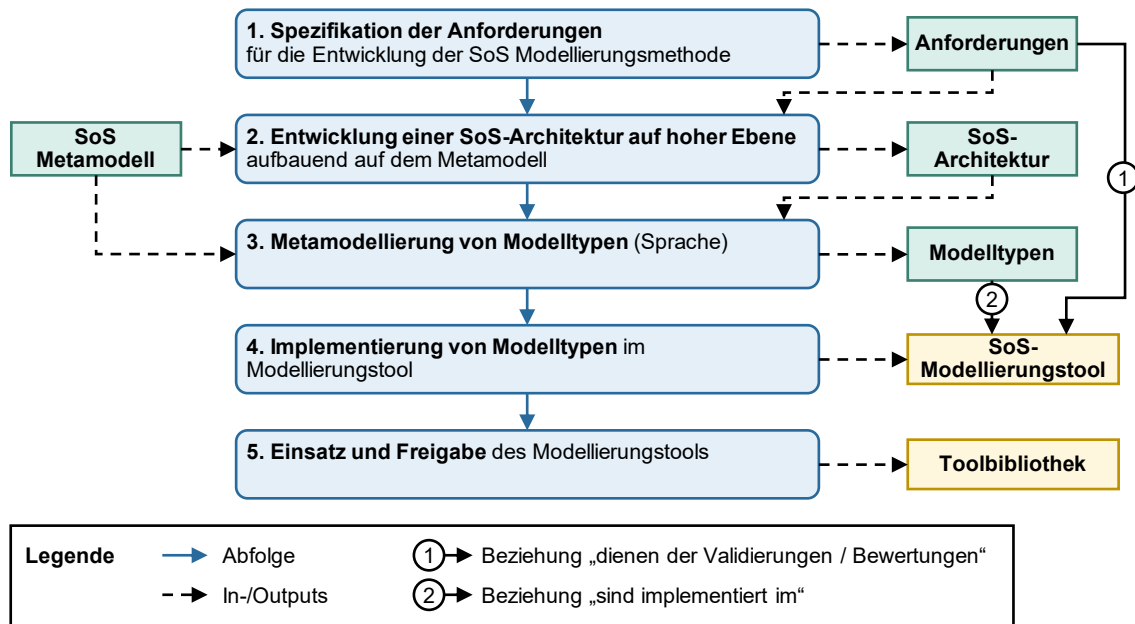


Bild 3-10: Prozess zur Entwicklung einer generischen Modellierungsmethode für das SoSE nach BAEK et al. [BMS+20, S. 339]

Wird das Vorgehen durchlaufen, entsteht eine passende Modellierungsmethode für die zu Beginn gestellten Anforderungen der SoS-Gestaltung. Zusätzlich wird durch den konsequenten Einsatz des Meta-Modells, der Austausch von Informationen zwischen unterschiedlichen Unternehmen ermöglicht, wenn der Zugriff auf das Modell entsprechend von jedem Unternehmen freigegeben ist.

Bewertung: Die frühe Analyse von Anforderungen an die SoS-Gestaltung hat den Vorteil, dass je nach SoS eine passende Methode und Sprache entwickelt wird. So kann auf die unterschiedlichen Bedarfe und Herausforderungen von SoS eingegangen werden. Das generische Meta-Modell, kann die Kommunikation zwischen Unternehmen unterstützen. Voraussetzung ist jedoch die Freigabe und der Zugriff von allen Informationen zwischen den beteiligten Unternehmen. Dies ist insbesondere in Konkurrenzsituationen nicht immer gegeben.

3.2.1.7 Rahmenwerk für die SoS-Synthese, -Bewertung und gesteuerte Entwicklung nach SELBERG und AUSTIN

Ein weiteres Vorgehen zur modellbasierten Gestaltung von SoS beschreiben SELBERG und AUSTIN. Das grundsätzliche Vorgehen basiert auf einem Bottom-Up Ansatz, in dem die bereits bestehenden Systeme und Strukturen in die SoS-Analysen mit einbezogen werden. Dazu werden zunächst die existierenden Systeme formal modelliert, inklusive des Verhaltens, ihrer Eigenschaften und Schnittstellen. Dies ist die Basis für die Analyse und Synthese des SoS und ein wichtiger Input für das Rahmenwerk (Bild 3-11). Das Vorgehen gliedert sich in vier Schritte. Im **ersten Schritt** werden u. a. funktionale Anforderungen, Performance- oder auch Kostenanforderungen gesammelt. Im **zweiten Schritt**

wird ein formales SoS-Modell inklusive der Standards und Schnittstellen sowie Vor- und Nachbedingungen der Interaktionen erstellt. Im Rahmen der Synthese werden zusätzlich bestehende Systemmodelle mit einbezogen. Der **dritte Schritt** dient der Entwicklung der Systemarchitektur. Dabei wird sowohl die Kompatibilität zwischen den Systemen als auch das SoS-Verhalten unter Berücksichtigung der Emergenz betrachtet. Im **vierten Schritt** wird das SoS oder das Update für das SoS implementiert. Das in dem Vorgehen erarbeitete SoS-Modell ist die Basis für weitere Iterationsschleifen. Jede Weiterentwicklung baut demnach auf dem bestehenden Modell auf und unterstützt die Evolution des SoS.

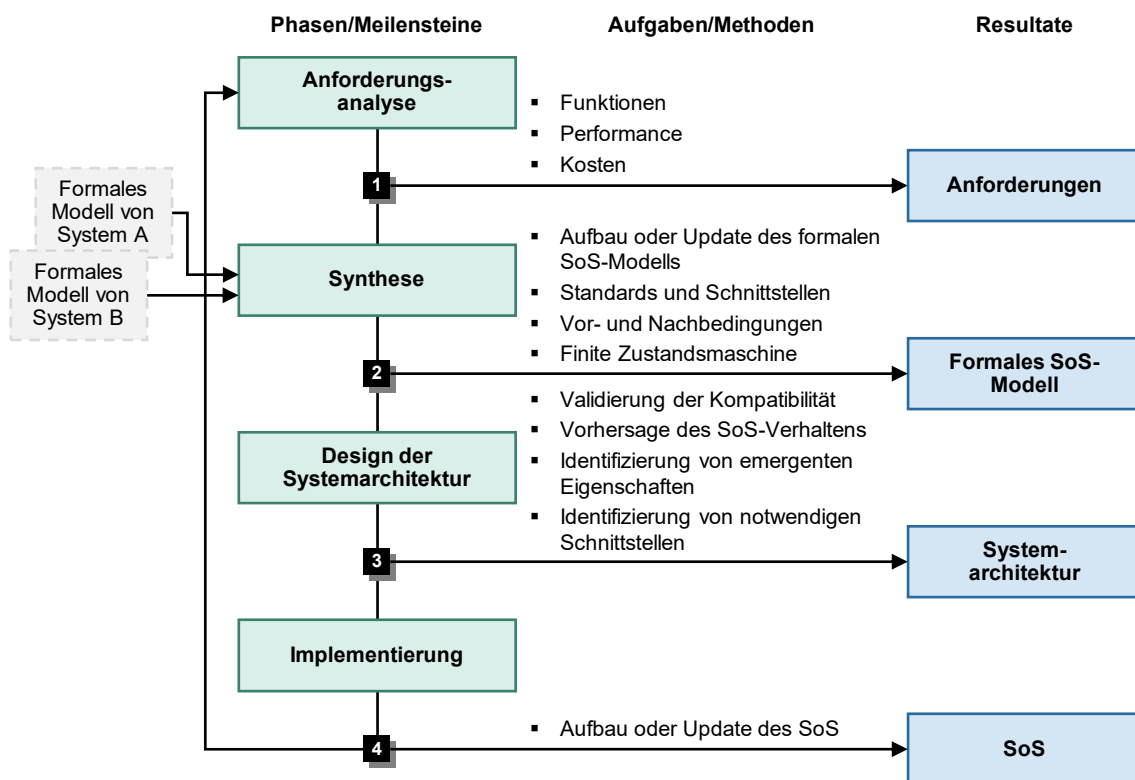


Bild 3-11: Rahmenwerk zur Synthese, Evaluation und gesteuerten Entwicklung von SoS nach SELBERG und AUSTIN [SA14, S. 1075]

Bewertung: Der Ansatz nach SELBERG und AUSTIN stützt sich auf einem Bottom-Up Vorgehen, welches bereits bestehende Strukturen und Systeme mit einbezieht. Ein wesentlicher Erfolgsfaktor sind Standards und Schnittstellen, die die Interaktion zwischen den bereits existierenden Systemen und die Integration in bereits bestehende Strukturen im Rahmen von Evolutionen ermöglichen. Die Unterscheidung von unterschiedlichen Kooperationsformen zwischen Unternehmen, die einen wesentlichen Einfluss auf die Bereitstellung der formalen Systemmodelle haben, wird durch das Rahmenwerk nicht unterstützt.

3.2.1.8 Prozessrahmenwerk für SoS nach ISO 21840

Die ISO 21840 ist eine Richtlinie für die Nutzung der ISO 15288 im Kontext von SoS. Als Standard des Systems Engineering beschreibt die ISO 15288 insgesamt 30 Prozesse aus vier Bereichen für eine erfolgreiche Entwicklung von abgegrenzten Systemen (vgl. Kapitel 2.3.1). Diese werden durch die ISO 21840 um SoS-spezifische Aspekte erweitert. Dabei wird berücksichtigt, dass im Gegensatz zur ISO 15288 kein klar abgegrenztes System entwickelt wird und mehrere Unternehmen in den Prozessen involviert sind. Insgesamt zeigen sich in allen Prozessen Anpassungsbedarfe, die in kurzen Texten erläutert werden.

Die Änderungen in den **Vertragsprozessen** lassen sich auf die Managementunabhängigkeit zurückführen. Um die Interaktion zwischen den eigenständigen Systemen im SoS zu regeln, sind Vereinbarungen notwendig. Diese können sich je nach Managementstruktur unterschiedlich gestalten. Insbesondere wenn es keine übergeordnete Instanz gibt, die Einfluss auf das SoS und die einzelnen Systeme hat, müssen Vereinbarungen getroffen werden, um die Interoperabilität zwischen den beteiligten Systemen zu klären. Bspw. kann eine Vereinbarung regeln, welcher Standard für die Interaktion genutzt wird. Dabei müssen die unterschiedlichen Ziele einzelner Unternehmen berücksichtigt werden, die durch Verträge zwischen den Beteiligten im SoS geregelt und ausgeglichen werden können.

Die **organisatorischen Unterstützungsprozesse** definieren die Rahmenbedingungen für die Projekte und legen fest, in welcher Reihenfolge diese bearbeitet werden. Diese Prozesse finden auch auf SoS-Ebene statt, sind jedoch stark von den Organisationsstrukturen abhängig. Ein Beispiel ist die Priorisierung von SoS-Projekten und in welcher Reihenfolge z. B. SoS-Updates entwickelt werden (Bild 3-12). Gibt es ein verantwortliches SoS-Team, welches für das Management des SoS zuständig ist, führt dieses die Prozesse durch. Gibt es kein zentrales SoS-Management, müssen Vereinbarungen getroffen werden, wie diese Prozesse gesteuert werden.

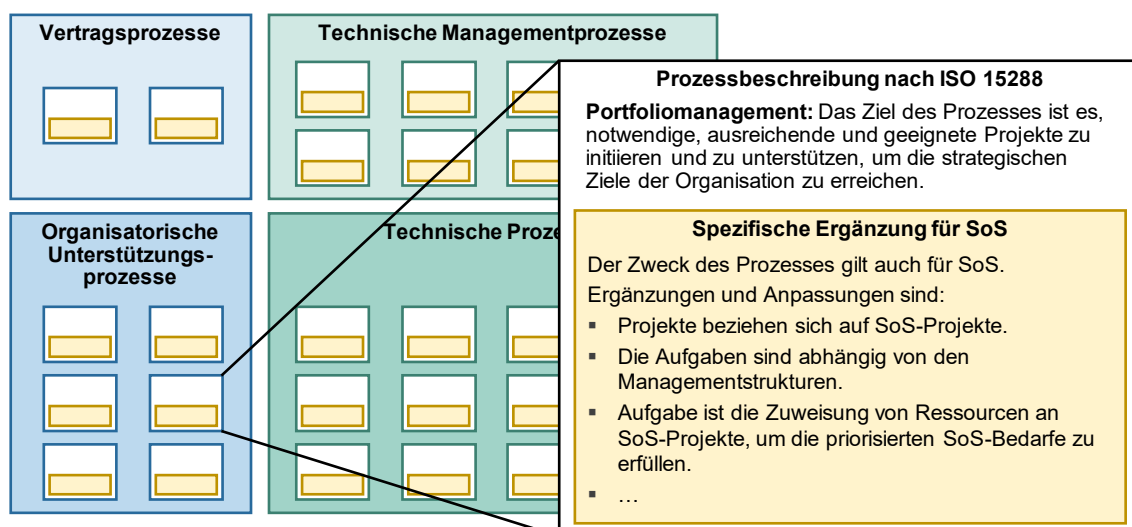


Bild 3-12: Prozessrahmenwerk nach ISO 21840 [ISO21840]

Technische Managementprozesse beziehen sich zum großen Teil auf die Erreichung der Zielvorgaben im Hinblick auf die Kosten, den Zeitrahmen und die zu erreichenden Ergebnisse. Diese finden auch auf SoS-Ebene Anwendung. Da die Verantwortlichkeit im SoS häufig nicht klar geregelt ist und die Kontrolle über die eigenständigen Systeme fehlt, können die Technischen Managementprozesse, wie z. B. das Entscheidungsmanagement, zum Teil nicht vollständig durchgeführt werden. Dies hängt jedoch stark von der Managementunabhängigkeit ab. In stark unabhängigen Beziehungen können die individuellen Ziele der Unternehmen dazu führen, dass die Entscheidung zwischen Alternativen unterschiedlich ausfallen. In Kooperationsbeziehungen sollten die Ziele übereinstimmen und somit auch die Entscheidungsprozesse auf SoS-Ebene vereinfachen.

Gleiches gilt für die **Technischen Prozesse**. Auch diese werden auf SoS-Ebene durchgeführt, um das SoS auf Basis von Stakeholderbedarfen zu entwickeln und in die Nutzung zu überführen. Die Prozesse können je nach Managementstruktur sowohl von verantwortlichen SoS-Teams durchgeführt werden als auch aus Sicht von Einzelunternehmen, die einen Teil des SoS gestalten.

Bewertung: Die Anpassungsbedarfe für SoS-Gestaltungsprojekte im Gegensatz zu Systemprojekten werden in der ISO 21840 deutlich. Insbesondere die Unterschiede durch die veränderten Managementstrukturen werden hervorgehoben. Die Beschreibungen weisen auf bestimmte Aspekte hin, die Unternehmen in diesem Zusammenhang zu beachten haben, gehen jedoch nicht konkret auf diese ein. Außerdem werden zahlreiche Hinweise gegeben, die im Zusammenhang mit den unterschiedlichen Ausprägungen von SoS stehen. Diese werden jedoch sporadisch und für den Anwender ohne erkennbare Struktur gegeben.

3.2.2 Ansätze aus verwandten Forschungsbereichen

Die Forschung im Bereich der SoS betrachtet bestimmte Aspekte und Herausforderungen, wie bspw. die Managementunabhängigkeit. Im Folgenden werden Ansätze aus verwandten Forschungsbereichen aufgegriffen, die sich in Teilen mit den Fragestellungen der SoS-Gestaltung auseinandersetzen. Zum einen wird die Gestaltung von vernetzten Systemen durch die Betrachtung von CPS, kollaborativen und software-intensiven Systemen aufgegriffen. Zum anderen werden Ansätze betrachtet, die sich mit dem dynamischen Verhalten und Emergenz auseinandersetzen. Dazu gehören Ansätze aus dem Bereich der Softwaretechnik zu Multiagentensystemen sowie Ansätze aus der Systemforschung zur Systemdynamik.

3.2.2.1 Planung von CPS nach WESTERMANN

CPS und SoS sind vernetzte Systeme mit ähnlichen Fragestellungen. CPS zeichnen sich durch eine hohe Integration von Daten und Diensten aus (vgl. Kapitel 2.2.2). WESTERMANN definiert in seiner Arbeit eine Reifegradmodell-basierte Planung von CPS. Diese

bezieht individuelle Ausgangssituationen auf Basis von Reifegraden mit ein und systematisiert anschließend die Schritte bis hin zur Umsetzungsplanung. Ein Vorgehensmodell beschreibt insgesamt sechs Phasen, um ausgehend von zuvor ermittelten Erfolgspotentialen hin zu einer Auswahl an Lösungskonzepten zu gelangen (Bild 3-13).

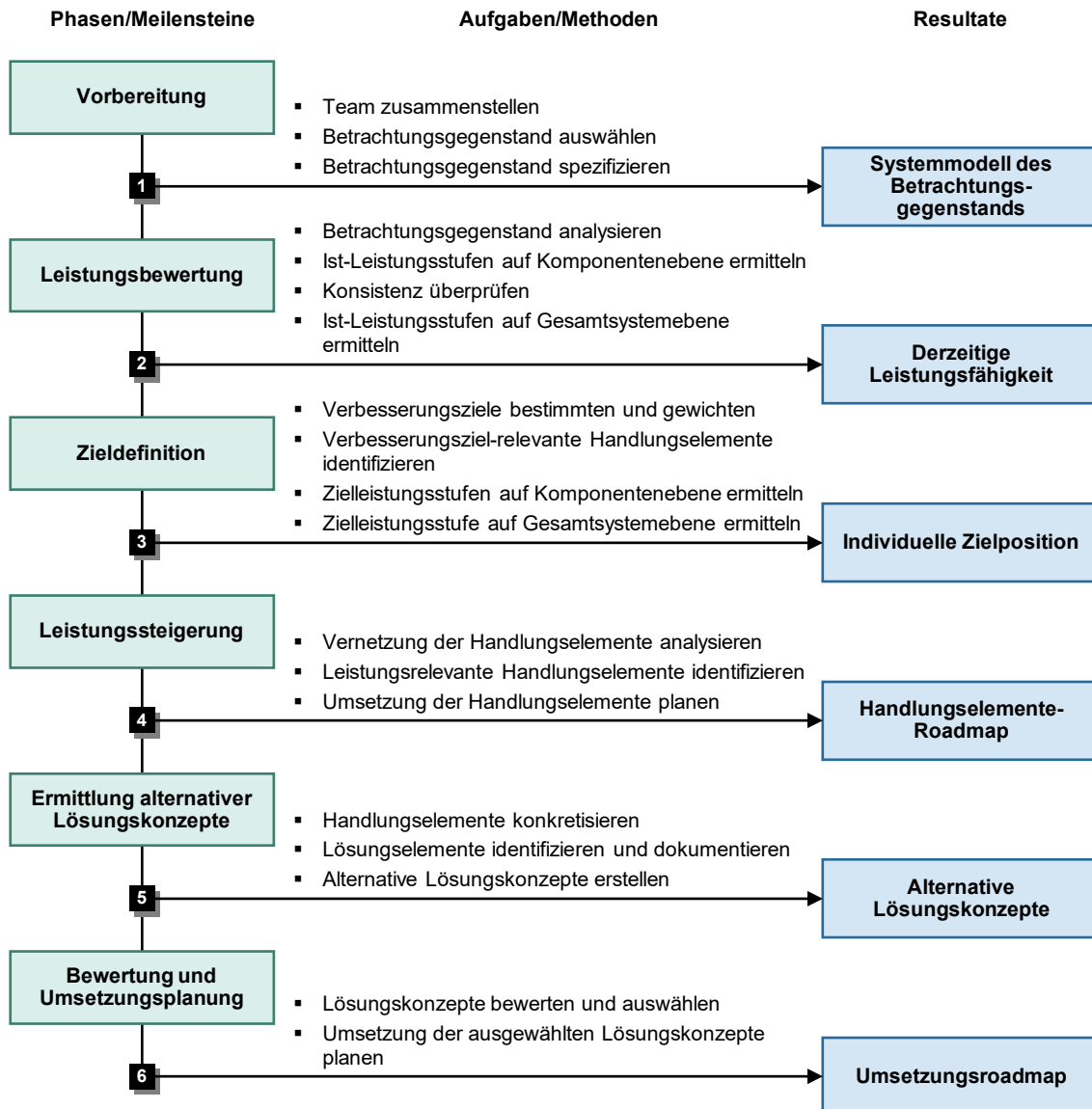


Bild 3-13: Vorgehensmodell zur Reifegradmodell-basierten Planung von CPS nach WESTERMANN [Wes17, S. 134]

In der **Vorbereitung** wird zunächst ein Team zusammengestellt und das System abgegrenzt, welches im weiteren Verlauf spezifiziert wird. Im Rahmen der **Leistungsbewertung** wird das abgegrenzte System analysiert und die Leistung bewertet. Die Analyse wird auch auf Gesamtproduktebene durchgeführt, um den Handlungsspielraum in Verbindung mit den weiteren Teilsystemen zu ermitteln. Dies ist die Ausgangsbasis für die **Zieldefinition**, die in Abhängigkeit der Leistungsfähigkeit und der Verbesserungsziele bestimmt wird. In der vierten Phase wird die **Leistungssteigerung** analysiert und geplant.

Hierzu wird der Vernetzungsgrad der Handlungselemente bestimmt und der Wirkungsgrad einer Änderung auf das Gesamtsystem bestimmt. Bei starker Vernetzung ist der Einfluss einer Änderung auf das Gesamtsystem groß, jedoch auch mit einem großen Aufwand verbunden. Im Anschluss werden **alternative Lösungskonzepte** mithilfe eines morphologischen Kastens ermittelt. Auf Basis der Exploration von Alternativen werden alternative Kombinationen für die Gesamtlösung erstellt. Im letzten Schritt findet die **Bewertung und Umsetzungsplanung** dieser Alternativen mithilfe von Bewertungsmethoden, wie die Nutzwertanalyse, statt [Wes17, S. 133ff.].

Bewertung: Die Umsetzungsplanung von CPS-Konzepten erfolgt anhand eines systematischen Vorgehens, welches durch zahlreiche Methoden und Hilfsmittel gestützt wird. Der Fokus liegt dabei auf Teilsystemen von CPS, die von einem Unternehmen verantwortet und beeinflusst werden können. Die Zusammenhänge auf Gesamtsystemebene werden durch eine Leistungsbewertung mit einbezogen. Die Umsetzung von Änderungen für unabhängige Teilsysteme wird dabei jedoch nicht explizit betrachtet.

3.2.2.2 Modellbasiertes Engineering von kollaborativen eingebetteten Systemen nach BÖHM et al.

BÖHM et al. stellen eine umfangreiche Systematik zur Gestaltung von kollaborativen eingebetteten Systemen vor. Die Systematik umfasst Beschreibungsmodelle (Meta-Modelle) und Methoden mit dem Ziel, dynamische Strukturen und Verhaltensweisen von kollaborativen Systemen abzubilden. Im Fokus stehen eingebettete Systeme (Embedded Systems), die als Einheit aus Software und Hardware über Sensoren und Aktoren mit einem Gesamtsystem (CPS) verbunden sind und Überwachungs-, Steuerungs- oder Regelaufgaben übernehmen. Diese Systeme können sich vernetzen und somit kollaborative Systemgruppen bilden, in denen ein gemeinsames Ziel verfolgt wird. Bild 3-14 gibt einen Überblick über das zugrundeliegende Konzept.

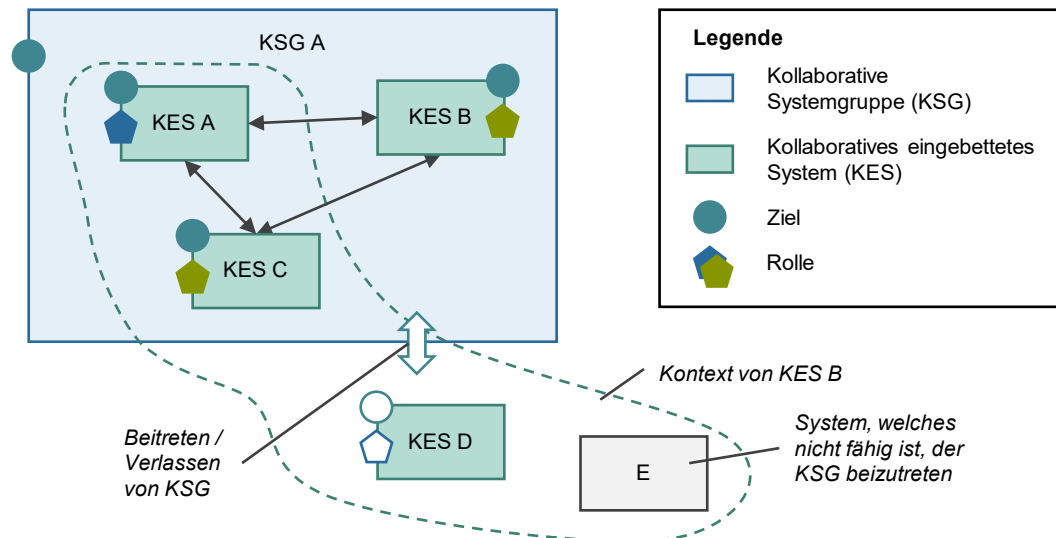


Bild 3-14: Kollaborative eingebettete Systeme und Systemgruppen nach BÖHM et al. [BBK+21, S. 26]

Zur Entwicklung solcher Systeme stellen BÖHM et al. eine Methode zur Entwicklung dynamisch vernetzter eingebetteter Systeme vor. Die Methode wird in Bild 3-15 dargestellt. Sie umfasst vier wesentliche Schritte, die jeweils über Designartefakte gekoppelt sind. Dabei wird zwischen der Ebene der kollaborativen eingebetteten Systeme (KES) und der kollaborativen Systemgruppe (KSG) unterschieden.

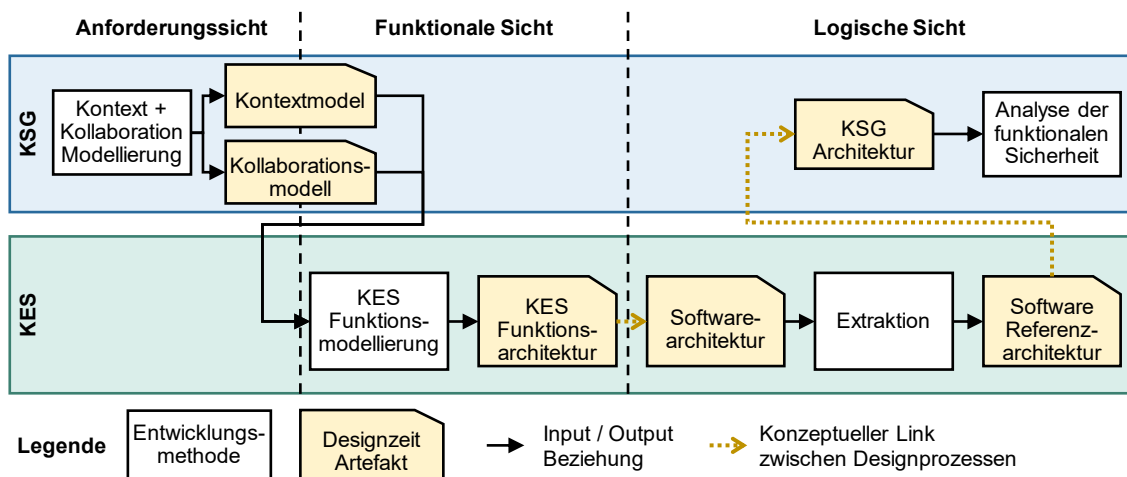


Bild 3-15: Methode zur Entwicklung kollaborativer eingebetteter Systeme nach BÖHM et al. [BBK+21, S. 97]

Die **Modellierung von Kontext und Kollaboration** bezieht neben der Struktur, den Schnittstellen und dem beobachtbaren Verhalten zusätzlich die Kommunikation zwischen den KES sowie deren Rolle und Ziele innerhalb des Systemnetzwerks mit ein (vgl. Bild 3-14). Die Modellierung von Rollen ermöglicht die Definition der notwendigen Funktionen, die innerhalb einer Kollaborationsgruppe erfüllt werden müssen. KES können diese Rollen im Kollaborationsfall einnehmen und müssen dabei die definierten

Funktionen der Rolle erfüllen. Dies wird zusätzlich durch einen vertragsbasierten Ansatz sichergestellt. Formalisierte Verträge beschreiben die notwendigen Voraussetzungen an der Schnittstelle zur Interaktion in Systemgruppen.

Im zweiten Schritt erfolgt die **KES Funktionsmodellierung**. Für jedes KES wird eine funktionale Architektur beschrieben, welche die Funktionen und deren Zusammenhänge innerhalb des Systems umfasst. Zusätzlich werden unterschiedlichen Kollaborationsfälle mit einbezogen. Je nachdem welche Rolle ein KES in der KSG einnimmt, erfüllt es unterschiedliche Funktionen. Daher wird im Rahmen der Funktionsmodellierung bspw. zwischen aktuellen und potenziellen Funktionen unterschieden, wobei sich der Status in Abhängigkeit der Zeit und der Ziele verändern kann.

Die **Extraktion** der Architektur erfolgt auf Basis von existierenden Systemarchitekturen. Dabei werden wiederkehrende Elemente genutzt und in eine Referenzarchitektur überführt. Im Vordergrund steht die Software-Architektur, die automatisiert anhand etablierter Methoden der Softwaretechnik abgeleitet wird. Ein Beispiel ist die statische Konnektivität-Matrix-Analyse auf Basis von Matlab-Modellen.

Als letzter Schritt wird die **Analyse der funktionalen Sicherheit** aufgeführt. Da der Fokus der Systematik auf sicherheitskritischen eingebetteten Systemen liegt, wird dieser Schritt umso relevanter. Für die Kollaboration in Systemgruppen besteht die Herausforderung, dass bestimmte Tests nur im Zusammenhang mit den anderen Systemen durchgeführt werden können. Daher werden gesonderte Methoden vorgestellt, wie bspw. die Injektion von Kommunikationsfehlern in die Systemgruppe, um das Verhalten zu testen [BBK+21].

Bewertung: Der Ansatz ist umfassend und berücksichtigt mehrere Aspekte. Zum einen wird zwischen der Systemebene und der Systemgruppenebene unterschieden, in denen unterschiedliche Methoden angewendet werden. Darüber hinaus ermöglicht die Modellierung von Rollen, die Dynamik von kollaborativen Systemen abzubilden. Sie spiegeln die Struktur und das Verhalten von Systemgruppen wider und bieten so eine Möglichkeit, diese zu analysieren und zu gestalten. Der Schwerpunkt liegt auf der Gestaltung von Software, wodurch die Zusammenhänge zu weiteren Domänen nicht explizit betrachtet wurden. Ebenso wurde keine Unterscheidung verschiedener Arten von Kollaborationsgruppen vorgenommen und die Organisationsperspektive nicht mit einbezogen.

3.2.2.3 Multiagenten-System Engineering Methode nach WOOD und DELOACH

Agentenbasierte Ansätze haben ihren Ursprung in der Softwaretechnik. Das Ziel besteht darin, unabhängige Systeme für den Einsatz in Systemnetzwerken zu realisieren. Ein Beispiel ist der Ansatz nach WOOD und DELOACH. Dieser basiert auf einer Analyse- und einer Designphase und gliedert sich in sieben Schritte, die iterativ durchlaufen werden (Bild 3-16).

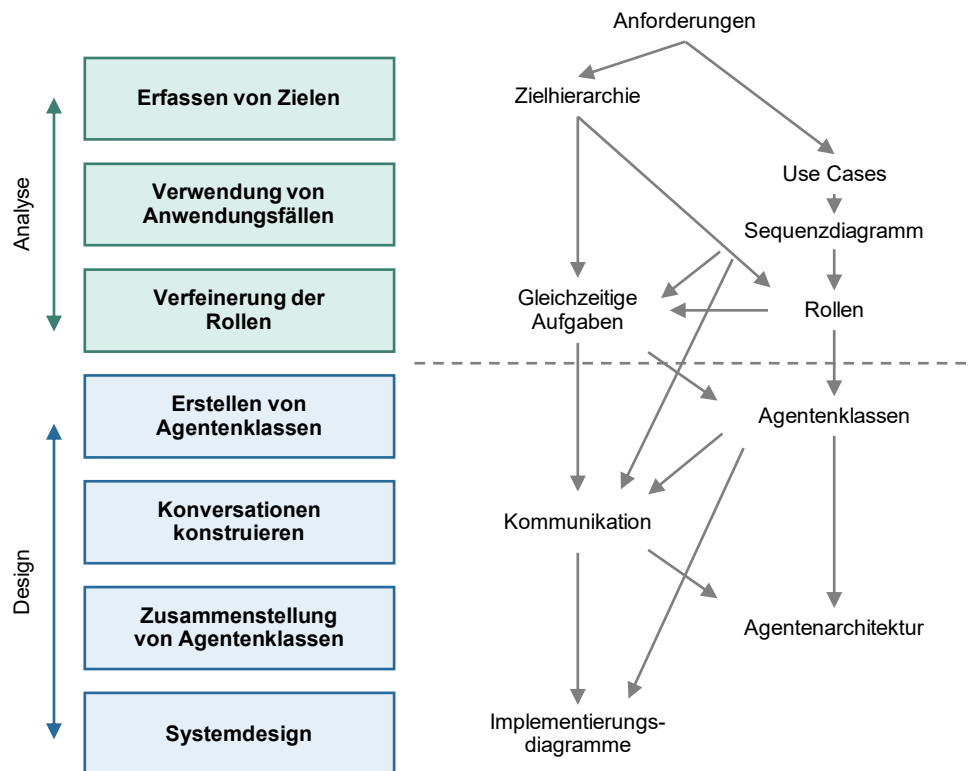


Bild 3-16: Vorgehen zur Entwicklung von Multiagentensysteme nach WOOD und DELOACH [WD01, S. 209]

Im Rahmen der Analyse werden auf Basis von Anforderungen die Ziele in Form einer Zielhierarchie erfasst. Zusätzlich werden Anwendungsfälle und Sequenzdiagramme erstellt, um im nächsten Schritt Rollen abzuleiten. Diese sind die Ausgangsbasis zur Auslegung der Agenten. Klassendiagramme legen sowohl die Agenten als auch deren Kommunikation untereinander fest. Im letzten Schritt werden die einzelnen Agenten entwickelt [WD01].

Bewertung: Die Methode zur Gestaltung von Multiagentensystemen nach WOOD und DELOACH ist ein systematischer Ansatz zur Gestaltung von Software-Agenten innerhalb von Multiagentensystemen. Ein wichtiges Element ist dabei die Beschreibung von Rollen. Diese stützen sich auf einer Ziel- und Anwendungsfallanalyse und beschreiben lösungsneutral, welche Aufgaben innerhalb eines Netzwerks erfüllt werden müssen. Die Nutzung von Rollen als Beschreibungselement innerhalb von flexiblen Netzwerken stellt einen vielversprechenden Ansatz dar und wird auch im Rahmen von SoS-Ansätzen diskutiert.

3.2.2.4 Ansätze der Systemdynamik nach LANE

Innerhalb der Systemwissenschaft wird die Analyse von dynamischen Verhalten mit dem Gebiet der Systemdynamik verbunden. LANE stellt in seiner Arbeit zwei etablierte An-

sätze der Systemdynamik vor. Das Ziel dieser Ansätze ist das zeitlich veränderbare Verhalten in Verbindung mit Systemstrukturen zu analysieren. Dazu wird zu Beginn eine Problemstellung definiert und im Anschluss ein Konzept mit Annahmen entwickelt. Anschließend kann ein Kausalschleifendiagramm (causal loop diagram) oder ein Bestands-/Flussdiagramm (stock/flow diagram) genutzt werden, um die Annahmen zu modellieren (Bild 3-17). Beide Repräsentationen helfen dabei, die Zusammenhänge zwischen variablen Größen zu verstehen und durch die explizite Darstellung die Kommunikation innerhalb von Teams zu fördern [Lan08].

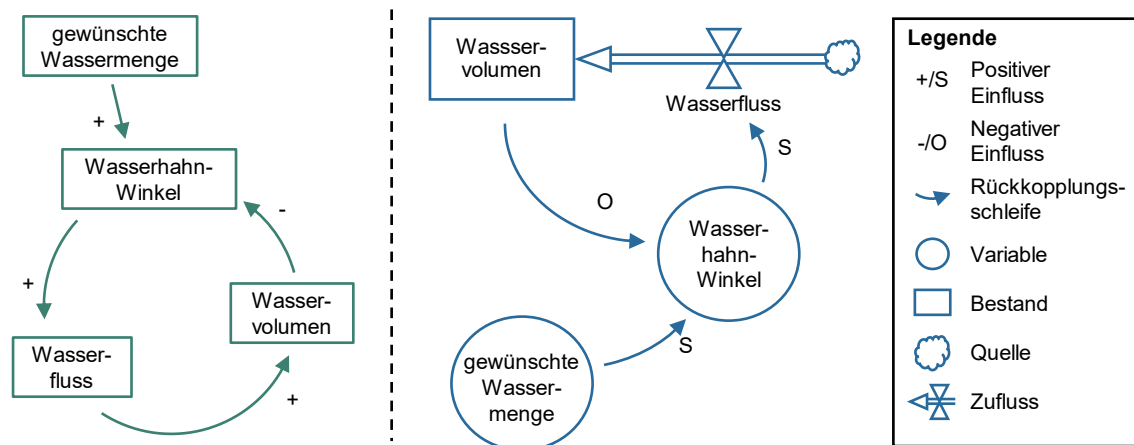


Bild 3-17: Ansätze der Systemdynamik nach LANE, Kausalschleifendiagramm (links) und Bestands-/Flussdiagramm (rechts) [Lan08]

Bewertung: Die beschriebenen Ansätze sind ein leichtgewichtiges Mittel, um Zusammenhänge innerhalb von Systemen zu modellieren und dabei das Verständnis über Systemzustände und deren Veränderung zu erhöhen. Im Rahmen von SoS können diese Ansätze genutzt werden, um bspw. Veränderungen zu analysieren, die sich aus dem Verlassen oder Hinzutreten von Systemen in Systemverbünde ergeben. Je größer die Unwägbarkeit in dem SoS, desto mehr Annahmen müssen im Rahmen der Systemdynamikanalyse getroffen werden.

3.3 Wissensbasierte Ansätze zur Unterstützung der SoS-Gestaltung

Im Folgenden werden Ansätze zur Unterstützung der SoS-Gestaltung vorgestellt, die auf einer Form von explizitem Wissen aufbauen. Dazu wird zum einen in Kapitel 3.3.1 ein Überblick über Wissensansätze mit Fokus auf der SoS-Gestaltung gegeben. Diese umfassen sowohl musterbasierte Ansätze als auch Designprinzipien, die als Handlungsanweisungen in Gestaltungsprozessen unterstützen. Darüber hinaus werden in Kapitel 3.3.2 methodische Ansätze zur Nutzung von Wissen in Entwicklungsprozessen betrachtet.

3.3.1 Wissen in der SoS-Gestaltung

3.3.1.1 Modellierungsmuster für SoS-Architekturen nach INGRAM et al.

Im Rahmen des EU-Projekts COMPASS wurden Lösungsmuster im Kontext von SoS entwickelt. In dem Beitrag von INGRAM et al. werden insgesamt fünf Lösungsmuster vorgestellt, die im Wesentlichen aus dem Bereich der Softwaretechnik und Datenmodellierung abgeleitet wurden [IPP+14].

Die Muster wurden auf SoS übertragen und einheitlich strukturiert. Sie setzen sich aus insgesamt fünf Teilbereichen zusammen. Als erstes werden der **Hintergrund** und die allgemeine Grundlage des jeweiligen Musters zusammengefasst. Im zweiten Bereich werden die **Ziele** beschrieben, die durch das Muster erreicht werden können. Dies kann im Beispiel des Service-orientierten Architekturmusters die Unterstützung der Langlebigkeit von SoS sein. Der dritte Teilbereich beschreibt die grundlegende **Struktur** der Lösungsmuster. Dabei wird auf die notwendigen Bestandteile des SoS eingegangen. Der Grundgedanke und die **Begründung**, die dem Muster zugrunde liegt, werden im vierten Teilbereich beschrieben. Dabei wird u. a. auf den Zusammenhang und die Besonderheiten in Bezug auf die Gestaltung eingegangen. Der fünfte Teilbereich umfasst ein **Beispiel**, in dem sich das Muster ausprägt. Dies soll das Verständnis für das jeweilige Muster erhöhen.

Ein Beispiel für ein Architekturmuster ist das *zentralisierte Architekturmuster* (Bild 3-18). Dieses beschreibt ein SoS, in dem ein zentraler Knotenpunkt das Verhalten des SoS steuert.

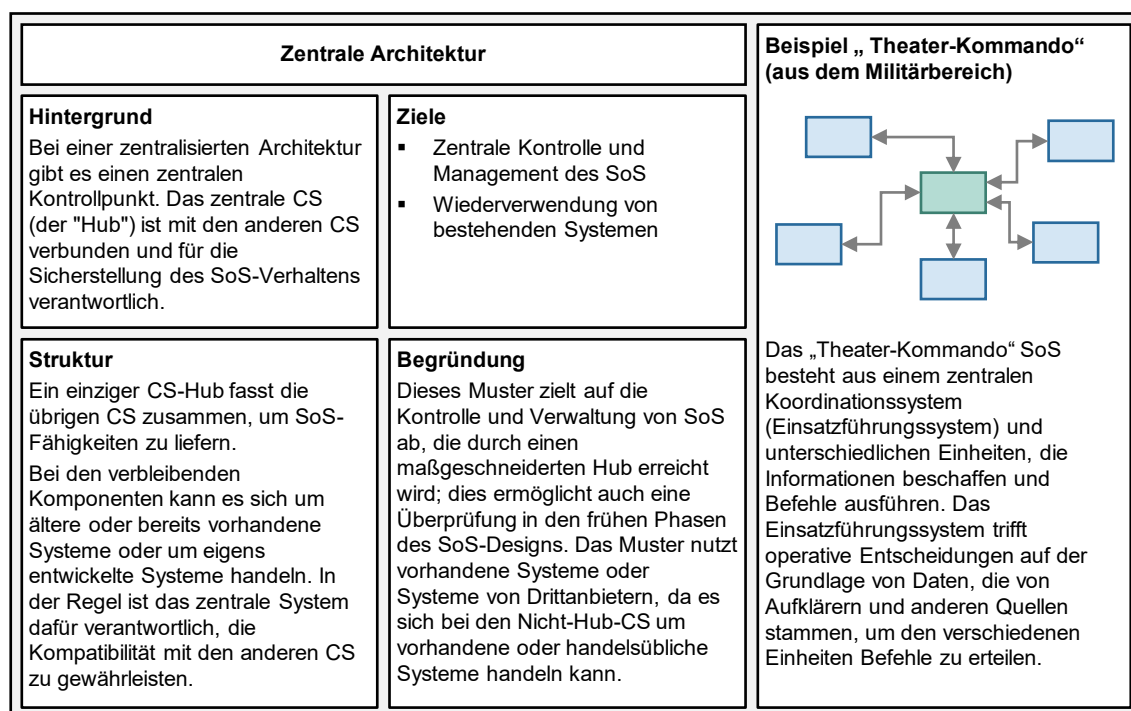


Bild 3-18: Zentralisiertes Architekturmuster nach INGRAM et al. [IPP+14, S. 148f.]

Neben dem zentralisierten Architekturmuster werden vier weitere Muster aufgeführt. Zum einen wird die *Service-orientierte Architektur* beschrieben. Dieses Muster charakterisiert eine Grundstruktur aus Service-Anbietern (im Fall von SoS eigenständigen Systemen), die bestimmte Funktionalitäten anbieten und innerhalb einer Anwendung genutzt werden können. Des Weiteren werden die Muster *Publish-Subscribe*, *Pipes and Filter* und *Blackboard Architektur* vorgestellt. Diese unterscheiden sich sowohl in den Arten des Datentransfers als auch in den jeweiligen Fähigkeiten der beteiligten Systeme zum Kontrollieren und Beeinflussen von Daten innerhalb der Interaktion.

Bewertung: Die Architekturmuster nach INGRAM et al. adressieren die Gestaltung von SoS. Sie gehen auf die Herausforderungen von SoS ein und beschreiben typische Architekturmuster, die mögliche Grundstrukturen für eine Interaktion von mehreren Systemen beschreiben. Im Rahmen der Zielbeschreibung wird teilweise auf die charakteristischen Merkmale von SoS eingegangen, sodass der Mehrwert für SoS deutlich wird. Es wird jedoch ein starker Bezug zur Softwaretechnik deutlich. Am Beispiel der Service-orientierten Architektur wird deutlich, dass die Übertragung auf multidisziplinäre SoS nicht vollständig ist und lediglich auf den Datenaustausch eingegangen wird. Außerdem wird die Sprache der Softwaretechnik genutzt. Dies wirkt sich negativ auf die Verständlichkeit in interdisziplinären Teams der SoS-Gestaltung aus.

3.3.1.2 Rekonfigurationsmuster für SoS nach PETITDEMANGE et al.

Die Rekonfiguration von SoS stellt eine wesentliche Herausforderung von SoS dar. Zur Laufzeit muss ein SoS die Vernetzung und Entkopplung von Systemen ermöglichen und dabei die Grundfunktionalität aufrechterhalten. Um dies zu ermöglichen, beschreiben PETITDEMANGE et al. den Einsatz von Rekonfigurationsmustern, die zur Laufzeit durch ein SoS instanziiert werden können. Ein Rekonfigurationsmuster beschreibt die grundsätzliche Verhaltensweise eines SoS wenn Veränderungen eintreten. Beispiele sind Ein- und Austritte von eigenständigen Systemen in ein SoS. Ein Rekonfigurationsmuster wird anhand von sechs Merkmalen beschrieben: (1) Namen, (2) Ziel, (3) Kontext, (4) Problem, (5) Lösung und (6) Konsequenzen, die durch Anwendung des Musters zu erwarten sind [PBB16].

Ein Beispiel ist die *Co-Evolution* von Konfigurationen. Dieses Muster beschreibt eine Möglichkeit, wie eine Rekonfiguration von einer alten in eine neue Konfiguration durch die parallele Aufrechterhaltung des Informationsgehalts realisiert werden kann. Hierzu werden die beiden Konfigurationen auf einem alten und neuen System durch ein Skript synchronisiert und die einzelnen Entitäten (eigenständige Systeme) nach und nach migriert. Dabei werden beide Konfigurationen aufrechterhalten, bis die vollständige Rekonfiguration erfolgt ist. Das Muster ist insbesondere für SoS mit direkten Managementstrukturen geeignet, da es einen zentralen Operator geben muss, der während der Migration als Verbindungsglied zwischen den Systemen agiert und diese beaufsichtigt.

Bewertung: Die Rekonfigurationsmuster nach PETITDEMANGE et al. gehen auf eine konkrete Herausforderung von SoS ein und sind als Hilfestellung für Entwicklerinnen und Entwickler geeignet, um das Problem der Rekonfiguration durch Muster zu unterstützen. Die Muster folgen einer einheitlichen Struktur und sind somit gut verständlich. Ein Zusammenhang zu weiteren Aspekten und Herausforderungen von SoS sowie eine Auflistung weiterer Beispiele für Rekonfigurationsmuster fehlt jedoch, sodass eine breite Anwendung im Rahmen der SoS-Gestaltung nicht möglich ist.

3.3.1.3 Interoperabilitätsmuster nach BRÖRING et al.

In der viel zitierten Arbeit von BRÖRING et al. werden fünf *Interoperabilitätsmuster* aufgeführt, die in einem offenen Marktplatz für das Zusammenspiel zwischen IoT-Plattformen und IoT-Apps unterstützt werden sollten. Eine IoT-Plattform kann beispielsweise eine Smart Watch sein, die Gesundheitsdaten sammelt und diese an unterschiedliche Anwendungen (Apps) weitergeben kann. Die fünf Muster stellen die wesentlichen Interaktionsmechanismen dar, wie IoT-Plattformen und die entsprechenden Anwendungen interagieren müssen, um einen übergreifenden Mehrwert zu bieten, der nicht durch einzelne Verbindungen realisiert werden kann. Die Vision ist dabei eine dynamische Suche und Identifikation von Zusammenhängen in Anwendungen über einzelne Plattformen und Domänen hinaus. So kann bspw. eine Verknüpfung von Informationen zwischen der Smart Home Domäne und der Smart City Domäne viele Vorteile bieten [BSS+17].

Um dies zu erreichen, sind neben einer definierten Schnittstelle (API), die von allen Teilnehmern unterstützt werden muss, auch unterschiedlichen Arten der Interoperabilität notwendig. Dazu gehören Eins-zu-Eins-Beziehungen (Muster 3) sowie Vielfach-Beziehungen über Domänen hinweg. Die Muster werden textuell beschrieben und durch grafische Modelle visualisiert (Bild 3-19).

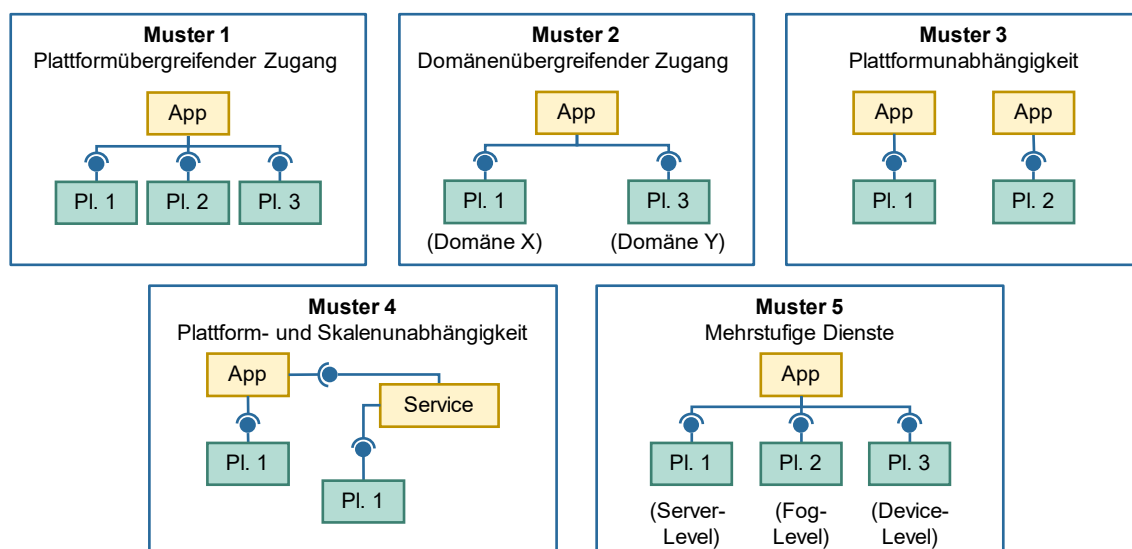


Bild 3-19: Übersicht der Interoperabilitätsmuster nach BRÖRING et al. [BSS+17]

Bewertung: Auch wenn die von BRÖRING et al. beschriebenen Lösungsmuster IoT-Systeme fokussieren, treffen sie im Wesentlichen den Bedarf von SoS sich dynamisch vernetzen zu können. Neben Eins-zu-Eins-Beziehungen werden auch Vielfachbeziehungen beschrieben, die domänenübergreifend ausgelegt sind. Die textuellen Beschreibungen werden durch Grafiken ergänzt, um das Verständnis zu erhöhen. Das Gesamtverständnis wird jedoch erschwert, da die Beschreibungen keiner einheitlichen Struktur folgen, wie bspw. bei INGRAM et al. Außerdem werden viele zusätzliche Informationen in dem Beitrag gegeben, die nicht im Rahmen der Muster aufgeführt werden. Hierzu gehören z. B. Hinweise zur notwendigen Herstellung einer syntaktischen und semantischen Interoperabilität als Grundvoraussetzung der Interoperabilitätsmuster. Dies wird u. a. durch MEENTEMEYER et al. aufgegriffen, indem sie notwendige Gestaltungsmaßnahmen der Interoperabilität als Designprinzip beschreiben (vgl. Kapitel 2.4.3).

3.3.1.4 Muster für vertragsbasierte SoS nach FALDIK et al.

Die operative Unabhängigkeit und Managementunabhängigkeit von Systemen im SoS führt zu einer Unverbindlichkeit zwischen Systemen. Die Interaktionen sind freiwillig und finden zwischen unterschiedlichsten Systemen statt. Um dennoch eine Verbindlichkeit herzustellen und eine zuverlässige Interaktion zu ermöglichen, beschreiben FALDIK et al. ein *Vertragsmuster*. Dieses schränkt die möglichen Zustände für eine Interaktion ein und stellt Vor- und Nachbedingungen. So werden zuverlässige Interaktionen garantiert, die auf verbindlichen Zusagen der Vertragspartner basieren [FPF+17].

Das Muster setzt sich aus fünf Aspekten zusammen, die beschrieben werden müssen:

- **Definition eines vertragsbasierten SoS:** Hier werden sämtliche Verträge aufgeführt, die im SoS definiert werden.
- **Vertragskonformität:** Hier werden sämtliche Systeme aufgeführt, die im Rahmen des SoS die Verträge erfüllen können.
- **Vertragsverbindungen:** Zeigt die Verbindungen und Schnittstellen zwischen Verträgen innerhalb des SoS auf.
- **Vertragsdefinition:** Definiert die Operationen und Zustände für einen einzelnen Vertrag.
- **Vertragsprotokoll:** Definiert die Reihenfolge der Nachrichten zwischen Systemen und ruft die Operationen der Systeme auf.

Bewertung: Das Vertragsmuster nach FALDIK et al. beschreibt einen Ansatz, wie der Austausch zwischen Systemen im SoS geregelt werden kann. Das Muster formalisiert die Beschreibung von Verträgen im Rahmen der SoS-Modellierung. Der Ansatz zeigt das vielfältige Verständnis von Mustern und die unterschiedlichen Einsatzzwecke. Das von FALDIK et al. beschriebene Muster unterstützt die Modellierung und ist weniger für das

Gesamtverständnis von SoS geeignet, auch wenn der Grundgedanke von Verträgen zur Unterstützung der Interaktion innerhalb von SoS eine hohe Relevanz hat.

3.3.1.5 Designprinzipien für resiliente SoS nach UDAY und MARAIS

Resilienz beschreibt die Fähigkeit eines Systems auf Störungen zu reagieren und sich davon zu erholen. Dynamische SoS sind kontinuierlich Veränderungen ausgesetzt, sodass ein resilientes Design besonders wichtig ist. UDAY und MARAIS stellen in ihrer Arbeit Handlungsrichtlinien in Form von Designprinzipien vor, die die Resilienz von SoS erhöhen (Tabelle 3-1). Ein Beispiel ist die *physische Redundanz*, die durch die Verfügbarkeit von Systemen mit gleicher Funktion erreicht wird. Im Fall eines Transportnetzwerkes können bspw. zusätzliche Busse vorgehalten werden, um auf äußere Veränderungen, wie ein zusätzliches Personenaufkommen, zu reagieren. Die Prinzipien werden durch textuelle Beschreibungen erläutert und durch Beispiele verdeutlicht [UM15].

Tabelle 3-1: Designprinzipien für resiliente SoS nach UDAY und MARAIS [UM15, S. 499ff.]

Prinzip	Erklärung
Physische Redundanz	Einsatz von Backups (redundante Hardware) mit gleichen Funktionen, für den Fall, wenn Primärsysteme im SoS ausfallen
Funktionale Redundanz	Schaffung von Redundanz ohne zusätzliche Systeme, indem Funktionalitäten einzelner Systeme erweitert werden
Eigenschaften auf Systemebene verbessern	Verbesserung der Flexibilität, Robustheit und Anpassungsfähigkeit einzelner Systeme
Reparierbarkeit	Sicherstellung der Verfügbarkeit von Systemen durch Verkürzung der Wiederherstellungszeit
Übergreifende Systeminteraktion	Verbesserung der Kommunikationsmöglichkeiten zwischen Systemen im SoS indem jedes System mit jedem System interagieren kann
Lokalisierte Kapazität	Lokaler Ausfall sollte nicht zu einem globalen Ausfall führen
Einsatz von Menschen	Menschen können den Ausfall durch kreative Lösungen entgegenwirken
Präventive Korrektur	Einleitung von Resilienzmaßnahmen vor der Störung
Verbesserte Kommunikation auf Organisationsebene	Erleichterung des Austauschs von Echtzeitinformationen zwischen Interessensgruppen und Betreibern zur Minimierung des Missmanagements im Anschluss einer Störung
Kombination von Resilienzprinzipien	Kombination der zuvor genannten Prinzipien, um Störungen zu verhindern

Bewertung: Die Designprinzipien geben Erfahrungswissen wieder, die Entwickler bei der Gestaltung beachten können, um resiliente SoS zu realisieren. Im Gegensatz zu Lösungsmustern wird weniger auf die Grundstrukturvoraussetzungen des SoS eingegangen,

sondern vielmehr auf die unterschiedlichen Maßnahmen, die in der Gestaltung berücksichtigt werden können. Diese werden textuell in Form von Prinzipien beschrieben, wodurch eine breite Anwendbarkeit gegeben ist. Auch wenn die Prinzipien wenig konkret sind, stellen sie wertvolle Ideen für Herausforderungen bei der Gestaltung bereit.

3.3.1.6 Naturbasierte Designprinzipien für SoS nach AZANI

AZANI differenziert in seiner Arbeit zwei Konzepte, die SoS und vor allem deren Gestaltung beeinflussen. Zum einen verfolgt das *geschlossene Konzept* proprietäre Lösungen in denen Standards und Spezifikationen unter Verschluss gehalten werden oder nur gegen Gebühren verfügbar sind. Demgegenüber wird im *offenen Konzept* auf etablierte Standards gesetzt, in denen Systeme dynamisch interagieren und austauschbar sind. Insbesondere für das Konzept der offenen SoS stellt AZANI Designprinzipien zur Verfügung, die auf die Herausforderungen solcher SoS eingehen (Tabelle 3-2). Diese basieren auf etablierten Mechanismen der Natur, die sich über Jahrhunderte der Evolution als Erfolgsfaktoren gezeigt haben. Ein Beispiel ist die Etablierung von austauschbaren funktionalen Einheiten durch das Prinzip der *Modularität* [Aza08].

Tabelle 3-2: Naturbasierte Designprinzipien nach AZANI [Aza08, S. 2ff.]

Prinzip	Erklärung
Selbstregulierung	In einem offenen System sollten Elemente sich selbst regulieren können, sowohl proaktiv als auch reaktiv.
Emergenz	Emergenz ist das Resultat der Interaktion von mehreren Elementen. Diese kann nicht durch die Betrachtung einzelner Elemente verstanden werden.
Erhaltung	Ein SoS sollte ganzheitlich gedacht werden und wenig Verluste erzeugen. Der Output eines Systems sollte durch den Input eines anderen verwertet werden, um Ressourcen zu schonen.
Rekonfiguration	Je nach Rahmenbedingung sollte ein SoS sich anpassen können, um flexibel unterschiedliche Szenarien zu unterstützen.
Symbiose	Erfolgreiche SoS können durch integrierte Produkt- und Prozessteams erreicht werden, in denen die wichtigsten Stakeholder vertreten sind. Voraussetzungen sind sich ergebende Vorteile für die jeweiligen Seiten.
Modularität	Modularität wird durch autonome Systeme erreicht, die über standardisierte Schnittstellen interagieren. Module können einzeln verwaltet werden und Komplexität, Evolution und Wartbarkeit unterstützen.

Bewertung: Die vorgestellten Designprinzipien basieren auf etablierten Mechanismen der Natur. Es zeigt sich, das Gestaltungswissen für SoS noch nicht ausreichend zur Verfügung steht und AZANI daher auf Analogien aus der Natur setzt. Diese geben Aufschluss über bestimmte Eigenschaften von SoS, die durch Designprinzipien unterstützt werden sollten.

3.3.1.7 Evolvabilitätsbezogene Prinzipien für SoS nach RICCI et al.

RICCI et al. beschreiben insgesamt zwölf Prinzipien, die die Weiterentwicklung von SoS vor dem Hintergrund der Evolution unterstützen. Ähnlich wie im Ansatz nach AZANI et al. werden diese textuell beschrieben und dem Anwender zur Verfügung gestellt.

Tabelle 3-3: Evolvabilitätsbezogene Prinzipien nach RICCI et al. [RRR14, S. 317]

Prinzip	Erklärung
Nutzung von Vorfahren	Verwendung von Designentscheidungen aus früheren Generationen
Disruptive architektonische Neugestaltung	Umgestaltung wesentlicher Teile des Systems zur gleichen Zeit anstelle von vielen kleinen Änderungen
Nachahmung	Nachahmung erfolgreicher Entwurfsentscheidungen aus anderen Systemen mit ähnlichem Zweck
Sinnvolle Adaptierung	Wiederverwendung von Entwurfsentscheidungen oder anderen Systemen, um Fähigkeiten bereitzustellen, für die sie ursprünglich nicht ausgewählt wurden
Dezentralisierung	Verteilung von Fähigkeiten auf mehrere geeignete Standorte
Gezielte Modularität	Isolierung von Teilen des Systems zur Verringerung der gegenseitigen Abhängigkeiten
Integrabilität	Schnittstellen für Kompatibilität und Gemeinsamkeiten auslegen, um eine effiziente Integration zu ermöglichen
Rekonfigurierbarkeit	Schaffung von Ähnlichkeiten, um die Wiederverwendung oder Neuzuweisung zu erleichtern
Redundanz	Duplizierung von Fähigkeiten oder Abläufen, um ihre zukünftige Umverteilung ohne Beeinträchtigung zu ermöglichen
Skalierbarkeit	Entwurfsentscheidungen treffen, die eine Skalierung von Ressourcen für zukünftige Unwägbarkeiten oder Bedürfnisse ermöglichen
Freie Kapazität	Überkapazitäten einplanen, um zukünftige Bedürfnisse zu erfüllen
Freiraum	Über- oder Unterspezifikation von Ressourcen, um Überkapazitäten zu bewältigen

Neben den Designprinzipien bieten RICCI et al. eine Grundlage zur Nutzung von Prinzipien für Gestaltungsoptionen. Demnach gelten die Designprinzipien als Inspiration für Gestaltungsoptionen, die durch Wegbereiter (Systeme) und Änderungsmechanismen (Methoden) umgesetzt werden können (Bild 3-20) [RRR14].

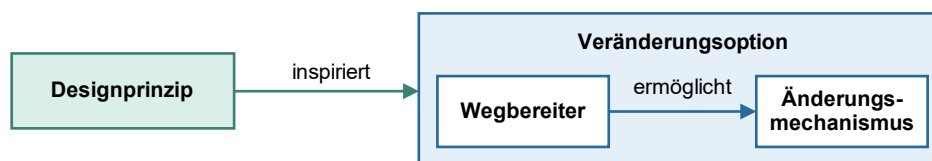


Bild 3-20: Zusammenhang zwischen Designprinzipien und Veränderungsoptionen nach RICCI et al. [RRR14, S. 318]

Bewertung: Die Designprinzipien geben wertvolle Hinweise, die im Rahmen der Gestaltung von SoS beachtet werden können, um die Evolution von SoS zu unterstützen. Zusätzlich wird die Grundidee der Nutzung von Designprinzipien durch eine einfache und leichtverständliche Logik zum Ausdruck gebracht. Diese unterstützt die Anwendung der Designprinzipien und ist auf sämtliche Designprinzipien übertragbar.

3.3.2 Wissensbasierte Ansätze der Produktentwicklung

3.3.2.1 Musterkatalog nach GAMMA et al.

Es existieren eine Vielzahl an Muster, die sich in der Granularität und Abstraktion voneinander unterscheiden. Häufig haben diese innerhalb eines Anwendungsbereichs Gemeinsamkeiten oder stehen in Beziehungen zueinander. Aus diesem Grund beschreiben GAMMA et al. eine Sammlung von 23 Softwaremustern, die in Form eines Katalogs strukturiert werden. Dies ermöglicht es, die Lösungsmuster, die sich in unterschiedlichen Dimensionen voneinander unterscheiden können, zu organisieren und die Nutzbarkeit zu erhöhen. Beispielsweise können Lösungsmuster leichter identifiziert oder ähnliche Muster innerhalb einer Kategorie referenziert und verglichen werden. Auch kann eine Kategorie Hinweise geben, die zur Identifizierung neuer Lösungsmuster führen [GHJ+93].

Der Katalog nach GAMMA et al. wird anhand der Kriterien *Charakterisierung* und *Zuständigkeit* strukturiert. Die Kriterien haben jeweils drei Ausprägungen, die zu insgesamt neun unterschiedlichen Kategorien führen. In diese werden die unterschiedlichen Lösungsmuster aus dem Themenbereich der objektorientierten Softwareentwicklung eingeordnet. Tabelle 3-4 gibt einen Überblick über die Kategorien sowie jeweils ein Lösungsmusterbeispiel. Der initiale Katalog bietet die Möglichkeit, sukzessive erweitert zu werden und somit das Wissen des Themengebiets zu sammeln.

Tabelle 3-4: Lösungsmusterkategorisierung nach GAMMA et al. [GHJ+93, S. 409]

		Charakterisierung		
		Gestalterisch	Strukturell	Verhaltensbezogen
Zuständigkeit	Klasse	▪ Fabrikmethodenmuster	▪ Adaptermuster (Klasse) ▪ (...)	▪ Vorlagemuster ▪ (...)
	Objekt	▪ Abstraktes Fabrikmuster ▪ (...)	▪ Adaptermuster (Objekt) ▪ (...)	▪ Verantwortungskettenmuster ▪ (...)
	Verbindung	▪ Erstellungsmuster	▪ Zusammensetzungsmuster ▪ (...)	▪ Interpreter Muster ▪ (...)

Bewertung: Lösungsmusterkataloge umfassen das explizierte Wissen innerhalb eines bestimmten Themengebiets. Die Kategorisierung ermöglicht es dem Anwender, bedarfsorientiert passende Lösungsmuster zu finden. Gegeben dem Fall, dass die Einordnungskriterien relevant sind und dem Themengebiet entsprechen, kann der Suchbereich anhand der Kategorien eingegrenzt werden und Lösungsmuster leichter identifiziert werden. Gleichzeitig wird die Nutzbarkeit von Lösungsmustern erhöht, da ähnliche oder alternative Muster innerhalb einer Kategorie aufgeführt und verglichen werden können. Auch bietet eine Kategorie Hinweise auf mögliche weitere Lösungsmuster, die es zu identifizieren gilt.

3.3.2.2 Systemarchitekturmuster nach CLOUTIER

Eine strukturierte und dem Anwendungsgebiet entsprechende Dokumentation ist die Basis für die Nutzung und Anwendung von Lösungsmustern. CLOUTIER beschreibt in seiner Arbeit ein Notationsschema, welches sich in Anlehnung an ALEXANDER et al. an die vier wesentlichen Kategorien *Name*, *Kontext*, *Problem* und *Lösung* orientiert [AIS+77, S. xf.]. Auf Basis seiner Untersuchungen konkretisiert CLOUTIER die Kategorien für den Anwendungsbereich komplexer Systeme. Insgesamt 16 Kategorien sollen dem Anwender alle relevanten Informationen eines Lösungsmusters zusammenfassen (Tabelle 3-5). Dies soll die Nutzung und Wiederverwendung von Wissen in einem effizienten und effektiven Engineering Prozess ermöglichen [Clo06].

Das Notationsschema umfasst neben allgemeinen Informationen wie Name, Referenzen und Autor auch Kategorien, die Rückschlüsse und Zusammenhänge zu weiteren Lösungsmustern ermöglichen. Dazu gehört die Diskussion der Schnittstellen zu angrenzenden Elementen und verwandten Mustern. Das Problem selbst wird in einem Feld beschrieben und durch eine Problemeinordnung in einem weiteren Feld ergänzt. Die Lösungsbeschreibung erfolgt über mehrere Kategorien, in denen auf mehrere Aspekte eingegangen wird. Insbesondere die Möglichkeit, Diagramme zu ergänzen, gilt als wesentlicher Faktor, um das Verständnis im Rahmen des Engineerings zu erhöhen.

Tabelle 3-5: Systemarchitekturmuster nach CLOUTIER [Clo06, S. 42]

Kategorien	Erklärung der Kategorien
Name des Musters	Der Name des Musters sollte beschreibend sein, damit der Nutzer des Musters die Verwendung verstehen kann.
Aliase	Weitere Namen, unter denen das Muster bekannt sein könnte.
Schlüsselwörter	Schlüsselwörter, die beim Auffinden geeigneter Muster in einer Datenbank helfen.
Problemkontext	Kurze Diskussion über die Arten von Situationen, in denen das Problem auftreten kann. Sie sollte eine beliebig große Anzahl von Situationen berücksichtigen, in denen das Problem auftreten kann.
Problembeschreibung	Welches Problem kann mit diesem Muster gelöst werden?
Herausforderungen	Herausforderungen, die bei dem vom Muster behandelten Problem bestehen, und die Probleme, die bei der Anwendung des Musters auftreten.
Lösung	Beschreibung, wie das Muster das beschriebene Problem löst.
Diagramme	Dies können ein oder mehrere Diagramme sein, die zur Darstellung des Musters erforderlich sind. Dabei kann jede beliebige Notationsmethode verwendet werden.
Schnittstellen	Diskussion der kritischen Schnittstellen oder Informationsflüsse, die für die Implementierung des Musters erforderlich sind.
Resultierender Kontext	Was sind die ungelösten Probleme, die bei der Anwendung des Musters verbleiben?
Beispiele	Ein Beispiel, wie das Muster angewandt werden kann.
Funktionsweise	Begründung, warum das Muster funktioniert.
Übertragungsmöglichkeiten	In welchen anderen Anwendungen kann das Muster eingesetzt werden?
Verwandte Muster	Andere Muster, die in Verbindung mit diesem Muster stehen.
Referenzen	Weitere Informationen, die für das Verständnis oder die Anwendung des Musters nützlich sein können.
Autoren	Wer hat das Muster dokumentiert?

Bewertung: Die Strukturierung und Auswahl an Kategorien bietet eine fundierte Basis zur Beschreibung von Lösungsmustern. Neben den relevanten Kategorien werden zahlreiche Zusatzinformationen in Kategorien aufgeführt, die das Verständnis erhöhen und gleichzeitig die Wiederverwendung wahrscheinlicher machen. Die Vielzahl an Kategorien bedeutet jedoch einen hohen initialen Aufwand bei der Dokumentation und kann den Anwender mit zu vielen Informationen überlasten. Insbesondere die Nutzung von Diagrammen für ein schnelles Verständnis von Zusammenhängen wird für die Gestaltung daher als sinnvoll erachtet.

3.3.2.3 Lösungsmusterbasierter Entwurf fortgeschrittener mechatronischer Systeme nach ANACKER

ANACKER stellt ein Vorgehen für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme im Kontext der Produktentstehung vor (Bild 3-21). Der Einsatz der Lösungsmuster wird in Einklang mit dem Entwicklungsgeschehen gebracht und auf unterschiedlichen Konkretisierungsstufen integriert. Der Abgleich zwischen Problemstellungen und bereits existierender Lösungsmuster wird durch einen zuvor eingeführten Wissensraum unterstützt, in dem sich die Lösungsmuster je nach Generalisierungs- und Komplexitätsgrad einordnen lassen [Ana15].

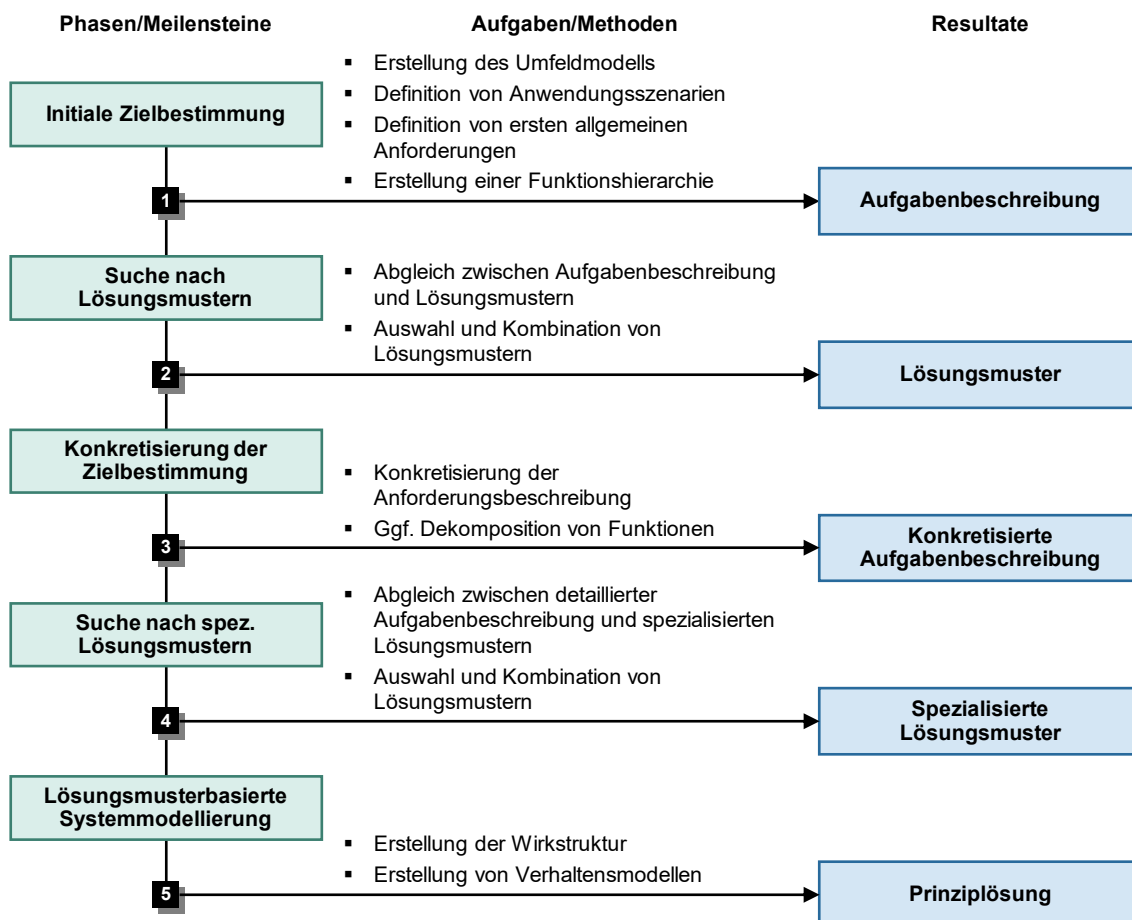


Bild 3-21: Vorgehen für einen lösungsmusterbasierten Systementwurf nach Anacker [Ana15, S. 131]

Initiale Zielbestimmung: Gegenstand dieser Phase ist eine möglichst lösungsneutrale Aufgabenbeschreibung auf Basis von etablierten Entwicklungsschritten der Systementwicklung. Dazu gehören die Beschreibung eines Umfeldmodells, Anwendungsszenarien sowie allgemeinen Anforderungen und eine Funktionshierarchie. Diese geben Entwicklerinnen und Entwickler ein umfassendes Bild des zu erreichenden Ziels, welches im weiteren Verlauf erreicht werden soll.

Suche nach Lösungsmustern: In der zweiten Phase werden erste Lösungsmuster auf Basis der Aufgabenbeschreibung identifiziert. Zwei Suchstrategien werden vorgeschlagen, um passende Lösungsmuster zu erkennen. Dabei wird die Funktionshierarchie entweder Top-Down oder Bottom-Up mit entsprechenden Lösungsmustern verglichen.

Konkretisierung der Zielbestimmung: In der dritten Phase werden die identifizierten Lösungsmuster genutzt, um die Aufgabenbeschreibung durch das erlangte Wissen zu konkretisieren. Dazu gehören neue Anforderungen sowie die weitere Dekomposition der Funktionshierarchie, die sich aus der Auswahl und Kombination bestimmter Lösungsmuster ergibt.

Suche nach spezialisierten Lösungsmustern: Durch die konkretisierte Zielbestimmung ergeben sich neue Ansatzpunkte für Lösungsmuster. Insbesondere spezialisierte Lösungsmuster sollten in dieser Phase noch einmal durchsucht werden.

Lösungsmusterbasierte Systemmodellierung: Anhand der Auswahl an Lösungsmustern wird die Spezifikation konkretisiert. Hierzu werden eine Wirkstruktur sowie Verhaltensmodelle erarbeitet, die zusammen die Prinzipiellösung des Systems darstellen. Die Entwicklung erfolgt sowohl durch die Adaptierung der Lösungsmuster auf die spezifische Problemstellung als auch durch neu entwickelte Lösungselemente.

Bewertung: ANACKER beschreibt ein systematisches Vorgehen, in dem Lösungsmuster über mehrere Konkretisierungsstufen in den Systementwurf einfließen. Hintergrund sind klar abzugrenzende Systeme, deren Entwicklung durch Lösungsmuster unterstützt werden. Dies trifft auf SoS nicht zu, da zum einen unterschiedliche Rahmenbedingungen vorliegen können und SoS als Netzwerke keine klaren Grenzen und Umfänge besitzen. Dennoch zeigt das Vorgehen, dass ein kontinuierlicher Abgleich mit vorhandenem Lösungswissen sinnvoll ist. Dieser kann an unterschiedlichen Stellen nach einer initialen Zielbestimmung und Problembeschreibung erfolgen. Der Autor weist darauf hin, dass nicht für jede Problemstellung Lösungsmuster in dokumentierter Form vorliegen und daher Entwickler nach wie vor neue Lösungen finden müssen, die wiederum in neue Lösungsmuster überführt werden sollten. Dennoch wird das Entwicklungsgeschehen deutlich effizienter und Entwicklerinnen und Entwickler werden durch Lösungsmuster inspiriert, neue Wege zu gehen.

3.4 Bewertung und Handlungsbedarf

Im Folgenden werden die im Stand der Forschung vorgestellten Ansätze mit den in Kapitel 2.6 definierten Anforderungen an ein *Rahmenwerk zur musterbasierten Gestaltung von SoS* abgeglichen. Eine Zusammenfassung der Bewertung wird in Bild 3-22 visualisiert.

A1) Definition von SoS im Kontext von technischen Systemen: Es existieren mehrere Definitionen von SoS. Mit Ausnahme von TEKINERDOGAN, der auf weitere Typen wie bspw. soziotechnische SoS eingeht und den Ansätzen aus angrenzenden Themengebieten,

legen die vorgestellten Ansätze den Fokus auf technische SoS. Dennoch werden Unterschiede deutlich, sodass der Bedarf an einer einheitlichen Definition besteht.

A2) Berücksichtigung der charakteristischen Merkmale von SoS: Es gibt mehrere Quellen, die SoS anhand von charakteristischen Merkmalen beschreiben. Diese unterscheiden sich jedoch, sodass ein Bedarf an einer einheitlichen Ausgangsbasis deutlich wird. Viele Ansätze berücksichtigen die aufgeführten charakteristischen Merkmale von SoS. Jedoch gehen sie meist nur auf einzelne Merkmale ein und lassen andere Merkmale unberücksichtigt. Um die Herausforderungen von SoS vollständig zu adressieren, muss das Rahmenwerk auf sämtliche Merkmale eingehen.

A3) Klassifizierung von SoS: Es zeigt sich insgesamt, dass die Unterscheidung von SoS anhand von Klassifikationen etabliert ist, jedoch unterschiedlich ausgeprägt wird. Die Anforderung an eine Taxonomie erfüllt TEKINERDOGAN durch die hohe Komplexität nicht. Zusätzlich sind die Unterscheidungen der Klassen nicht immer eindeutig und eine anschauliche Repräsentation der Thematik wird nicht gegeben, sodass der Nutzen für Anwender gering ist. Es besteht der Bedarf an einer sparsamen und eingängigen Klassifikation, die leicht verständlich ist und eine Ausgangsbasis für die Strukturierung von Gestaltungsmaßnahmen bietet.

A4) Berücksichtigung von SoS- und Systemperspektiven: Die Notwendigkeit einer Unterscheidung zwischen der SoS- und der Systemperspektive zeigen beispielsweise die Ansätze nach BAEK et al. oder auch BÖHM et al. Dies gilt es im Rahmen einer ganzheitlichen Lösung zu berücksichtigen.

A5) Gestaltungsmethoden strukturieren und bereitstellen: Der Ansatz nach ADAMS und KEATING zeigt, dass Gestaltungsmethoden je nach Ausgangssituation des SoS angepasst werden sollten. Es mangelt jedoch an der konkreten Bereitstellung von Methoden. Das Wellenmodell nach DAHMANN et al. bietet eine erste gute Grundlage zur Unterscheidung von Gestaltungsmethoden, in dem es zwischen Initialisierungs- und Update-Phase unterscheidet. Jedoch ist der Ansatz lediglich für SoS mit klaren Managementstrukturen geeignet.

A6) Anwendbarkeit sicherstellen: Die Anwendbarkeit ist in mehreren Ansätzen gegeben. Häufig sind die Ansätze jedoch zu generisch, zu spezifisch oder nur mit einem hohen Modellierungs- und/oder Anpassungsaufwand verbunden, was die Akzeptanz von Anwendern beeinträchtigen kann.

A7) Wissenstransfer ermöglichen: Der Lösungsmusteransatz eignet sich generell, um die Kommunikation und Kooperation zwischen Unternehmen zu fördern. Die Lösungsmuster nach INGRAM et al und PETITDEMANGE et al. folgen einer einheitlichen Strukturierung, was die Anwendbarkeit deutlich steigert. Die Verständlichkeit der analysierten Lösungsmuster ist nur in Teilen gegeben. Neben sehr spezifischen Lösungsmustern, wie im Fall der Rekonfigurationsmuster, wird häufig auf domänenspezifische Sprache zu-

rückgegriffen. Dies wird bei den stark aus der Softwaretechnik geprägten Ansätzen deutlich. Ein Bedarf an einheitlichen Mustern in domänenübergreifender Sprache wird deutlich.

A8) Bereitstellung von SoS-Lösungswissen: Wie u. a. UDAY und MARAIS beschreiben, eignen sich Designprinzipien als Orientierung für die Gestaltung. Die analysierten Ansätze für Designprinzipien zeigen die vielfältigen Hinweise für die Gestaltung und verdeutlichen den Bedarf, diese zu strukturieren und im Rahmen der SoS-Gestaltungen verfügbar zu machen. Keiner der Ansätze geht jedoch über die Auflistung und textuelle Beschreibung hinaus. Eine gute Ausgangsbasis für die Darstellung von Wissen in Form von Lösungsmustern stellen die Lösungsmuster nach INGRAM et al. dar. Diese sind als einzelne Muster beschrieben, ohne Zusammenhänge und Beziehungen zu SoS oder weiteren Mustern herzustellen.

Keiner der untersuchten Ansätze sowie keine triviale Kombination erfüllt alle Anforderungen in vollem Umfang. Dies hängt nicht zuletzt mit den vielfältigen Ausprägungen von SoS zusammen, die zu unterschiedlichen Lösungsansätzen geführt haben und häufig nur Teilaspekte von SoS berücksichtigen. Es besteht demnach dringender Handlungsbedarf nach einem *Rahmenwerk zur musterbasierten Gestaltung von SoS*.

Bewertungsschema			Anforderungen							
			A1	A2	A3	A4	A5	A6	A7	A8
☐ nicht erfüllt ☒ teilweise erfüllt ☒ voll erfüllt										
Ansätze zur Beschreibung von SoS	Charakterisierung	Charakterisierung nach MAIER	☒	☒	☐	☒	☐	☐	☐	☐
		Charakterisierung nach BOARDMAN und SAUSER	☒	☒	☐	☒	☐	☐	☐	☐
		Merkmale nach ABBOTT	☒	☒	☐	☒	☐	☐	☐	☐
	Klassifikation	Taxonomie nach MAIER	☒	☒	☒	☒	☐	☐	☒	☐
		Erweiterte Klassifikation nach DAHMANN und BALDWIN	☒	☒	☒	☒	☐	☐	☒	☐
		Multidimensionale Klassifikation nach TEKINERDOGAN	☒	☒	☒	☒	☐	☐	☐	☐
Ansätze zur Gestaltung von SoS	SoS	Wellenmodell der SoS-Entwicklung nach DAHMANN et al.	☒	☒	☒	☒	☒	☒	☐	☐
		SoSE-Prozess nach SAGE und BIEMER	☒	☒	☐	☒	☒	☒	☐	☐
		SoSE-Methodik nach ADAMS und KEATING	☒	☒	☐	☐	☐	☐	☐	☐
		SoS Lifecycle Engineering nach FORTE et al.	☒	☒	☐	☒	☐	☒	☐	☐
		Rahmenwerk für SoS-Gestaltungsprozesse nach SHAKED und REICH	☒	☐	☐	☐	☐	☒	☐	☐
		Modellbasierter SoSE-Ansatz nach BAEK et al.	☒	☒	☐	☒	☒	☒	☒	☐
		Rahmenwerk für die SoS-Synthese, Bewertung und gesteuerte Entwicklung nach SELBERG und AUSTIN	☒	☒	☐	☒	☒	☒	☒	☐
		Prozessrahmenwerk für SoS nach ISO 21840	☒	☒	☒	☒	☐	☒	☐	☐
	Vernetzte Systeme	Planung von CPS nach WESTERMANN	☒	☒	☐	☐	☒	☒	☐	☐
		Modellbasiertes Engineering von kollaborativen eingebetteten Systemen nach BÖHM et al.	☐	☒	☒	☒	☒	☒	☐	☐
		Multiagenten-System Engineering Methode nach WOOD und DELOACH	☐	☒	☐	☐	☒	☒	☐	☐
		Ansätze der Systemdynamik nach LANE	☐	☒	☐	☐	☒	☒	☐	☐
Wissensbasierte Ansätze zur Unterstützung der SoS-Gestaltung	SoS-Wissen	Modellierungsmuster für SoS-Architekturen nach INGRAM et al.	☒	☒	☐	☐	☐	☒	☒	☒
		Rekonfigurationsmuster für SoS nach PETITDEMANGE et al.	☒	☒	☐	☐	☐	☒	☒	☒
		Interoperabilitätsmuster nach BRÖRING et al.	☒	☒	☐	☐	☐	☒	☒	☒
		Muster für vertragsbasierte SoS nach FALDIK et al.	☒	☒	☐	☐	☐	☐	☒	☒
		Design-Prinzipien für resiliente SoS nach UDAY und MARAIS	☒	☒	☐	☐	☐	☒	☒	☒
		Naturbasierte Designprinzipien für SoS nach AZANI	☒	☒	☐	☐	☐	☒	☒	☒
		Evolvabilitätsbezogene Prinzipien für SoS nach RICCI et al.	☒	☒	☐	☐	☐	☒	☒	☒
	Wissensbasierte Ansätze	Musterkatalog nach GAMMA et al.	☐	☐	☐	☐	☐	☒	☒	☐
		Systemarchitekturmuster nach CLOUTIER	☐	☐	☐	☐	☐	☒	☒	☐
		Lösungsmusterbasierter Entwurf für fortgeschrittene mechatronische Systeme nach ANACKER	☐	☐	☐	☐	☐	☒	☒	☐

Bild 3-22: Bewertung des untersuchten Stands der Forschung anhand der Anforderungen

4 Rahmenwerk zur musterbasierten Gestaltung von SoS

Dieses Kapitel bildet den Kern der vorliegenden Arbeit. Es stellt ein *Rahmenwerk zur musterbasierten Gestaltung von System of Systems* vor. Dieses hat den Anspruch, dem beschriebenen Handlungsbedarf gerecht zu werden und die in Kapitel 2.6 gestellten Anforderungen zu erfüllen. Das Rahmenwerk berücksichtigt den aktuellen Stand der Forschung und erweitert diesen. Zunächst gibt **Kapitel 4.1** einen Überblick über das Rahmenwerk und stellt die einzelnen Bestandteile vor. Das Fundament des Rahmenwerks bildet **Kapitel 4.2** mit einer Definition und Charakterisierung von technischen SoS. Sowohl das Vorgehen als auch das Ergebnis der Definition für technische SoS werden detailliert beschrieben. Darauf aufbauend wird in **Kapitel 4.3** der Gestaltungsrahmen vorgestellt. Dieser strukturiert einzelne Gestaltungskonzepte zur Gestaltung von SoS unter Berücksichtigung von unterschiedlichen Ausprägungen der SoS-Charakteristika. Ein Vorgehensmodell unterstützt die Anwendung des Gestaltungsrahmens. Darüber hinaus werden in **Kapitel 4.4** Hilfsmittel zur Gestaltung von SoS beschrieben. Die Hilfsmittel gliedern sich in Designprinzipien sowie Lösungsmuster und sind ein integraler Bestandteil des Lösungsansatzes.

4.1 Das Rahmenwerk im Überblick

Aufgrund der beschriebenen Herausforderungen bei der Gestaltung von SoS lassen sich diese nicht ausreichend durch die derzeitigen Ansätze beherrschen. In den klar abgegrenzten Systemmodellen des MBSE werden bspw. die Unvorhersagbarkeit von Interaktionen zwischen ES oder auch die unklaren Systemgrenzen von SoS nicht ausreichend adressiert. Neben einer klaren Definition und Beschreibung von SoS werden Ansätze und Hilfsmittel benötigt, die es den Unternehmen ermöglichen, die Komplexität von SoS anzunehmen und Lösungen zu finden.

Vor diesem Hintergrund wird im Folgenden das *Rahmenwerk zur musterbasierten Gestaltung von System of Systems* vorgestellt (Bild 4-1). Das Fundament bildet die **Definition und Klassifizierung von System of Systems**, die die charakteristischen Merkmale von SoS berücksichtigt. Aufbauend auf einem einheitlichen Verständnis von SoS wird ein **Gestaltungsrahmen** mit fünf Gestaltungskonzepten vorgestellt, die auf die unterschiedlichen Merkmale und Ausprägungen von SoS eingehen. Für die jeweiligen Gestaltungskonzepte werden geeignete Leitlinien bereitgestellt und auf Methoden verwiesen. Zusätzlich wird in der zweiten Säule **Gestaltungswissen** strukturiert und für die Gestaltung von SoS vorbereitet. Dieses umfasst Designprinzipien sowie Lösungsmuster in Form von explizitem Wissen, welches für die Anwendung zur Verfügung gestellt wird.

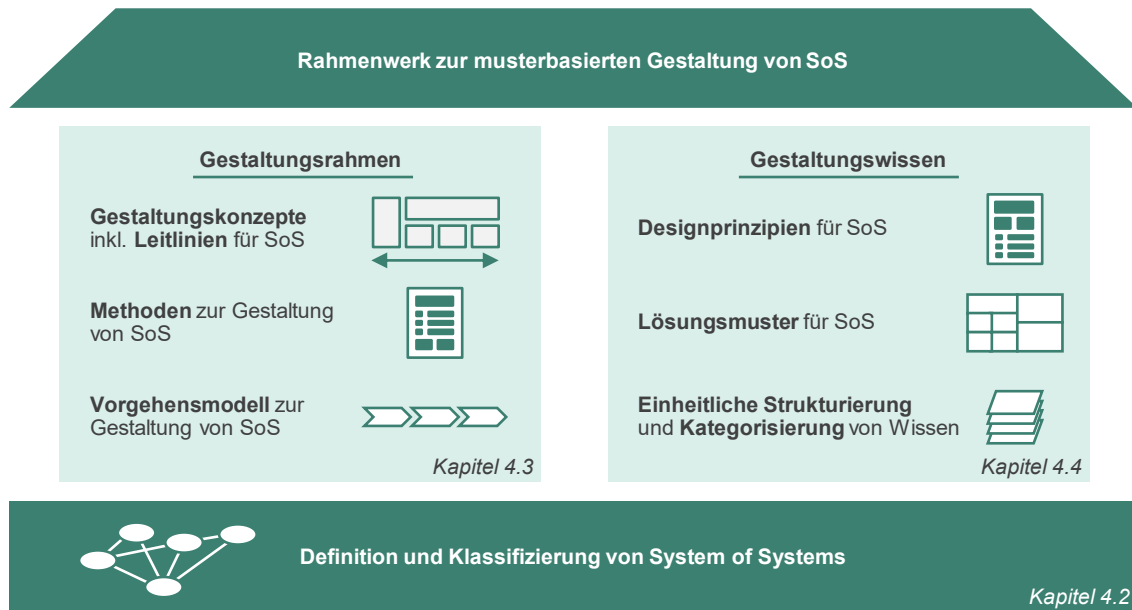


Bild 4-1: Rahmenwerk zur musterbasierten Gestaltung von SoS

Das Ziel des Rahmenwerks ist es, die Gestaltung von SoS durch Unternehmen und Organisationen unter Berücksichtigung von spezifischen Herausforderungen zu unterstützen. Auf Basis eines einheitlichen Verständnisses über System of Systems und den unterschiedlichen Ausprägungen soll die Kommunikation und Koordination von unterschiedlichen Unternehmen auf SoS-Ebene gefördert werden. Es gilt, unterschiedliche Zusammenarbeitsmodelle zu berücksichtigen und eine Grundlage für die Gestaltung in unterschiedlichen Konstellationen bereitzustellen. Die SoS-Gestaltungskonzepte sollen eine Ausgangsbasis für die Zusammenarbeit darstellen, ohne die Entwicklung bis ins letzte Detail vorzudefinieren. Dies hängt nicht zuletzt mit den zahlreichen Unwägbarkeiten von SoS zusammen. In Anlehnung an die Modellierungszwecke nach TSCHIRNER sowie FRIEDENTHAL et al. soll vielmehr die Entwicklung eines ersten Konzepts unterstützt werden, welches die Ausgangsbasis für detaillierte Designs darstellt [Tsc16, S. 111], [FMS12, S. 22]. Aus diesem Grund und zur Unterstützung des Ziels der unternehmensübergreifenden Kommunikation und Kooperation wird in dem Rahmenwerk ein semi-formaler Ansatz angestrebt. Dieser kann in konkreten Anwendungsgebieten als Grundlage für formale Ansätze genutzt werden und in diese übergehen. Ein semi-formaler Ansatz unterstützt die Dokumentation und Kommunikation von Zusammenhängen während der Gestaltung. Außerdem wird die Kooperation durch die Vermeidung eines initial hohen Aufwands zur Abstimmung von formalen Beschreibungsmodellen vermieden. Um darüber hinaus die zahlreichen Herausforderungen bei der Gestaltung von SoS zu adressieren, soll kollektives Wissen über etablierte Lösungsansätze in Form von Designprinzipien und Lösungsmustern aufbereitet und strukturiert zur Verfügung gestellt werden.

4.2 Definition und Klassifizierung von SoS

Die heterogenen Betrachtungsweisen und Beschreibungen von SoS in der Literatur sind Hindernisse für eine unternehmensübergreifende Gestaltung von SoS. Ein gemeinsames Verständnis wird benötigt, um SoS einzuordnen und die Komplexität greifbar zu machen sowie die unternehmensübergreifende Kooperation zu fördern. Die Problemanalyse hat gezeigt, dass das Verständnis von SoS sehr heterogen ist und es unterschiedliche Ausprägungen von SoS gibt. Dies hat nicht zuletzt mit der Vielzahl an Anwendungsgebieten und der verfügbaren Literatur zu dem Thema zu tun.

Zur Analyse und Konsolidierung eines allgemeinen Verständnisses wird daher die Methode der systematischen Literaturanalyse (SLR) angewandt. Die Analyse stützt sich auf dem Vorgehen nach XIAO und WATSON und gliedert sich in drei Phasen: Planung, Durchführung und Dokumentation der Literaturanalyse [XW19, S. 103].

In der **Planungsphase** wird das Problem beschrieben sowie das Ziel formuliert (vgl. Kapitel 2.2.4). Um das Ziel einer einheitlichen Definition zu erreichen, wird eine geeignete Datenbank ausgewählt und anhand folgender Suchanfrage durchsucht:

Suchanfrage (SCOPUS): (system of systems OR systems of systems) AND (engineering OR design) AND (definition OR classification OR characteristics).

Im Rahmen der **Durchführungsphase** wird die Suche initiiert und der Datensatz von insgesamt 778 Literaturquellen durch unterschiedliche Filtermechanismen und Relevanzprüfungen eingegrenzt. Bild 4-2 gibt einen Überblick über die durchgeführte Analyse.

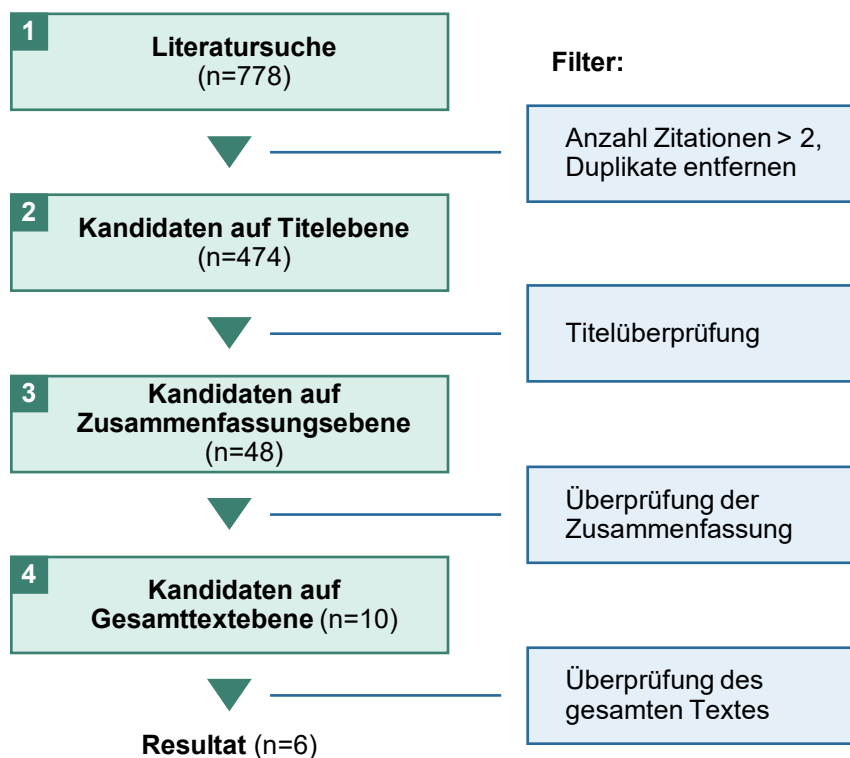


Bild 4-2: Vorgehen im Rahmen der systematischen Literaturanalyse

Der finale Datensatz von sechs Literaturquellen stellt die Ausgangsbasis für die Analyse und Synthetisierung der Informationen hinsichtlich der Problemstellung dar. Diese wird im folgenden Kapitel vorgestellt. Die im Vorgehen von XIAO und WATSON beschriebene letzte **Phase der Dokumentation** erfolgte im Rahmen einer Veröffentlichung und wird innerhalb dieser Arbeit fortgeführt [AGW+22].

4.2.1 Definition und Charakterisierung von technischen SoS

Die Ergebnisse der SLR zeigen, dass es mehrere Definitionen für SoS gibt, die jeweils unterschiedliche Schwerpunkte setzen. Die unterschiedlichen Quellen stimmen überein, dass ein SoS ein Verbund aus mehreren Einheiten ist. Folgende Beschreibung vereint das Verständnis über die verschiedenen Literaturquellen und kann auf diese zurückgeführt werden:

„Ein SoS ist ein Verbund aus mehreren Systemen oder Komponenten, die miteinander interagieren und kommunizieren, um eine übergeordnete Aufgabe zu erfüllen.“ [AGW+22, S. 180]

Zusätzlich wird der Bedarf deutlich, SoS über ihre Charakteristika genauer zu beschreiben. Diese unterscheiden sich in den unterschiedlichen Quellen. Aus diesem Grund werden die genannten Merkmale aus den identifizierten Quellen in Anlehnung an das Vorgehen nach MANDEL et al. extrahiert und konsolidiert [AMY+18, S. 2800f.]. Tabelle 4-1 zeigt die Zusammenführung der genannten Merkmale aus den relevanten Quellen der SLR.

Tabelle 4-1: Konsolidierung der charakteristischen Merkmale von SoS anhand der Beschreibung in den jeweiligen Literaturquellen

Literaturquelle						Merkmals-Nr.:
MAIER (1998, 2015)	DAHMAN und BALDWIN (2008)	NIELSEN et al. (2015)	BOARDMAN und SAUSER (2006)	TEKINERDOGAN (2019)	ABBOTT (2006)	
Managerial Independence	Managerial Independence	Autonomy of Constituents	Autonomy	Managerial Independence		→ 1.
Operational Independence	Operational Independence	Independence	Autonomy / Diversity	Operational Independence		→ 2.
Geographic Distribution		Distribution		Geographic Distribution		→ 3.
Evolutionary Development		Evolution		Evolution	Continually but slowly evolving	→ 4.
		Dynamic Reconfiguration	Connectivity / Diversity	Physical and logical config.	Open at the top, open at the bottom	→ 5.
Emergent Behavior		Emergence of Behavior	Emerging	Emergent Behavior		→ 6.
		Interdependence	Belonging / Diversity			→ 7.
		Interoperability	Diversity	Interoperability		→ 8.

Auf Basis der Auswertung und der Beschreibung in den unterschiedlichen Quellen wird im Folgenden ein konsolidiertes Verständnis der acht wesentlichen Merkmale von SoS aufgeführt:

- 1) **Managementunabhängigkeit:** Die Systeme im SoS verhalten sich nach eigenen Regeln, die von unabhängigen Managements gesetzt werden. Sie werden einzeln entwickelt und in das SoS integriert, ohne von anderen Systemen kontrolliert zu werden [Mai98, S. 271], [BS06, S. 119].
- 2) **Operative Unabhängigkeit:** Die betriebliche Eigenständigkeit beschreibt die Möglichkeit, dass interagierende Systeme unabhängig von anderen Systemen einen Zweck oder eine Aufgabe erfüllen können, ohne von anderen Systemen beeinflusst zu werden. Dies ist der eigentliche Grund für die Existenz eines Systems, der auch im SoS nicht verändert wird. Wird ein SoS in seine Einzelteile zerlegt, so können die Systeme unabhängig voneinander nützliche Operationen ausführen. [Mai98, S. 271], [BS06, S. 119], [Mai15, S. 13].
- 3) **Verteilung:** Die (geografische) Verteilung beschreibt die räumliche Distanz zwischen interagierenden Systemen, die häufig groß und verteilt ist. Im größtmöglichen Ausmaß ist eine ausschließlich auf Informationen-basierte Interaktion im SoS möglich. Die Interaktion ist jedoch nicht limitiert auf den Austausch von Informationen, sondern kann auch Energie und Masse umfassen [NLF+15, S. 10], [Mai15, S. 13].
- 4) **Evolution:** Der Großteil der SoS ist langlebig und verändert sich im Laufe der Zeit. Zum einen können sich die Funktionalitäten und deren Qualität ändern, zum anderen ändert sich die Struktur, die Zusammensetzung oder auch der Zweck eines SoS [Abb06, S. 42], [Mai15, S. 13].
- 5) **Dynamische Rekonfiguration:** Die dynamische Rekonfiguration beschreibt die Fähigkeit, die Struktur und die Zusammensetzung des SoS zu ändern, ohne jeglichen Eingriff von außen. Dies erfordert die Fähigkeit, Schnittstellen anzupassen und dynamisch Systeme zu vernetzen und wieder zu entkoppeln. Die Anpassung kann zusätzlich durch die Erweiterung des SoS sowohl auf übergeordneter als auch auf untergeordneter Ebene erfolgen [BS06, S. 119f.], [NLF+15, S. 10f.], [Abb06, S. 42].
- 6) **Emergentes Verhalten:** Im Zusammenhang mit SoS bezieht sich die Emergenz auf das Verhalten, das sich aus dem Zusammenwirken von interagierenden Systemen ergibt. Das resultierende Verhalten ermöglicht eine höhere Funktionalität als die interagierenden Systeme eigenständig bereitstellen können. Die Emergenz zeigt sich in gewolltem und ungewolltem Verhalten, welches im Design nicht vollständig vorhergesagt werden kann und sich während des Betriebs einstellt [Mai15, S. 13], [BS06, S. 120f.].
- 7) **Interdependenz:** Die Interdependenz beschreibt die gegenseitige Abhängigkeit zwischen den interagierenden Systemen zur Erfüllung des Ziels des SoS. Weicht

das individuelle Ziel der einzelnen Systeme vom Gesamtziel des SoS ab, so verzichten die interagierenden Systeme auf einen Teil ihres eigenen Verhaltens, um die Anforderungen des SoS zu erfüllen [NLF+15, S. 11], [BS06, S. 119].

- 8) **Interoperabilität:** Um einen Verbund von Systemen zu bilden, muss das SoS in der Lage sein, eine Auswahl heterogener Systeme einzubinden. Dazu gehören die Integration und Anpassung von Schnittstellen, Standards und Protokollen, um alte und neue Systeme zu verbinden [NLF+15, S. 11], [BS06, S. 120].

Gemeinsam mit der zuvor genannten Beschreibung stellen die acht charakteristischen Merkmale das Verständnis von SoS im Rahmen dieser Arbeit dar.

4.2.2 Klassifizierung von SoS

Die Definition und Charakterisierung von SoS aus dem vorangegangenen Kapitel ist die Ausgangsbasis für die weiteren Ausarbeitungen. Die charakteristischen Merkmale sind wesentlich für die Unterscheidung zwischen einem System und einem SoS und stellen gleichzeitig die zentralen Herausforderungen für die Gestaltung von SoS dar.

In den Ausführungen der Autoren wird deutlich, dass die einzelnen Merkmale unterschiedlich stark ausgeprägt sein können, ohne dass konkrete Merkmalsausprägungen genannt werden. Aufbauend auf den theoretischen Grundlagen werden daher SoS-Beispiele herangezogen, um die Relevanz der Merkmale und die unterschiedlichen Ausprägungen zu analysieren. Um eine geeignete Analysegrundlage zu bieten und die Vergleichbarkeit der Beispiele zu gewährleisten, wurde ein Steckbrief erarbeitet, der einen groben Überblick über die Struktur, die Zusammenhänge sowie die Ziele des SoS gibt. Zusätzlich wurde für jedes Merkmal eine dreistufige Skala von einer geringen bis starken Ausprägung eingeführt. Im Rahmen eines Workshops mit acht Industriepartnern wurden insgesamt zehn Beispiele von SoS in Form des Steckbriefs ausgearbeitet und anhand der Merkmale charakterisiert. Zusätzlich wurden zwölf Literaturbeispiele im selben Schema beschrieben und bewertet. Bild 4-3 zeigt einen ausgearbeiteten Steckbrief am Beispiel „Intelligentes Laden von Fahrzeugen“. Die weiteren Beispiele werden im Anhang A1.1 aufgeführt.

Die Analyse der Beispiele liefert mehrere Erkenntnisse. Zum einen zeigen sich in sämtlichen Beispielen die zuvor genannten Merkmale von SoS. Dadurch lässt sich feststellen, dass die analysierten Beispiele als SoS einzustufen sind und gleichzeitig die Merkmale selbst geeignet sind, um zwischen eigenständigen Systemen und SoS zu unterscheiden. Zum anderen wird festgestellt, dass sich die Ausprägungen der Merkmale zwischen den Beispielen unterscheiden. Dennoch zeigen sich bei einem ersten Vergleich gewisse Tendenzen, die auf Gemeinsamkeiten zwischen bestimmten Beispielen schließen lassen. Aufgrund dieser Erkenntnisse wird eine Datenanalyse basierend auf den Merkmalsausprägungen durchgeführt.

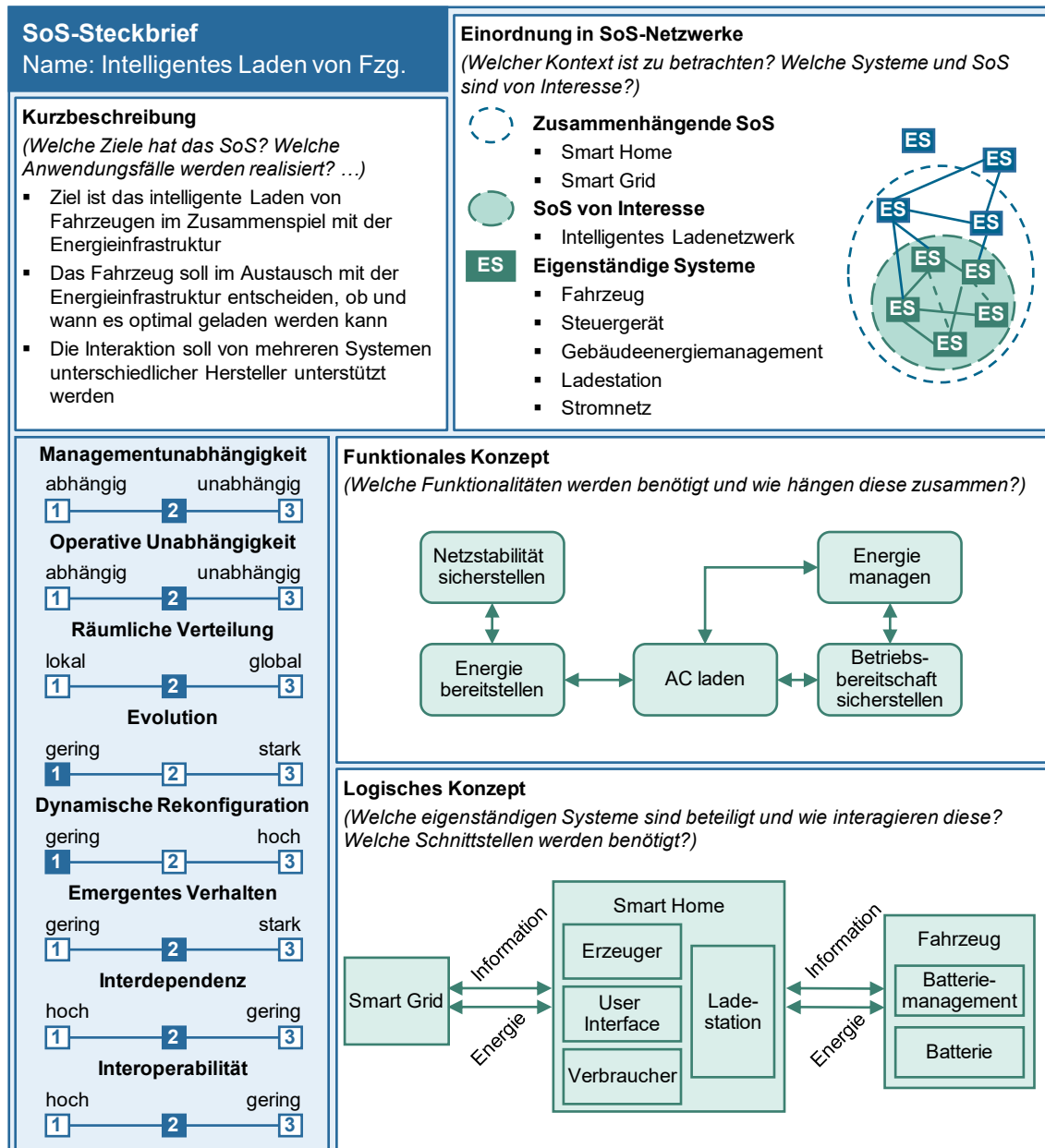


Bild 4-3: SoS-Steckbrief am Beispiel „Intelligentes Laden von Fahrzeugen“

Durch eine empirische Analyse der Daten in Form einer Clusteranalyse¹ werden diese Tendenzen auf Basis von Gemeinsamkeiten innerhalb von Clustern (interne Homogenität) und Unterschiede zwischen den Clustern (externe Heterogenität) deutlich. Nach Anwendung einer Statistiksoftware werden die Cluster mittels einer multidimensionalen Skalierung in einem mehrdimensionalen Raum visualisiert. Auch wenn die Datenanalyse keine Schlussfolgerung auf eine eindeutige Anzahl an Clustern zulässt, wurde die Tendenz der Auswertung in Bild 4-4 visualisiert. Als wesentliches Unterscheidungsmerkmal zeigt sich dabei die Managementunabhängigkeit, die zusätzlich einen Einfluss auf die

¹ Die analysierten Beispiele inkl. des abgeleiteten Datensatzes sowie das Vorgehen der Clusteranalyse werden im Anhang A1 beschrieben.

Ausprägung der Interoperabilität und Emergenz hat. Nichtsdestotrotz sind die Ergebnisse nicht eindeutig und müssen auf Basis von weiteren Erkenntnisse weiterentwickelt werden.

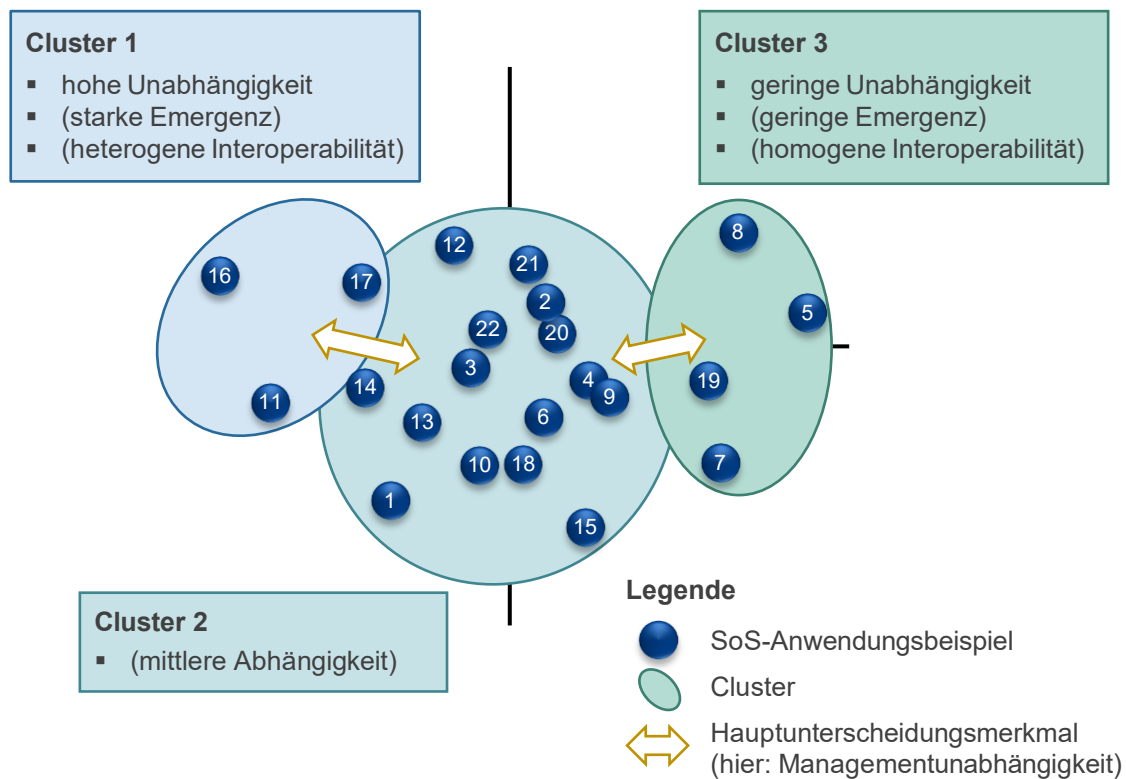


Bild 4-4: Vorläufige Ergebnisse der Clusteranalyse

Vor diesem Hintergrund wurde entsprechend dem Vorgehen zur Klassifizierung nach NICKERSON et al. im zweiten Schritt eine konzeptuell-zu-empirische Taxonomie angestrebt [NVM13, S. 345]. Diese baut auf den bestehenden Grundlagen sowie Erfahrungswerten auf, um relevante Dimensionen einer Klassifizierung abzuleiten [NVM13, S. 346].

Um eine dem Anwendungszweck entsprechende Klassifizierung vorzunehmen, wurden Erkenntnisse aus der Problemanalyse und dem Stand der Forschung um die Erfahrungswerte aus der Beispielanalyse ergänzt. Die resultierende Klassifizierung stützt sich vorrangig auf der Dimension der Managementunabhängigkeit. Diese wird sowohl in der Taxonomie nach MAIER (vgl. Kapitel 3.1.2.1) hervorgehoben als auch in der Beispielclusteranalyse als eines der Hauptunterscheidungsmerkmale deutlich. Ergänzt werden die Klassifikation u. a. durch Erkenntnisse zur Emergenz, wie sie in Kapitel 2.2.2.2 deutlich werden. Demnach verhält sich die Emergenz proportional zum Unabhängigkeitsgrad. Durch die Analyse der Gemeinsamkeiten innerhalb der Cluster und Unterschiede zwischen den Clustern werden drei Abstufungen zwischen geringer und starker Unabhängigkeit identifiziert. Diese sind gleichzeitig die Namensgeber der Klassen von SoS und werden in Bild 4-5 visualisiert.

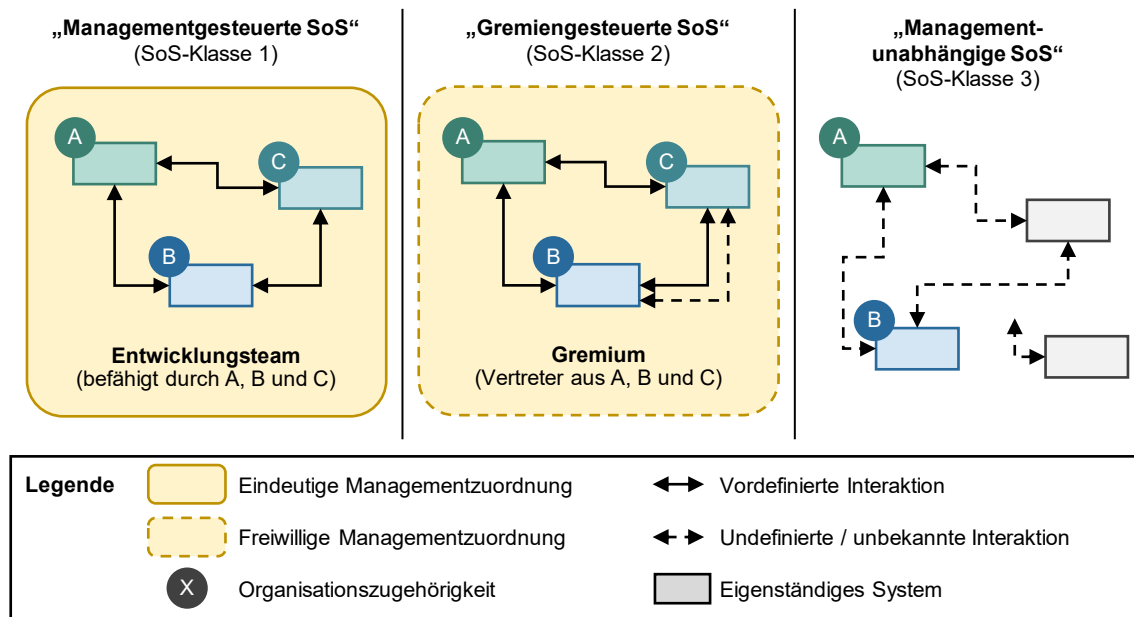


Bild 4-5: Klassifizierung von SoS

Die erste Klasse wird als **managementgesteuerte SoS** bezeichnet. Diese zeichnen sich durch eine eindeutige Managementzuordnung aus, indem Unternehmen kooperieren und das Zusammenspiel der eigenständigen Systeme im SoS definieren. Die Interaktionen sind vordefiniert und eng aufeinander abgestimmt.

In der zweiten Klasse werden **gremiengesteuerte SoS** zusammengefasst. Diese besitzen kein direktes Management, sondern werden durch Interessensgemeinschaften (Gremien) gesteuert, in denen Abläufe und Standards zur Interaktion zwischen eigenständigen Systemen für ein Anwendungsgebiet vordefiniert werden.

Die **managementunabhängigen SoS** stellen die dritte Klasse dar und zeichnen sich durch viele unbekannte Größen aus. Sowohl die beteiligten Systeme und deren Zugehörigkeit als auch die Verhaltensweisen sind zum größten Teil unbekannt und können nur schwierig vorhergesagt werden. Emergentes Verhalten zeigt sich hier besonders stark.

Die definierten Klassen erfüllen die Anforderungen an Klassifikationen nach NICKERSON et al. [NVM13, S. 341]. Sie sind *sparsam* und somit leicht verständlich für den Nutzer. Sie stützen sich auf eine begrenzte Anzahl an Faktoren und beziehen neben der Managementunabhängigkeit vor allem die Faktoren Interoperabilität und Emergenz mit ein. Die weiteren Faktoren, wie u. a. die Evolution, stellen keinen ausschlaggebenden Faktor dar und folgen keiner Logik. Daher wurden sie in der Klassifikation nicht berücksichtigt. Die drei beschriebenen Klassen sind außerdem *zweckmäßig* und werden im weiteren Verlauf zur Ableitung von passenden Handlungsweisen im Rahmen der Gestaltung von SoS genutzt.

4.3 Gestaltungsrahmen für das SoS-Engineering

Die Gestaltung von SoS ähnelt in den Grundzügen der von komplexen Systemen. Die operative Unabhängigkeit, die Managementunabhängigkeit, das emergente Verhalten sowie die stetige Weiterentwicklung sorgen jedoch dafür, dass nicht alle klassischen Entwicklungsprozesse, wie sie bspw. in der ISO 15288 beschrieben sind, bis ins letzte Detail durchgeführt werden können. Insbesondere bei der Architekturgestaltung von SoS können häufig nur Teile des Gesamtkonzepts von SoS beschrieben werden, da detaillierte Informationen über die interagierenden Systeme fehlen [ISO21840, S. 42]. Je nach Ausprägung der Merkmale von SoS kann mehr oder weniger gestaltet und auf das Zusammenspiel der Systeme in unterschiedlichem Maße Einfluss genommen werden. Ein Beispiel sind Abstufungen der Managementunabhängigkeit, welche maßgeblichen Einfluss auf die Möglichkeiten während der Gestaltung haben.

Um zwischen Handlungsweisen zu unterscheiden und bedarfsgerechte Methoden und Hilfsmittel für die Gestaltungsprozesse bereitzustellen, werden im Folgenden die Klassen von SoS als Ausgangsbasis genutzt (vgl. Kapitel 4.2.2). Es ist sinnvoll, Klassen zu nutzen, wenn die Objekte einer Kategorie wichtige Gemeinsamkeiten haben und ähnliche Ansätze und Best-Practices für die Entwicklung und Nutzung zeigen [Mai15, S. 12]. Objekte einer Kategorie haben einen bestimmten Bedarf und müssen für diesen gezielt unterstützt werden.

Daher wird in **Kapitel 4.3.1** zunächst ein Gestaltungsrahmen entwickelt, der auf der Klassifikation von SoS aufbaut und Konzepte für bestimmte Entwicklungsbedarfe bündelt. Die Gestaltungskonzepte berücksichtigen die unterschiedlichen Kooperationstypen zwischen den Unternehmen, die wesentlichen Einfluss auf die technischen Konzepte haben. Mit dem Gestaltungsrahmen wird daher eine gleichzeitige Betrachtung von technischen Herausforderungen und Managementherausforderungen unterstützt.

In **Kapitel 4.3.2** wird eine Methodensammlung bereitgestellt, welche verschiedene Methoden für die Gestaltung von SoS umfasst. Diese werden anhand der zuvor erarbeiteten Gestaltungskonzepte strukturiert und unterstützen unterschiedliche Herausforderungen bei der Gestaltung von SoS.

Kapitel 4.3.3 umfasst ein Vorgehensmodell zur Anwendung der Gestaltungskonzepte. Es ermöglicht die bedarfsorientierte Auswahl eines passenden Gestaltungskonzepts auf Basis der zuvor ermittelten Ausgangssituation.

Als Ergänzung wird in **Kapitel 4.3.4** ein zusätzliches Nutzungskonzept für den Gestaltungsrahmen vorgestellt. Dieses stellt einen Zusammenhang zwischen den verschiedenen Gestaltungskonzepten her und erläutert fünf Nutzungsoptionen.

4.3.1 Gestaltungskonzepte des SoS-Engineerings

Unternehmen müssen sich im Gestaltungsfeld von SoS mit veränderten und neuen Tätigkeiten auseinandersetzen. Aufbauend auf der Klassifikation von SoS in Kapitel 4.2.2 werden in diesem Kapitel die Auswirkungen auf die Gestaltung der einzelnen Klassen von SoS betrachtet.

Ein SoS ist nie vollständig und entwickelt sich stetig weiter. Daher ist die Begrenzung von Entwicklungstätigkeiten innerhalb von SoS auf ein einzelnes Projekt, wie es in Systementwicklungsprojekten üblich ist, nicht sinnvoll. Dies soll der Begriff Gestaltungskonzept zum Ausdruck bringen. Die Gestaltungskonzepte bündeln Hinweise und Leitlinien, die für bestimmte SoS-Gestaltungsumfänge innerhalb einer Klasse von SoS und einer bestimmten Phase der Gestaltung relevant sind. Gleichzeitig können Gestaltungskonzepte kombiniert oder wiederkehrend angewendet werden, um Veränderungen im SoS zu begleiten.

Die Entwicklung der Gestaltungskonzepte innerhalb des Gestaltungsrahmens erfolgte anhand der Action Design Research (ADR)-Methode nach SEIN et al. im Rahmen des Forschungsprojektes MoSyS (Bild 4-6) [SHP+11]. Das ADR-Team bestand aus Forschenden und Anwendern, die den Aufbau des Gestaltungsrahmens kontinuierlich begleitet haben.

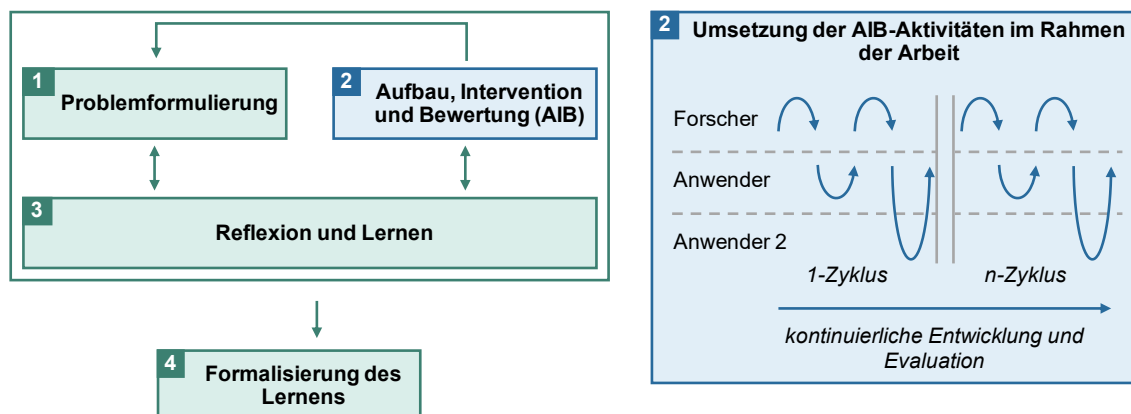


Bild 4-6: Action Design Research nach SEIN et al. und Instanziierung der Methode im Rahmen der Arbeit [SHP+11, S. 41f.]

Die **erste Stufe** des Vorgehens umfasst die Problemdefinition, die bereits in Kapitel 2.3 ausführlich diskutiert wurde. Wesentliche Problemstellung, die sich auch im Forschungskonsortium gezeigt hat, ist der unterschiedliche Bedarf an Gestaltungsansätzen in den unterschiedlichen Ausprägungen von SoS. Dieser wird durch die Analyse der sehr heterogenen Anwendungsbeispiele der Anwender deutlich. Konkret zeigt sich dies u. a. anhand des Beispiels eines produzierenden Unternehmens, welches den Materialfluss innerhalb einer Produktionslinie optimiert. Auch wenn die Fertigungsanlagen selbst von unterschiedlichen Unternehmen bereitgestellt werden, liegt das zentrale Management beim Betreiber der Produktionslinie. Dieser kann die Schnittstellen und die Kommunika-

tion zwischen den Systemen definieren und die Anforderungen an die jeweiligen Unternehmen weitergeben. Dabei tauschen sich die Unternehmen direkt miteinander aus und bestimmen die Interaktion innerhalb der Produktionslinie. Die dafür relevanten Informationen stehen zur Verfügung, wodurch eine optimale Abstimmung der Systeme aufeinander im Rahmen der Entwicklung möglich ist. Im Vergleich dazu wurde das Beispiel „Intelligentes Laden von Fahrzeugen“ analysiert. Dabei werden sowohl die Ladeinfrastruktur als auch die Fahrzeuge von unterschiedlichen Herstellern ohne ein zentrales Management betrieben. Ein Austausch und eine Interaktion zwischen den Systemen muss daher auf definierten Standards basieren, die von den beteiligten Systemen unterstützt werden und von unternehmensübergreifenden Gremien definiert wurden. Während im ersten Beispiel auf konkrete Gestaltungsansätze, bspw. aus dem MBSE oder der Produktionssystementwicklung, zurückgegriffen werden kann, sind diese im zweiten Beispiel nicht sinnvoll anwendbar. Die Anwendung und Nutzung von etablierten Gestaltungsansätze aus der Literatur (vgl. Kapitel 3.2) zeigt daher nur in einzelnen Beispielen einen entsprechenden Mehrwert. Eine bedarfsorientierte Auswahl an Gestaltungsmethoden zur Orientierung für Unternehmen wird benötigt. Diese muss den unterschiedlichen Einfluss der Unternehmen auf das SoS berücksichtigen.

In der **zweiten Stufe** wurde auf Basis der Problemstellung und den ersten Erkenntnissen eine erste initiale Lösung erarbeitet. Hierfür wurde zunächst die Klassifikation von SoS als Ausgangsbasis genutzt (vgl. Kapitel 4.2.2). Die Einordnung der unterschiedlichen SoS Beispielprojekte in die drei unterschiedlichen SoS-Klassen wurde wiederholt anhand der Beispiele der Anwender im Konsortium evaluiert und auf Basis der jeweiligen Erkenntnisse weiterentwickelt. Hierdurch wurden zwei weitere Klassen abgeleitet, die sich aus den unterschiedlichen Bedarfen der Anwendungsbeispiele ergeben haben. Ein Beispiel ist die Gestaltung von SoS aus Unternehmenssicht, die sich aus dem Bedarf ableitet, ein SoS durch eine Eigenentwicklung zu beeinflussen unter der Berücksichtigung von SoS-spezifischen Fragestellungen.

Durch die kontinuierliche Evaluation anhand von ergänzenden Literaturbeispielen, wurde **Stufe drei** erfüllt. Diese umfasst die Erweiterung des Betrachtungsrahmens auf einen größeren Kontext über die bereitgestellten Beispiele der Anwender hinaus. In **Stufe vier** wurde schließlich die Lösung generalisiert und für die weitere Anwendung formalisiert.

Insgesamt wurden sowohl zehn Anwendungsbeispiele aus dem Konsortium als auch zwölf Literaturbeispiele in die Entwicklung des Gestaltungsrahmens mit einbezogen (vgl. Kapitel 4.2.2). Als wesentliche Unterscheidungsdimensionen für die Auswahl von passenden Gestaltungsmethoden wurde zum einen die Managementunabhängigkeit identifiziert, die sich auch in der Klassifikation von SoS widerspiegelt und durch die Literatur gestützt wird. So beschreiben beispielsweise FENG et al., dass der Einflussgrad durch das Management die Zusammenarbeit im Rahmen der Entwicklung in Bezug auf Anforderungen, Schnittstellen, Datenformate und Technologien bestimmt [FZZ+23, S. 627]. Zum anderen wird die Phase der Entwicklung unterschieden, in die sich die Gestaltung einord-

net. Dabei wird in Anlehnung an die Ausarbeitung nach DAHMANN et al. die Initialisierungs- und die Update-Phase unterschieden (vgl. Kapitel 3.2.1.1) [DRL+11]. Im Rahmen der Analyse wurden die Unterscheidungsdimensionen kontinuierlich anhand der Beispiele überprüft. Ein Auszug der Beispiele und deren Bewertung wird in Tabelle 4-2 dargestellt.

Tabelle 4-2: Einordnung der Beispiele

Beispiel	Beschreibung	Mgmt.-unabh.	Konzept-phase
„Flexible Produktion von Objekten“	Übergabe von Teilen und Baugruppen zwischen verschiedenen Fertigungs- und Automatisierungslösungen, organisiert innerhalb einer flexiblen Produktionslinie eines produzierenden Unternehmens	gering	Update
„Smart Charging mit vernetzten Fahrzeugen“	Ladekonzept für E-Fahrzeuge im Verbund mit der Ladeinfrastruktur, Smart Homes und Netzbetreibern in Abhängigkeit der Verfügbarkeit, organisiert durch Gremien	mittel	Update
+ 8 weitere Anwendungsbeispiele	...	...	...
„Flottenfahrt“ nach AXELSSON [Axe19a]	Konzept zur Realisierung einer Flottenfahrt mit sehr geringem Abstand zwischen den beteiligten Fahrzeugen zur Reduzierung des Energieverbrauchs der hinterherfahrenden Fahrzeuge, organisiert in einer Interessensgemeinschaft	mittel	Initialisierung
„Krisenbewältigung“ nach PRAKASHA et al. [PKN+21]	Organisation der beteiligten Systeme in unvorhersagbaren Krisensituationen unter starker Einflussnahme von Menschen, die situativ Entscheidung treffen	hoch	Update
+ 10 weitere Literaturbeispiele	...	...	...

Die Beispiele und Erkenntnisse der Ausarbeitung wurden in insgesamt fünf Gestaltungskonzepte überführt (Bild 4-7). Dabei wurden Gemeinsamkeiten zwischen den Beispielen identifiziert, die sich in dem jeweiligen Gestaltungskonzept wiederfinden. Dazu gehören übergeordnete Ziele und Merkmale, die sich aus den Beispielen und der Literatur ableiten lassen. Zusätzlich wurden Leitlinien auf Basis der Erkenntnisse abgeleitet und für die jeweiligen Konzepte aufbereitet. Dazu gehören Hinweise zu technischen, aber auch zu zusammenhängenden Management-Herausforderungen in den jeweiligen Gestaltungskonzepten.

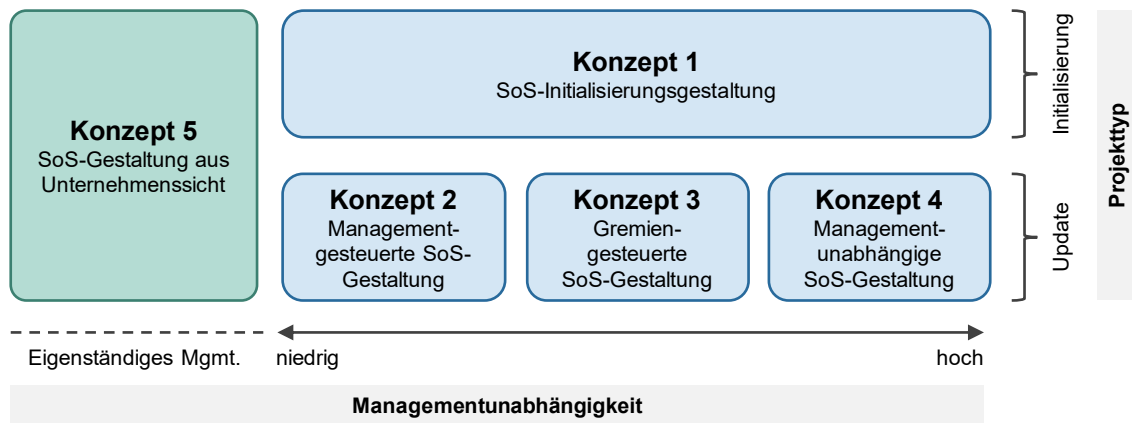


Bild 4-7: Gestaltungskonzepte für SoS

SoS können nicht bis ins letzte Detail gestaltet und optimiert werden. Dies wird auch in unterschiedlichen Literaturquellen, wie bspw. von HESTER, widerspiegelt [Hes12, S. 268]. Daher sind die Leitlinien der jeweiligen Gestaltungskonzepte als Orientierung und Hinweise zu verstehen, die dem Anwender wichtige Aspekte zur Verfügung stellen, die es innerhalb der jeweiligen Gestaltungskonzepte zu beachten gilt. Diese definieren keine abgegrenzten Entwicklungsumfänge, sondern können mehrere zu klärende Fragestellungen beinhalten. Die Fragestellungen können sich durch die stetig verändernden Rahmenbedingungen oder auch Weiterentwicklungen im SoS verändern, sodass ein Wechsel in ein anderes Gestaltungskonzept notwendig wird (siehe Kapitel 4.3.4). Je nach Konzept kann dabei mehr oder weniger Einfluss genommen werden. Im Folgenden werden die Erkenntnisse inklusive der grundlegenden Ziele, Merkmale und Leitlinien der jeweiligen Konzepte zusammengefasst. Dabei wird sowohl auf technische als auch managementbezogene Leitlinien eingegangen. Der Fokus liegt auf der Konzeptentwicklung als Basis für weitere Fragestellungen, wie bspw. Themen, die die konkreten Implementierungen oder die Herausforderungen bei der Verifikation und Validierung betreffen. Die Ausführungen wurden in komprimierter Form im Rahmen einer Vorveröffentlichung publiziert und kommuniziert [GGH+23].

Für einen einfachen Zugang zu den Gestaltungskonzepten, wurden diese in Form von Steckbriefen aufbereitet. Der grundsätzliche Aufbau der Gestaltungskonzeptsteckbriefe wird in Bild 4-8 vorgestellt. In Kurzform werden die wesentlichen Ziele, Leitlinien und Merkmale beschrieben und dem Anwender zur Verfügung gestellt. Neben allgemeinen Merkmalen wird eine Einschätzung der Managementunabhängigkeit, dem potenziellen emergenten Verhalten und der Interoperabilität visualisiert. Passend zu den beschriebenen Aufgaben innerhalb des Gestaltungskonzepts werden Methoden vorgeschlagen, die die Durchführung unterstützen. Neben einer empfohlenen Methode werden auch ergänzende Methoden vorgeschlagen, die je nach Herausforderung und Detailgrad der Ausarbeitung alternativ oder ergänzend durchgeführt werden können. Die Methoden sind Teil der Ausführungen in Kapitel 4.3.2 und werden dort ausführlich vorgestellt. Die Methodenauswahl hat keinen Anspruch auf Vollständigkeit und kann kontinuierlich auf Basis

von entstehenden Herausforderungen und dem sich stetig weiterentwickelnden Stand der Forschung erweitert werden.

ID	Name	
Zielsetzung <i>(Kurzbeschreibung der Zielsetzung)</i>		Merkmale <i>(Allgemeine Merkmale)</i>
Leitlinien <i>(Was sind grundlegende Handlungshinweise, die diesem Gestaltungskonzept zugrunde liegen?)</i> Managementbezogene Hinweise: <i>(Was muss auf organisationaler und strategischer Ebene beachtet werden?)</i> Technische Hinweise: <i>(Was muss bei der technischen Implementierung beachtet werden?)</i> Sonstiges: <i>(Weitere Hinweise)</i>		<i>(Charakteristische SoS-Merkmale inklusive Ausprägung)</i> Managementunabhängigkeit abhängig ————— unabhängig 1 ————— 2 ————— 3 Emergentes Verhalten gering ————— stark 1 ————— 2 ————— 3 Interoperabilität hoch ————— gering 1 ————— 2 ————— 3
Empfohlene Methode [ID] Name der Methode		Alternative / Ergänzende Methoden [ID] Name der Methode [ID] Name der Methode ...

Bild 4-8: Grundsätzlicher Aufbau der Gestaltungskonzeptsteckbriefe

4.3.1.1 Konzept 1: SoS-Initialisierungsgestaltung

Ein SoS ist ein Netzwerk aus verschiedenen Systemen, die sich zusammenfinden, um gemeinsam eine Aufgabe zu lösen. Damit eine Vernetzung und Interaktion stattfinden kann, muss zunächst eine Grundlage geschaffen werden, die durch die Initialisierungsgestaltung initial entwickelt wird. Die Basis ist ein übergeordnetes Ziel, welches durch die Interaktion von Systemen erreicht und durch alle Beteiligten angestrebt wird. Dies ist der Treiber für die Gestaltung und den Betrieb eines SoS. Häufig ist ein Ziel auf die klassischen Faktoren Zeit, Kosten oder Qualität zurückzuführen. Aber auch Nachhaltigkeitsziele spielen immer häufiger eine Rolle (vgl. [INC21-ol, S. 4]). Beispielsweise wird im Rahmen der Flottenfahrt das Ziel verfolgt, eine effiziente und damit ressourcenschonende Fahrt durch die Ausnutzung des Windschattens vom vorausfahrenden Fahrzeug zu realisieren.

Im Rahmen der **SoS-Initialisierungsgestaltung** wird zunächst genau dieses Ziel identifiziert und festgelegt. Ursprung können neue Ideen, veränderte Rahmenbedingungen oder Technologiesprünge sein, wodurch sich neue Möglichkeiten eröffnen. Auch durch die

Analyse des Nutzerverhaltens (Use Cases) und deren Bedarfe können Ziele identifiziert werden, welche die Basis für die Gestaltung darstellen. Die Initialisierungsgestaltung kann durch ein einzelnes Unternehmen gestartet werden, jedoch sollte spätestens bei der Zieldefinition eine Gruppe von Unternehmen involviert sein. Dies ist notwendig, um dem Charakter von SoS zu entsprechen, indem ein übergeordnetes Ziel der Treiber für die Zusammenarbeit darstellt. Dieses sollte nicht primär den Interessen eines einzelnen Unternehmens entsprechen.

Im nächsten Schritt werden die grundlegenden Informationen gesammelt, die für die Erreichung des Ziels notwendig sind. Dazu gehören die notwendigen Fähigkeiten, die möglichen Kernsysteme, die diese Fähigkeiten bereitstellen können, sowie die Nutzerrollen und -erwartungen an das System of Systems (vgl. [DRL+11, S. 214f.]). So werden erste Strukturen und Interaktionsmechanismen des SoS konzipiert, ohne auf konkrete Lösungen einzugehen oder konkrete Anforderungen für das SoS oder einzelne Systeme zu spezifizieren. Hieraus kann eine erste Zusammensetzung von relevanten Unternehmen abgeleitet werden. Wird die Zusammenarbeit mit weiteren Unternehmen notwendig, die nicht bereits in der Initialisierungsgestaltung involviert sind, sollten diese angesprochen und mit integriert werden.

Dabei werden häufig erste Zielkonflikte deutlich, die mit der Zusammenarbeit unterschiedlicher Organisationen zusammenhängen. Im Vordergrund muss dabei das übergeordnete Ziel des SoS stehen. Zielkonflikte, bei denen Ziele nicht gleichzeitig erfüllt werden können, können teilweise durch Veränderungen in den Organisationsstrukturen gelöst werden. Ein Beispiel ist der Zielkonflikt zwischen *größtmöglichem Kundennutzen* und einer möglichst *effiziente Interaktion* zwischen Systemen. Wird bspw. das Ziel verfolgt, einen größtmöglichen Nutzen für Endnutzer zu erzielen, so sollten möglichst viele Systeme von unterschiedlichen Unternehmen im SoS interagieren können. Demgegenüber steht die effiziente Interaktion zwischen Systemen, die sich teilweise nur durch die optimale Abstimmung zwischen bestimmten Systemen erzielen lässt. Dies erfordert eine direkte und enge Kooperation zwischen Unternehmen. In beiden Fällen sollten sich diejenigen Unternehmen zusammenfinden, die dem jeweiligen übergeordneten Ziel mehr Relevanz zuordnen. Die Entscheidung darüber hängt nicht zuletzt mit der Offenlegung von schützenswertem Wissen zusammen, welches nicht mit Konkurrenzunternehmen geteilt werden soll.

Die Initialisierungsgestaltung kann sowohl durch Unternehmen durchgeführt als auch durch Formate wie ein Forschungsprojekt erarbeitet werden. Dabei kommen Unternehmen und Forschungseinrichtungen zusammen, um Ideen zu entwickeln und diese in erste Konzepte und Prototypen zu überführen. Die Ergebnisse können die Ausgangsbasis für die Etablierung eines SoS sein. Ein Beispiel ist die smarte Produktion, die durch zahlreiche Forschungsprojekte initial vorangetrieben und in konkreten Strukturen und Schnittstellenstandards wie OPC UA umgesetzt wurde. Bild 4-9 fasst die wesentlichen Aspekte der SoS-Initialisierungsgestaltung (K1) zusammen.

K1	SoS-Initialisierungsgestaltung		1
Zielsetzung Das Ziel ist die Ideenfindung und Konzipierung von neuen SoS-Lösungen. Diese werden zunächst unabhängig von bestehenden Strukturen und lösungsoffen analysiert.	Leitlinien - Festlegung grundlegender Informationen für den Beginn des SoS einschließlich der Ziele, Hauptnutzer, Nutzerrollen und -erwartungen sowie der Kernsysteme, die die Fähigkeiten unterstützen Managementbezogene Hinweise: - Notwendige Unternehmen identifizieren und Zusammenarbeitsmodelle klären Technische Hinweise: - Initiale Konzipierung der Interaktionsmechanismen und der für den Betrieb wesentlichen Systeme zunächst unabhängig von den kontrollierenden Unternehmen Sonstiges: - Entscheidung über Weiterentwicklung des initialen Konzepts durch Gestaltungskonzept K2, K3 oder K5: - K2: Gestaltung eines geschlossenen SoS durch direkte Kooperation mit Unternehmen - K3: Gestaltung eines offenen SoS innerhalb eines Gremiums - K5: SoS-Gestaltung getrieben durch ein einzelnes Unternehmen	Merkmale - Greenfield-Ansatz - Offene Herangehensweise ohne Einschränkungen durch Unternehmensinteressen Managementunabhängigkeit abhängig ————— unabhängig 1 ————— 2 ————— 3	
Empfohlene Methode M1-1 SoS-Initialisierung		Alternative / Ergänzende Methoden M1-2 Problemraumanalyse M1-3 Innovationswerkstatt (MBSE-Konzept 1) MX-1 Zielanalyse M2-1 Konzipierung von abgrenzbaren SoS M3-1 Rollenbasierte Konzipierung von SoS	

Bild 4-9: Gestaltungskonzeptsteckbrief 1 – SoS-Initialisierungsgestaltung

4.3.1.2 Konzept 2: Managementgesteuerte SoS-Gestaltung

Das Gestaltungskonzept 2 ist in der Update-Phase verortet und legt den Schwerpunkt auf SoS, in denen Unternehmen direkt miteinander kooperieren und gemeinsam die Struktur und das Verhalten des SoS prägen. Durch eine vorangegangene Analyse oder auch Vorgaben aus dem Management, sind bereits Unternehmen festgelegt, die im Gestaltungsprozess miteinander kooperieren. Diese stehen in vertraglichen Beziehungen und haben über diese die gemeinsamen Ziele festgelegt. Daher sind Zielkonflikte ein untergeordnetes Thema.

Im Rahmen der managementgesteuerten SoS-Gestaltung etablieren die Unternehmen ein gemeinsames Entwicklungsteam, welches das SoS ausgestaltet und managt. Das Team besteht aus Entwicklerinnen und Entwicklern der beteiligten Unternehmen oder einem speziell von den Unternehmen beauftragten Team. Es hat die Autorität, Entscheidungen zu treffen und Zielkonflikte zu lösen.

Durch die direkte Zusammenarbeit haben die beteiligten Unternehmen Einfluss auf die einzelnen Systeme und können sie gegebenenfalls anpassen und möglichst optimal aufeinander abstimmen. Falls eine Fähigkeit noch nicht im SoS abgedeckt ist, werden konkrete Anforderungen an ein Unternehmen weitergegeben. Dieses entwickelt ein System mit den entsprechenden Fähigkeiten und stellt dieses dem SoS bereit. Die Umsetzung der Anforderung des SoS obliegt den jeweiligen Unternehmen. Die enge Abstimmung über das SoS-Entwicklungsteam führt dazu, dass bei der Integration des Systems in das SoS nur geringe Emergenzeffekte zu erwarten sind.

Das Gestaltungskonzept 2 ähnelt dabei einer klassischen Systementwicklung. Das SoS kann abgegrenzt werden und es bestehen klare Rahmenbedingungen. Diese setzen sich aus Gesetzesvorgaben, dem Umfeld, in dem das SoS betrieben wird, Technologierestriktionen und den Rahmenbedingungen der beteiligten Unternehmen zusammen. Aus diesem Grund sind auch Emergenzen eher gering, da viele Faktoren und insbesondere das Verhalten und die Ziele der einzelnen Systeme zum großen Teil bekannt sind und in die Gestaltung mit einbezogen werden. Nichtsdestotrotz kann durch die dynamische Vernetzung der Systeme Verhalten entstehen, welches durch weitere Updates angepasst werden muss. Bild 4-10 fasst die wesentlichen Aspekte des Gestaltungskonzepts zusammen.

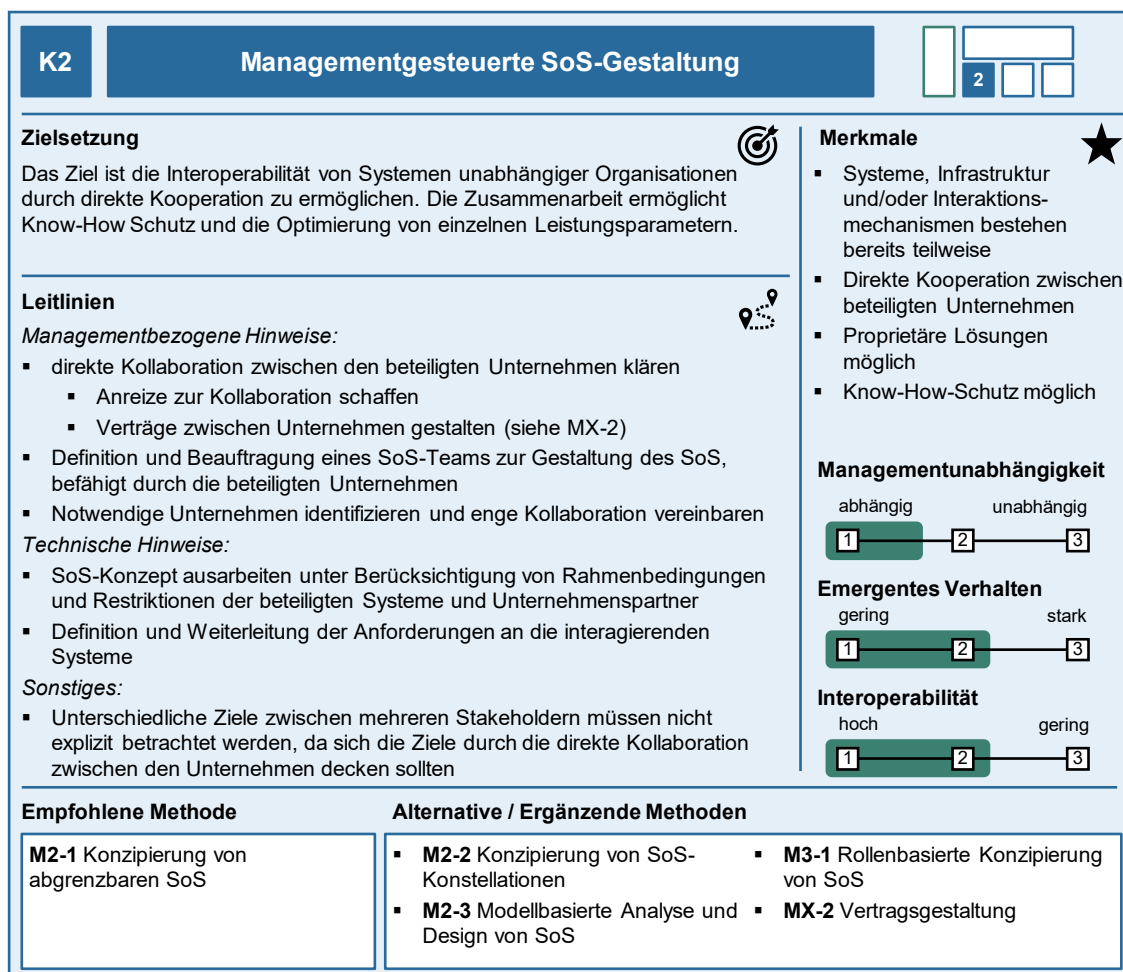


Bild 4-10: Gestaltungskonzeptsteckbrief 2 – Managementgesteuerte SoS-Gestaltung

4.3.1.3 Konzept 3: Gremiengesteuerte SoS-Gestaltung

Die gremiengesteuerte SoS-Gestaltung zeichnet sich durch ein heterogenes Umfeld aus, das gleichzeitig eine große Anzahl potenziell beteiligter Unternehmen umfasst. In diesem Gestaltungsansatz organisieren sich Unternehmen in Interessensgemeinschaften (Gremien), um die Interaktion zwischen heterogenen Systemen zu ermöglichen und übergeordnete Aufgaben zu lösen. Ein Gremium verfolgt das Ziel die Interaktion zu organisieren und strukturell zu unterstützen. In diesem Fall gibt es keine zentrale Autorität, die den einzelnen Systemen ein Verhalten aufzwingen kann (vgl. [Mai15, S. 17]). Systeme agieren freiwillig und müssen durch Anreize für eine Interaktion überzeugt werden.

Um dennoch eine unternehmensübergreifende Interaktion zu ermöglichen, wird das Zusammenspiel der Systeme auf abstrakter Ebene gestaltet. Dabei besteht die Herausforderung, dass die einzelnen Elemente des SoS nicht explizit dargestellt werden können, da diese nicht vollständig vorhanden oder bekannt sind. Außerdem gibt es zahlreiche Systeme, die auf unterschiedliche Art und Weise die Fähigkeiten für das SoS bereitstellen.

Ein Ziel in diesem Gestaltungskonzept ist die Definition von Standards, die die grundsätzliche Interoperabilität zwischen unabhängigen Systemen ermöglichen. Je mehr Unternehmen die jeweiligen Standards unterstützen, umso größer wird der Nutzen. Bei der Auslegung der Standards sollte daher primär auf die Unterstützung von grundlegender Funktionalität geachtet werden, ohne auf zu spezifische Anwendungsfälle von Einzelunternehmen einzugehen. Je mehr Funktionalitäten unterstützt werden, desto größer ist auch der Mehrwert und der Anreiz, den Standard zu nutzen. Daher bietet eine Partizipation von Unternehmen in Interessensgemeinschaften Vorteile für eine möglichst breite Berücksichtigung von Anwendungsfällen.

Die Analyse der Anwendungsfälle ist der Ausgangspunkt für die Ausarbeitung von Standards. Hieraus werden Mindestanforderungen und Rahmenbedingungen abgeleitet, die anschließend in den Standards berücksichtigt werden. Die Spezifikation erfolgt häufig anhand von generischen Rollen- und Interaktionsbeschreibungen, ohne die technischen Lösungen einzuschränken. Ein Beispiel ist der „Matter“-Standard, der die Interoperabilität im Smart Home ermöglicht und dabei den Vorteil für Nutzer in den Vordergrund stellt [ZJK23, S. 2]. Dieser wurde durch ein Konsortium aus unterschiedlichen Unternehmen ausgearbeitet, die teilweise auch in Konkurrenzsituationen stehen. Das Gestaltungskonzept 3 wird in Bild 4-11 zusammengefasst.

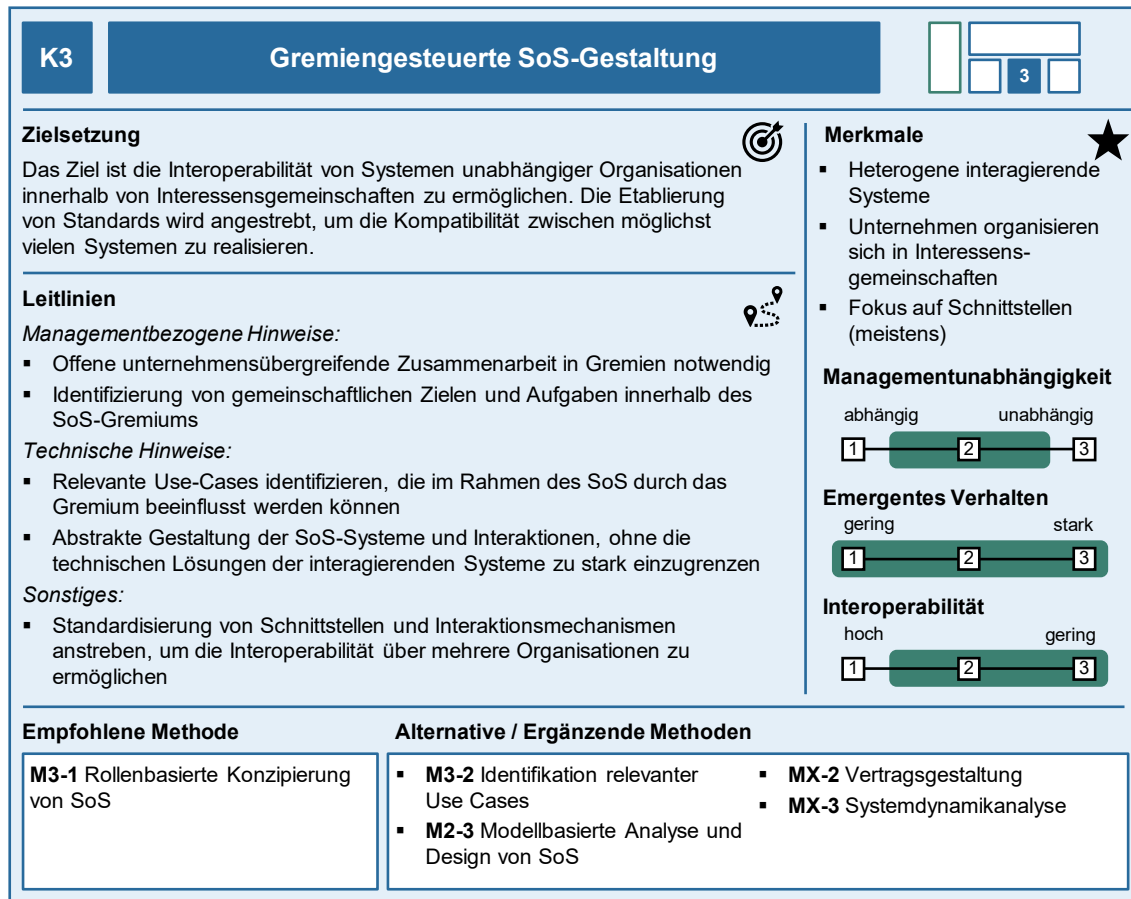


Bild 4-11: Gestaltungskonzeptsteckbrief 3 – Gremiengesteuerte SoS-Gestaltung

4.3.1.4 Konzept 4: Managementunabhängige SoS-Gestaltung

Im Gestaltungsbereich von Konzept 4 gibt es eine Vielzahl an Faktoren, die unbekannt sind. Weder die Systeme des SoS noch die beteiligten Unternehmen sind eindeutig zugeordnet und können im Rahmen dieser Gestaltung abgegrenzt werden. Die Vernetzung ist hochdynamisch und nicht durch übergeordnete Organisationen zu koordinieren. Einzig durch gesetzgebende Maßnahmen kann in diesem Gestaltungsbereich Einfluss ausgeübt werden. Bspw. kann in Form von Verkehrsschildern das Verhalten von Fahrzeugen im Straßenverkehr beeinflusst werden. Dennoch ergibt sich das tatsächliche Verhalten durch die Einzelentscheidungen der Verkehrsteilnehmer.

In diesem Gestaltungskonzept gibt es keinen direkten Adressaten, der die Interaktion innerhalb des SoS vollständig verstehen und beeinflussen kann. Vielmehr können durch unterschiedliche Initiativen einzelne Teilbereiche des SoS beeinflusst werden. So kann ein einzelnes Unternehmen durch eine Systementwicklung das Verhalten innerhalb des SoS verändern. Auch können einzelne Verhaltensweisen von Systemkonstellationen durch Interessensgruppen verändert werden. Diese Art der Gestaltung würde jedoch mit den jeweils anderen Gestaltungskonzepten einhergehen und nicht unter die managementunabhängige Gestaltung fallen. Ein Beispiel ist die Gestaltung des Internets. Hier wurde

als Grundlage das IP-Protokoll im Rahmen einer Interessensgemeinschaft entwickelt, welches die Kommunikation von unabhängigen Systemen ermöglicht [DB08, S. 2].

Zudem ist in diesem Gestaltungsbereich häufig der Mensch ein wesentliches Element, welcher das Verhalten beeinflusst. Die soziotechnischen Aspekte zeigen sich beispielsweise in der Krisenbewältigung nach PRAKASHA et al. [PKN+21]. Dabei koordinieren und steuern Menschen einzelne Systeme innerhalb eines Systemverbunds und sorgen so für eine situative Anpassung der Verhaltensweisen.

Die Analyse der Anwenderbeispiele zeigt, dass diese SoS auch im Rahmen aktueller Fragestellungen eine untergeordnete Rolle spielen, da nur in geringem Maße Einfluss ausgeübt werden kann. Dies wird auch in der Literatur beschrieben. Bspw. beschreibt FENG et al., dass auch die Definition von Zielen bei SoS ohne klare Managementstrukturen an seine Grenzen stößt [FZZ+23, S. 634]. Eine managementunabhängige Gestaltung sollte daher in Teilbereiche zergliedert und innerhalb eines anderen Gestaltungskonzepts betrachtet werden. Bild 4-12 fasst das Gestaltungskonzept 4 zusammen.

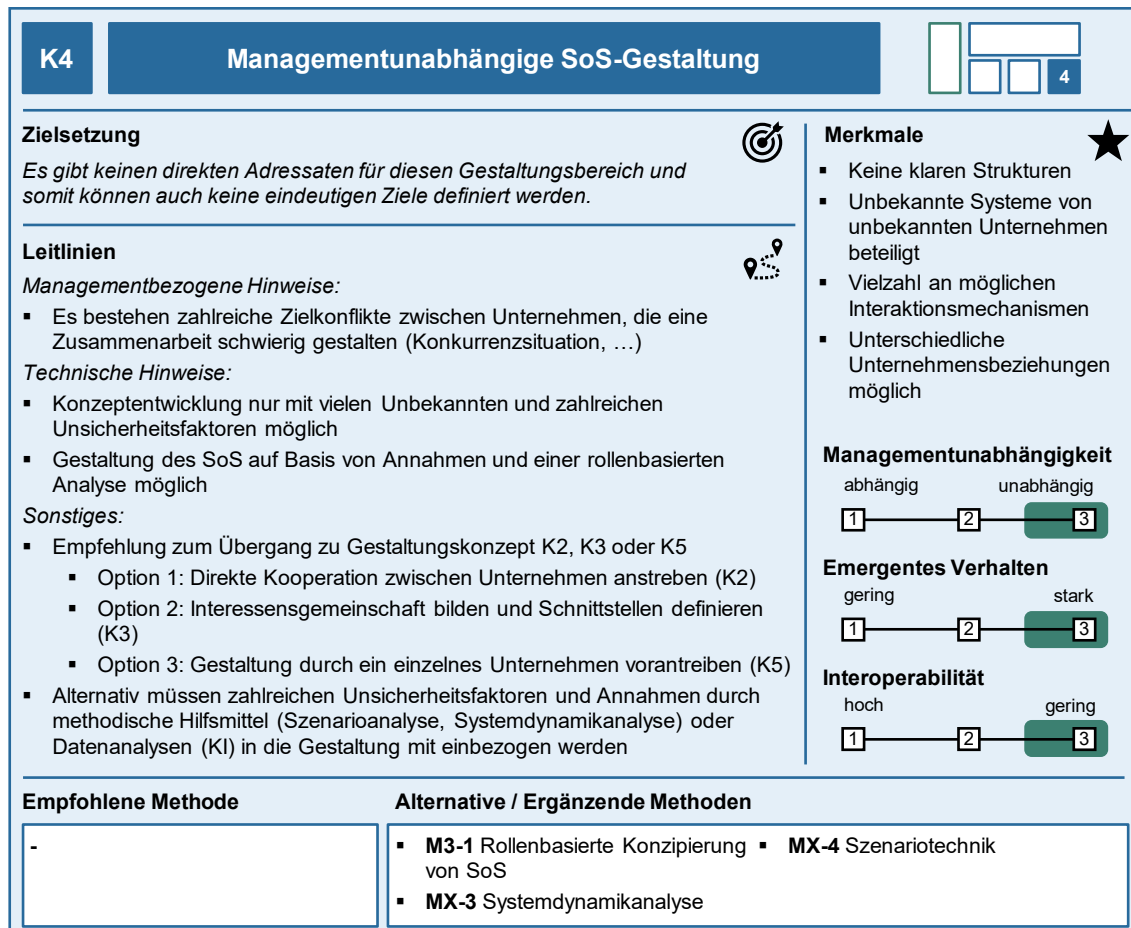


Bild 4-12: Gestaltungskonzeptsteckbrief 4 – Managementunabhängige SoS-Gestaltung

4.3.1.5 Konzept 5: SoS-Gestaltung aus Unternehmenssicht

Strenggenommen stellt das Konzept 5 kein SoS-Gestaltungskonzept dar. Es fokussiert die Unternehmenssicht, die unter dem Einfluss eigener Ziele steht und in diesem Handlungsrahmen agiert. Nichtsdestotrotz ist die Betrachtung der Handlungsoptionen aus Sicht eines Unternehmens wesentlich für die Realisierung von SoS-Konzepten. Daher werden in diesem Gestaltungskonzept unterschiedliche Aspekte zusammengefasst, die ein einzelnes Unternehmen im Kontext von SoS betreffen. Dazu gehören unterschiedliche Handlungsoptionen, die Einfluss auf die Mitwirkung in den zuvor aufgeführten Gestaltungskonzepten haben. Als einzelnes Unternehmen ist der Einflussbereich begrenzt und die Zusammenarbeit mit anderen Unternehmen essenziell, um übergeordnete Ziele zu erreichen. Im Folgenden werden drei Handlungsoptionen aufgeführt, die ein Unternehmen im Kontext von SoS hat.

Ein neues SoS initiieren

Ausgehend von eigenen Ideen und Zielen kann ein Unternehmen die Bildung eines SoS forcieren. Hierzu kann ein Unternehmen zunächst ein erstes grobes Konzept ausarbeiten und notwendige Unternehmen identifizieren, die zur Realisierung des Konzepts notwendig sind. Ist sowohl eine direkte Kooperation mit weiteren Unternehmen als auch die Bildung eines Gremiums möglich, müssen Vor- und Nachteile abgewogen werden. Dazu gehören sowohl die Analyse der eigenen Marktposition und die Betrachtung der Beziehungen zu anderen Unternehmen als auch die Abwägung der Nutzerinteressen, die vom SoS profitieren wollen. Im Anschluss wird auf die entsprechenden Unternehmen zugegangen und das SoS innerhalb des Gestaltungskonzepts 1 ausgearbeitet. Ist die eigene Marktposition mit einer hohen Marktmacht verbunden, kann beispielsweise ein eigenes Konzept forciert werden, dem weitere Unternehmen folgen. Herrscht ein heterogenes Marktumfeld, kann die Bildung einer Interessensgemeinschaft sinnvoll sein. Diese Analyse sollte auch bei einer Anfrage zur Initiierung eines neuen SoS durch ein anderes Unternehmen vorab erfolgen.

Ein SoS mitgestalten

Innerhalb etablierter SoS gilt es für Unternehmen, kontinuierlich mit anderen Unternehmen im Rahmen von Gestaltungskonzept 2 oder 3 zusammenzuarbeiten und das SoS weiterzuentwickeln. Durch eine kontinuierliche Mitarbeit können Weiterentwicklungen mit beeinflusst und insbesondere die Auswirkungen auf die eigenen Systeme zurückgeführt werden. Die Auswirkungen der Entscheidungen innerhalb der SoS-Gestaltung sollten laufend auf das eigene Unternehmen reflektiert werden. Bspw. kann eine potenzielle Veränderung eine Chance sein, indem ein Unternehmen die entsprechend geforderten Fähigkeiten des SoS entwickelt und anschließend durch ein eigenes System anbietet.

Anforderungen eines SoS erfüllen

Neben der eigenen Mitwirkung bei der Gestaltung und Etablierung eines SoS können auch andere Unternehmen und Organisationen die Grundlagen für ein SoS entwickeln

und für die Bildung eines SoS verantwortlich sein. Hieraus resultieren Anforderungen an einzelne Systeme, die durch unterschiedliche Unternehmen und deren Systeme erfüllt werden können. Dies bietet Unternehmen die Chance, durch eine Eigenentwicklung oder Anpassung eines Systems die entsprechenden Anforderungen zu erfüllen und eine Interaktion in diesem Systemverbund zu ermöglichen. Ein Beispiel ist die Erfüllung eines Schnittstellenstandards, der die Interaktion mit Systemen ermöglicht, die den gleichen Standard nutzen.

Unabhängig von den drei Handlungsoptionen bieten sich agile Ansätze für Unternehmen an, um auf Veränderungen im SoS reagieren zu können. Agilität beschreibt die Anpassungsfähigkeit an veränderte Rahmenbedingungen, die insbesondere bei unabhängigeren SoS eine Rolle spielt. Durch Iterationen können Anpassungen am eigenen System durchgeführt werden, um das gewünschte Verhalten im SoS zu erreichen. Auch die Etablierung einer Eng-Ops (Engineering Operations) Umgebung ist sinnvoll, in der das Feedback aus der Nutzungsphase (Operations) von Systemen für Weiterentwicklungen im Engineering genutzt wird.

Wird ein System in unterschiedlichen SoS genutzt, müssen Mechanismen etabliert werden, die den Wechsel zwischen den SoS ermöglichen. Dazu gehören Anpassungen im Verhalten eines Systems, die sich aus veränderten Systemrollen ergeben können. Ein System muss die eigene Rolle erkennen und die entsprechenden Funktionalitäten dieser Rolle freischalten oder sperren können. Ein Beispiel ist ein autonomes Fahrzeug, welches sowohl in einer Flotte als auch einzeln fahren kann. Fährt es in einer Flotte, muss es die Geschwindigkeit entsprechend der voranfahrenden Fahrzeuge anpassen können während in einer Einzelfahrt die Entscheidung selbst getroffen werden muss.

Weitere spezifische Tätigkeiten, die im Rahmen der Systementwicklung innerhalb eines Unternehmens für die Integration in SoS relevant sind, werden bereits in unterschiedlichen Publikationen aufgeführt. Beispiele sind die veränderten Tätigkeiten innerhalb der Verifikation und Validierung im SoS-Kontext, die durch WIECHER et al. beschrieben werden oder die modulare Gestaltung von offenen Systemen nach AZANI [WMG+24], [Aza09]. Auch die ISO 21480 geht auf die unterschiedlichen Tätigkeiten innerhalb der Systementwicklung ein und erweitert die jeweilige Tätigkeit um SoS-spezifische Aspekte (vgl. Kapitel 3.2.1.8) [ISO21840]. Leitlinien und Hinweise mit Bezug zu Gestaltungskonzept 5 werden in Bild 4-13 zusammengefasst.

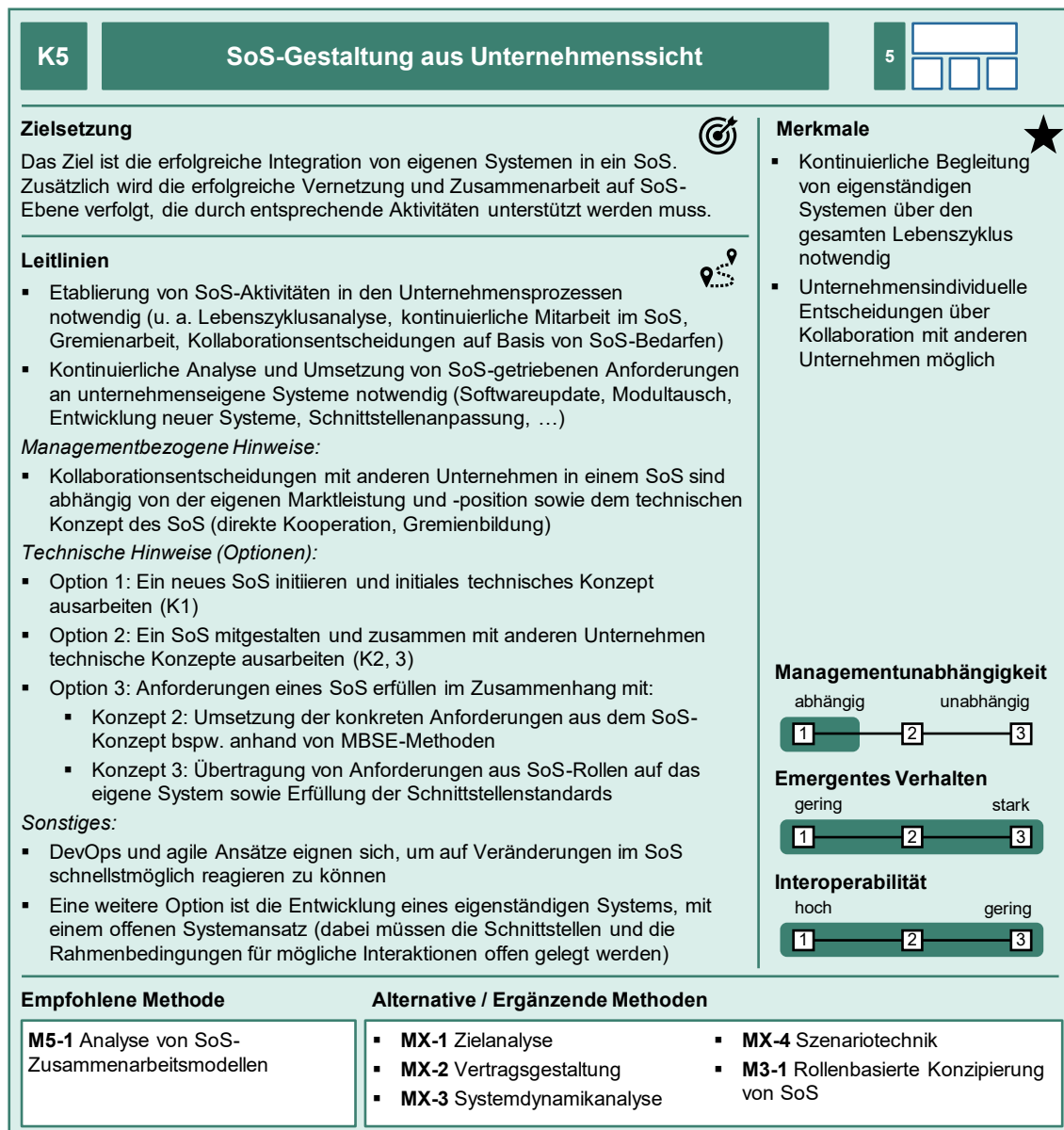
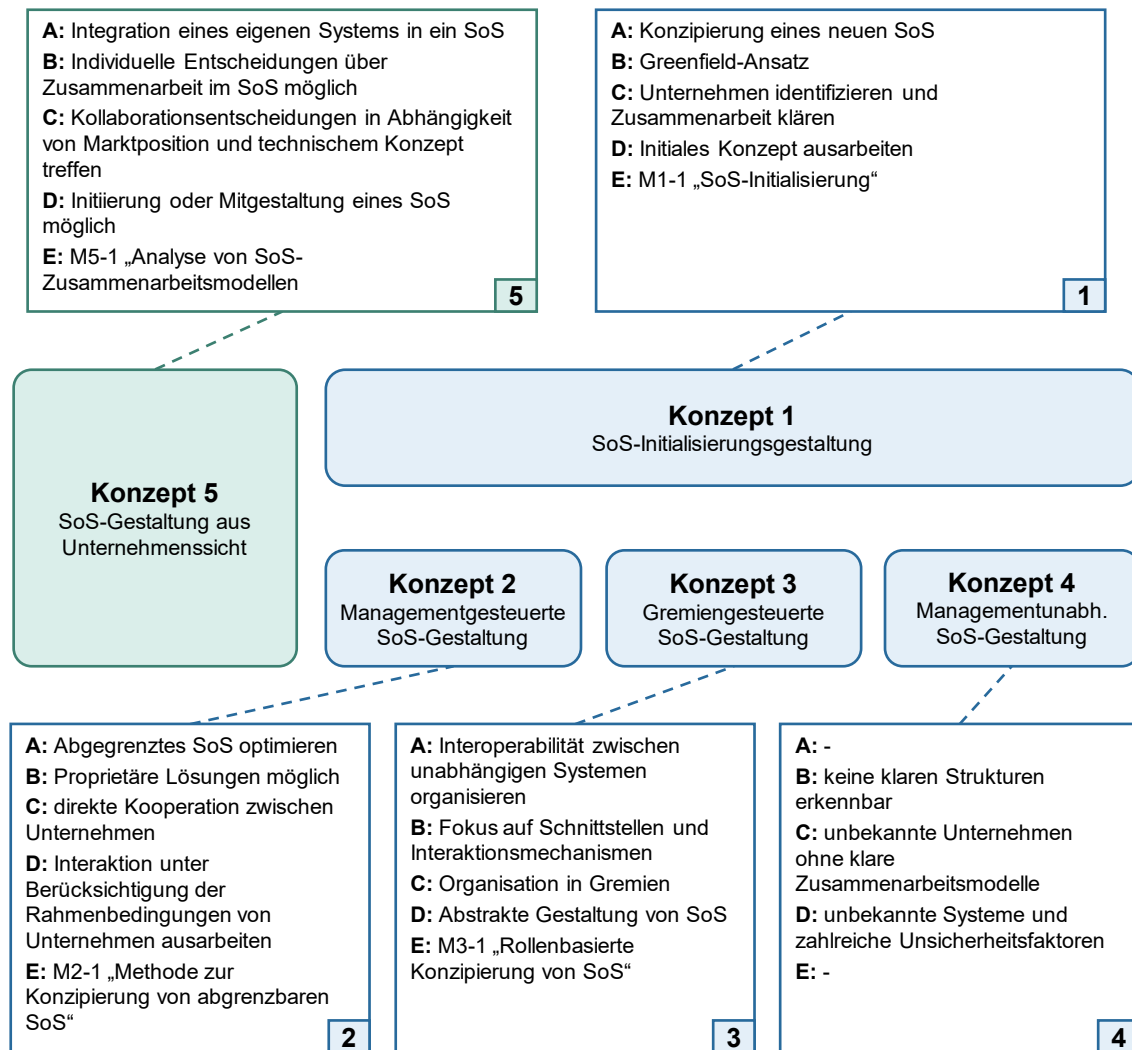


Bild 4-13: Gestaltungskonzeptsteckbrief 5 – SoS-Gestaltung aus Unternehmenssicht

4.3.1.6 Zusammenfassung der Gestaltungskonzepte

Insgesamt stellen die fünf Gestaltungskonzepte unterschiedliche Herangehensweisen für die Gestaltung von SoS dar. Sie hängen von den einzelnen Rahmenbedingungen der Unternehmens- und Systemstrukturen ab. Für die Gestaltung werden managementrelevante und technische Maßnahmen in einen Zusammenhang gebracht und auf die jeweiligen Rahmenbedingungen des SoS-Typs bezogen. Bild 4-14 fasst die wesentlichen Merkmale und Hinweise der Gestaltungskonzepte zusammen und gibt einen konsolidierten Überblick über die fünf Gestaltungskonzepte.

**Legende**

A = Ziel, B = Merkmale, C = Managementbezogene Hinweise, D = Technische Hinweise, E = Empfohlene Methode

Bild 4-14: Übersicht der wesentlichen Merkmale und Hinweise der einzelnen Gestaltungskonzepte

4.3.2 Methoden für die Gestaltung von SoS

Um die Fragestellungen innerhalb der Gestaltungskonzepte zu bearbeiten, werden Methoden benötigt. Sie beschreiben, wie Aufgaben erfüllt werden können [Est08, S. 2]. Im Rahmen des Systems Engineering und SoSE sind zahlreiche Methoden vorhanden, die unterschiedliche Analyse- und Synthese-Tätigkeiten unterstützen (vgl. Kapitel 2 und 3). ESTEFAN oder auch CZAJA stellen beispielsweise zahlreiche Methoden im Kontext des Systems Engineerings vor [Est08], [Cza19, S. 123].

Im Rahmen dieser Arbeit wurden ausgewählte Methoden gesammelt und in Form von Steckbriefen aufbereitet. Diese bieten Anwendern einen einfachen Zugang und stellen die wichtigsten Informationen der Methoden komprimiert zur Verfügung. Die Methodenauswahl erhebt keinen Anspruch auf Vollständigkeit. Vielmehr ist sie durch die einheitliche

Strukturierung erweiterbar und kann insbesondere infolge weiterer Methodenentwicklungen und dem stetig wachsenden Stand der Forschung ergänzt werden.

Relevant für die Auswahl sind die Ziele und Merkmale der einzelnen Gestaltungskonzepte und den damit einhergehenden Aufgaben. Aus diesem Grund umfasst der Steckbrief eine Zuordnungsmöglichkeit zu den SoS-Gestaltungskonzepten. Bild 4-15 zeigt einen Methodensteckbrief am Beispiel der *SoS-Initialisierung* in Anlehnung an DAHMANN et al. [DRL+11, S. 214f.]. Angelehnt an den Problemlösungszyklus nach HABERFELLNER et al. orientieren sich die Aktivitäten im Steckbrief an der Einordnung in problemorientierte und lösungsorientierte Aufgaben [HWF+19, S. 44]. Die Einordnung gibt zusätzlich Hinweise auf zusammenhängende Methoden. In Anlehnung an das Vorgehen nach SHAKED und REICH kann das Ergebnis einer vorangegangenen Methode die Problemanalyse einer nachfolgenden Methode unterstützen (vgl. Kapitel 3.2.1.5). So können die ausgewählten Methoden miteinander kombiniert und um zusätzliche Methoden erweitert werden.

M1-1	SoS-Initialisierung	Zuordnung zu den SoS-Gestaltungskonzepten:	1
Zielsetzung (<i>Was soll mit der Methode erreicht werden?</i>) Ziel ist die Ideenfindung und initiale Konzipierung einer neuen SoS-Lösung. Das initiale Konzept legt die grundlegenden Inhalte für den Start eines SoS fest. Es wird textuell beschrieben.		Beschreibung (<i>Was ist der Hintergrund der Methode und wie wird das Ziel erreicht?</i>) ▪ Zunächst wird die Ausgangssituation betrachtet und relevante Ziele, Stakeholder und Erwartungen analysiert ▪ Hieraus werden notwendige Fähigkeiten des SoS und die Kernsysteme zur Umsetzung abgeleitet ▪ Zusammenfassend wird das grundlegende Konzept beschrieben	
Aktivitäten (<i>Welche Schritte sind durchzuführen?</i>) <div style="display: flex; align-items: center;"> <div style="width: 30px; height: 100px; position: relative; margin-right: 10px;"> <div 2"="" style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; background: linear-gradient(to bottom, #007bff 49%, #007bff 49% 51%, #007bff 51% 51%, #007bff 51% 53%, #007bff 53% 53%, #007bff 53% 55%, #007bff 55% 55%, #007bff 55% 57%, #007bff 57% 57%, #007bff 57% 59%, #007bff 59% 59%, #007bff 59% 61%, #007bff 61% 61%, #007bff 61% 63%, #007bff 63% 63%, #007bff 63% 65%, #007bff 65% 65%, #007bff 65% 67%, #007bff 67% 67%, #007bff 67% 69%, #007bff 69% 69%, #007bff 69% 71%, #007bff 71% 71%, #007bff 71% 73%, #007bff 73% 73%, #007bff 73% 75%, #007bff 75% 75%, #007bff 75% 77%, #007bff 77% 77%, #007bff 77% 79%, #007bff 79% 79%, #007bff 79% 81%, #007bff 81% 81%, #007bff 81% 83%, #007bff 83% 83%, #007bff 83% 85%, #007bff 85% 85%, #007bff 85% 87%, #007bff 87% 87%, #007bff 87% 89%, #007bff 89% 89%, #007bff 89% 91%, #007bff 91% 91%, #007bff 91% 93%, #007bff 93% 93%, #007bff 93% 95%, #007bff 95% 95%, #007bff 95% 97%, #007bff 97% 97%, #007bff 97% 99%, #007bff 99% 99%, #007bff 99% 100%); </div> <div> <p>Problemorientiert</p> ▪ Ziele ermitteln ▪ Hauptnutzer und Stakeholder ermitteln ▪ Nutzerrollen und -bedarfe ermitteln ▪ Kernsysteme zur Umsetzung der notwendigen Fähigkeiten identifizieren und festlegen ▪ Umsetzung planen <p>Lösungsorientiert</p> </div> </div> </td> <td colspan="> Hilfsmittel (<i>Welche Hilfsmittel, Sprachelemente oder andere Werkzeuge helfen bei der Anwendung der Methode?</i>) - </div></div></div>			
Literatur (<i>Hintergrund und weiterführende Literatur zu der Methode oder einzelnen Methodenbausteinen</i>) ▪ Dahmann, J.; Rebovich, G.; Lane, J.; Lowry, R.; Baldwin, K.: An implementers' view of systems engineering for systems of systems: IEEE International Systems Conference (SysCon). 04.04.2011 - 07.04.2011, Montreal, QC, Canada, IEEE, 2011, S. 212–217			

Bild 4-15: Methodensteckbrief am Beispiel „SoS-Initialisierung“ in Anlehnung an DAHMANN et al. [DRL+11, S. 214f.]

Aufgrund der Charakteristika von SoS kann es im Rahmen der SoS-Gestaltung vorkommen, dass Informationen für die Durchführung einer Methode fehlen. Dies hängt nicht zuletzt mit der unternehmensübergreifenden Zusammenarbeit zusammen. In diesen Fällen können ergänzende Methoden, wie bspw. das Kausalschleifendiagramm nach LANE, unterstützen, um bestimmte Zusammenhänge im SoS-Verhalten zu identifizieren (siehe Kapitel 3.2.2.4) [Lan08, S. 9f.]. Häufig werden aber auch Iterationsschleifen notwendig, in denen das SoS im Betrieb getestet und kontinuierlich verbessert wird. Dabei wird auf

Felddaten zurückgegriffen, die Rückschlüsse für die nächste Iterationsschleife bieten. Durch agile Methoden (bspw. Scrum) können diese Entwicklungsvorgänge unterstützt werden. Solche allgemeinen, ergänzenden oder auch konzeptübergreifenden Methoden werden im Rahmen der Arbeit mit einem X in der Steckbriefkennzeichnung aufgeführt. Tabelle 4-3 fasst die Gestaltungsmethoden für SoS zusammen. Details zu den einzelnen Methoden in Form von Methodensteckbriefen sind dem Anhang A2.1 zu entnehmen.

Tabelle 4-3: Methoden zur Gestaltung von SoS

ID	Name	Ziel	Konzepte				
			1	2	3	4	5
M1-1	SoS-Initialisierung	Festlegen der grundlegenden Informationen für den Start des Gestaltungsprozesses inkl. Ziele, Hauptnutzer und Kernsysteme (nach Dahmann et al. [DRL+11a], siehe Kapitel 3.2.1.1)	X	(X)	(X)		
M1-2	Problemraumanalyse	Initiale Analyse der Treiber, Stakeholder, Anwendungsfälle und des Umfelds (siehe Anhang A2.1)	X	(X)	(X)		
M1-3	Innovationswerkstatt (MBSE-Konzept 1)	Unterstützung für die Ideenfindung und Planen und Klären der Aufgaben in Systementwicklungsprojekten (nach TSCHIRNER [TSC16], siehe Kapitel 2.3.3)	X	(X)	(X)		
M2-1	Konzipierung von abgrenzbaren SoS	Konzeptentwicklung für interagierende Systeme in direkten Unternehmenskooperationen (siehe Kapitel 4.3.2.1)		X	(X)		
M2-2	Konzipierung von SoS-Konstellationen	Analyse von SoS-Konstellationen durch das Einbeziehen von unterschiedlichen Hierarchieebenen (nach AXELSSON [Axe19a], siehe Anhang A2.1)		X	(X)		
M2-3	Modellbasierte Analyse und Design von SoS	Modellbasierte Entwicklung von SoS auf Basis eines einheitlichen Sprachmodells (nach BAEK et al. [BMS+20], siehe Kapitel 3.2.1.6)		X	(X)		
M3-1	Rollenbasierte Konzipierung von SoS	Lösungsneutrale Konzipierung der Interaktion unabhängiger Systeme in Systemverbünden (siehe Kapitel 4.3.2.2)		(X)	X		
M3-2	Identifikation relevanter Use Cases	Reduzierung der Vielzahl an möglichen Anwendungsfällen von offenen SoS (siehe Anhang A2.1)		(X)	X		
M5-1	Analyse von SoS-Zusammenarbeitsmodellen	Identifikation der Rolle eines Unternehmens im SoS und Ableitung von Zusammenarbeitsmodellen (siehe Kapitel 4.3.2.3)					X
M5-2	OOSEM	Methode zur Unterstützung der Analyse, des Designs und der Verifikation von Systemarchitekturen (nach INCOSE [INC15, S. 193 ff.], siehe Anhang A2.1)					X
MX-1	Zielanalyse	Methode zur Analyse von Abhängigkeiten zwischen Zielen (nach BÖHM et al. [BBK+21], siehe Anhang A2.1)	(X)	(X)	(X)	(X)	(X)
MX-2	Vertragsgestaltung	Methode zur Ausgestaltung von Verträgen (nach FALDIK et al. [FPF+17], vgl. Kapitel 3.3.1.4)		(X)	(X)		
MX-3	Systemdynamikanalyse	Analyse von dynamischem Verhalten in Verbindung mit Systemstrukturen (nach LANE [LAN08], vgl. Kapitel 3.2.2.4)	(X)	(X)	(X)	(X)	(X)
MX-4	Szenariotechnik	Methode zur Analyse von Zukunftsszenarien auf Basis von Einflussfaktoren (nach Gausemeier et al. [GDE+19, S. 123], siehe Anhang A2.1)	(X)	(X)	(X)	(X)	(X)

Neben ausgewählten Methoden, u. a. aus dem Stand der Forschung, wird zusätzlich je Gestaltungskonzept eine spezifische Methode zur Verfügung gestellt. Diese wurden in Anlehnung an das Vorgehen für die Entwicklung situativer Methoden nach BRINKKEMPER et al. aus bestehenden Methodenbausteinen etablierter Methoden zusammengesetzt [Bri96, S. 277ff.]. Dem Grundgedanken des Problemlösungszyklus folgend setzen sie sich aus Analyse- und Synthese-Tätigkeiten zusammen. Die in den Gestaltungskonzepten beschriebenen Merkmale gelten als Ausgangsbasis für die Auswahl passender Methodenbausteine. Im Rahmen des Forschungsprojektes MoSyS wurden die entwickelten Methoden in Beispielen angewandt und bei Bedarf angepasst. Neben der in Bild 4-15 visualisierten Methode für Gestaltungskonzept 1 wurden zusätzlich Methoden für die Konzepte 2, 3 und 5 entwickelt. Für Gestaltungskonzept 4 gibt es keinen direkten Adressaten, so dass keine Gestaltungsmethode entwickelt wurde.

Da MBSE im Rahmen von SoS noch ein unausgereiftes Thema ist, werden die Vorteile von Modellen nur in Teilen in den Methodensteckbriefen aufgegriffen [NLF+15, S. 4]. In den jeweiligen Fällen werden zusätzliche Hilfsmittel, wie Sprachelemente, für die modellbasierte Durchführung der Methoden bereitgestellt. Grundsätzlich sollen die Kommunikation und Entscheidungen zwischen alternativen Lösungswegen im Rahmen der SoS-Gestaltung unterstützt und semi-formale Ansätze angestrebt werden. Im Folgenden werden die spezifisch für die jeweiligen Gestaltungskonzepte entwickelten Methoden kurz vorgestellt.

4.3.2.1 Konzipierung von abgrenzbaren SoS (Konzept 2)

Im Rahmen von Gestaltungskonzept 2 wird ein SoS durch die direkte Kooperation von Unternehmen gestaltet. Die beteiligten Unternehmen und Systeme sind bekannt und können größtenteils abgegrenzt werden. Daher bieten sich Methoden aus dem klassischen Systems Engineering an, die die Abgrenzung von Systemen als Grundkonzept voraussetzen.

Die Methode zur Konzipierung von abgrenzbaren SoS baut auf den entsprechenden Methodenbausteinen aus etablierten SE-Methoden auf und ergänzt diese durch SoS-spezifische Aktivitäten, wie der Analyse der beteiligten Systeme und Unternehmen im Rahmen der Feldanalyse (Bild 4-16). Eine Zielkonfliktanalyse wird hier nicht explizit integriert, da die Ziele innerhalb der Kooperation bereits im Vorhinein geklärt sein sollten. Die Analysemethode kann jedoch im Rahmen des Gestaltungskonzepts 2 als ergänzende Methode eingesetzt werden, um das Verständnis über die Ziele und Rahmenbedingungen der bereits bestehenden Kooperation zu erhöhen.

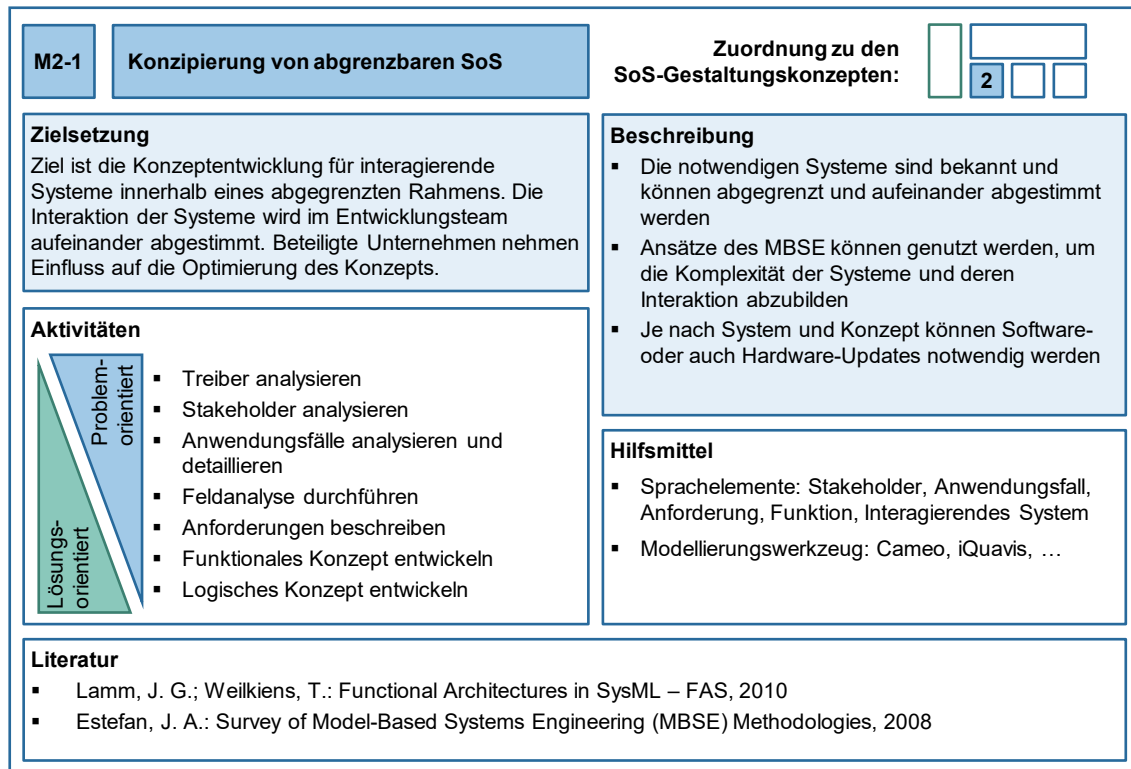


Bild 4-16: Methode „Konzipierung von abgrenzbaren SoS“

4.3.2.2 Rollenbasierte Konzipierung von SoS (Konzept 3)

In managementunabhängigen SoS können unterschiedliche Systeme miteinander interagieren, die unterschiedliche technische Ausprägungen haben. Diese sind häufig nicht bekannt oder werden von Unternehmen geheim gehalten. Zusätzlich sorgt die Vielzahl an möglichen Systemen und Emergenzen dafür, dass diese nicht explizit beschrieben werden können.

Die rollenbasierte Konzipierung von SoS, wie sie u. a. DRIDI et al. beschreiben, adressiert insbesondere die Merkmale der Heterogenität und Interoperabilität von interagierenden Systemen in SoS [DBB20, S. 141]. Rollen sind Abstraktionen von interagierenden Systemen und nehmen Bezug auf das Verhalten innerhalb der Interaktion. Rollen besitzen Fähigkeiten, die zur Erreichung eines übergeordneten Ziels im SoS benötigt werden. In Anlehnung an WOOD und DELOACH baut die Methode auf den Zielen und den Anwendungsfällen des SoS auf und leitet daraus die entsprechenden Rollen und Fähigkeiten ab (Bild 4-17) (vgl. Kapitel 3.2.2.3) [WD01, S. 209].

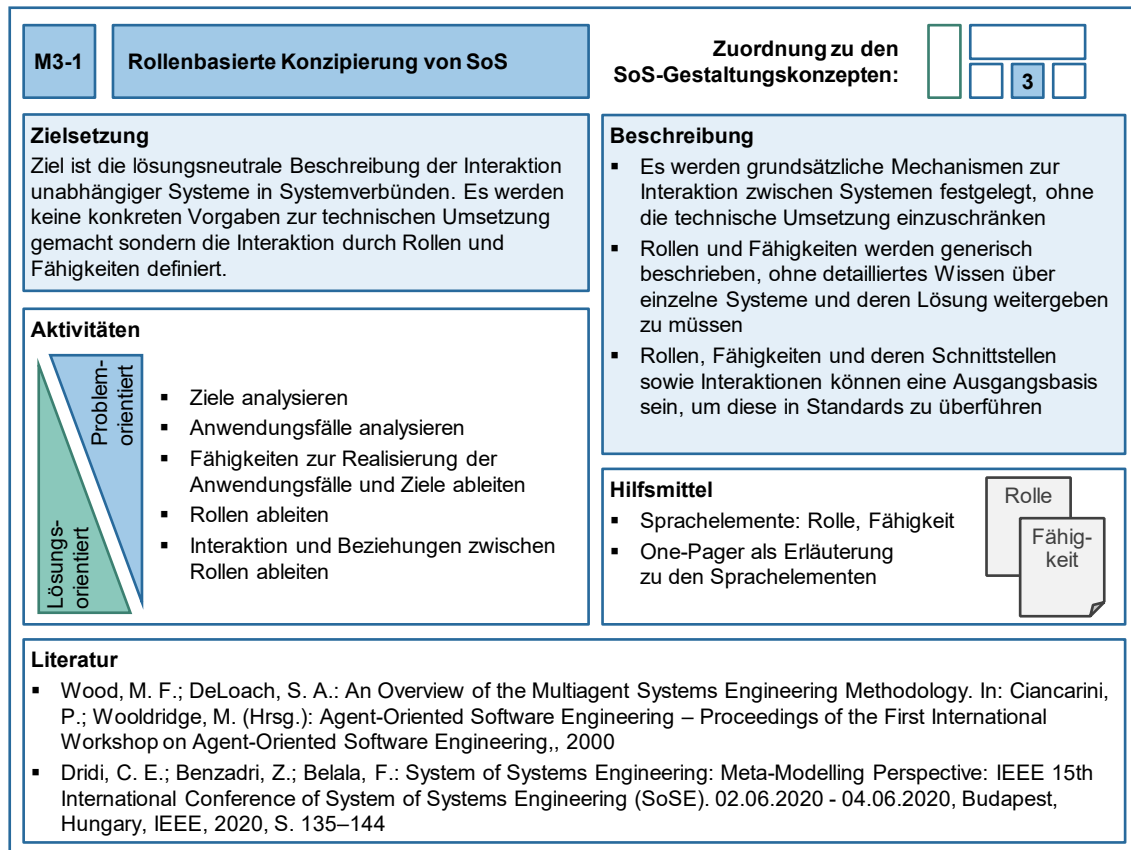


Bild 4-17: Methode „Rollenbasierte Konzipierung von SoS“

4.3.2.3 Analyse von SoS-Zusammenarbeitsmodellen (Konzept 5)

Um erfolgreich in SoS zu agieren, müssen Unternehmen die Betrachtungsgrenze erweitern und Entscheidungen hinsichtlich der Zusammenarbeit mit weiteren Unternehmen treffen. Entsprechend der Handlungsoptionen im Gestaltungskonzept 5 können Unternehmen ein neues SoS im Zusammenspiel mit weiteren Unternehmen forcieren, ein SoS im Rahmen von Kooperationen mitgestalten oder auf Anforderungen von bestehenden SoS reagieren und diese erfüllen, ohne eine aktive Rolle in der Gestaltung des SoS einzunehmen. Je nachdem wie sich das Marktumfeld und die Zusammensetzung des SoS gestaltet, können direkte Kooperationen mit Unternehmen oder auch die Mitarbeit in Interessensgruppen sinnvoll sein.

Die Methode zur Identifizierung von SoS-Zusammenarbeitsmodellen soll Unternehmen dabei helfen, in Abhängigkeit des SoS-Konzepts und der eigenen Marktposition die Beteiligung innerhalb der Gestaltungskonzepte 1 bis 3 zu planen (Bild 4-18). Besteht eine neue Idee und die Marktposition ist ausreichend groß, kann beispielsweise durch eine direkte Kooperation mit einem oder mehreren Unternehmen ein SoS initiiert und im Rahmen von Gestaltungskonzept 2 gemanagt werden. Dies könnte die Marktmacht ausbauen, indem proprietäre Lösungen entwickelt werden und der Zugang zum Markt durch andere Unternehmen erschwert wird. Wird bei der Durchführung der Methode deutlich, dass die

Beteiligung innerhalb einer bestehenden Interessensgemeinschaft sinnvoll ist, dann sollte entweder eine neue Interessensgemeinschaft gegründet oder innerhalb einer bestehenden Interessensgemeinschaft das SoS-Konzept weiterentwickelt werden. Diese Entscheidungen sind auch abhängig von der strategischen Ausrichtung eines Unternehmens und den damit einhergehenden Managemententscheidungen. Hierzu können Methoden der strategischen Produktplanung eingesetzt werden [GDE+19, S. 97ff.].

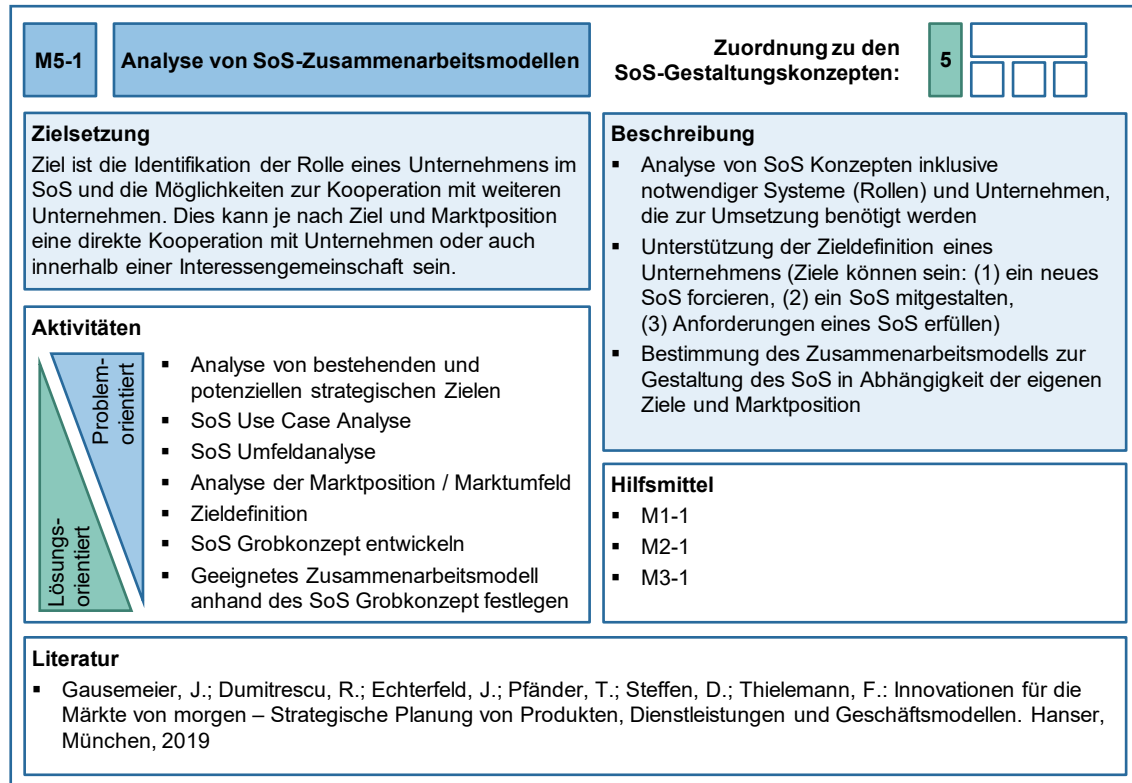


Bild 4-18: Methode zur „Analyse von SoS-Zusammenarbeitsmodellen“

4.3.3 Vorgehensmodell zur Gestaltung von SoS

Im Folgenden wird ein Vorgehen zur Anwendung des Gestaltungsrahmens vorgestellt. Dieses soll es Unternehmen ermöglichen, ein passendes Gestaltungskonzept zu identifizieren und ein SoS entsprechend den dort beschriebenen Leitlinien und Methoden zu gestalten. Hierzu wurde ein Vorgehensmodell entwickelt, welches den Anwender durch drei Phasen führt (Bild 4-19). Ausgangsbasis ist eine neue Idee oder ein zuvor ermittelter Bedarf für ein neues SoS. Die Idee oder der Bedarf kann systematisch durch Methoden der strategischen Produktplanung entwickelt oder durch Erkenntnisse aus dem Betrieb bereits bestehender SoS bzw. aus dem Bedarf von Stakeholdern abgeleitet werden. In der ersten Phase wird die Ausgangsbasis analysiert und in Form eines Grobkonzepts festgehalten. Die zweite Phase dient der Einordnung und Bewertung des Grobkonzepts zur Identifizierung eines passenden Gestaltungsansatzes. Dieser wird in Phase 3 entsprechend angewendet und umgesetzt. Das Vorgehen wird durch mehrere Hilfsmittel unterstützt, die im

Folgenden erläutert werden. Diese bauen auf den Erkenntnissen der vorherigen Kapitel auf.

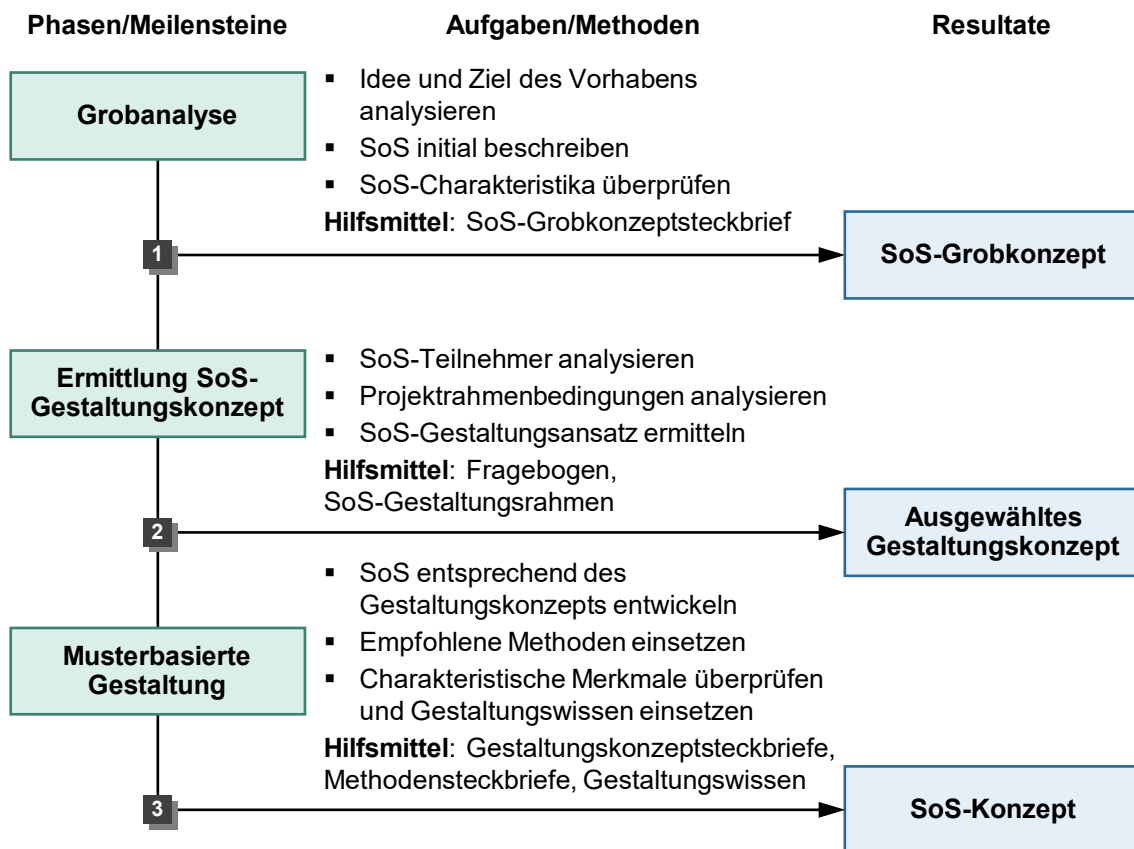


Bild 4-19: Vorgehensmodell zur Gestaltung von SoS

In **Phase 1** wird die Ausgangsbasis analysiert. Die initiale Idee und das Ziel des Vorhabens werden in der Regel aus Vorprojekten, Forschungsarbeiten, Gremiendiskussionen oder durch Produktmanager abgeleitet. Um die wichtigsten Inhalte zu strukturieren, wird die Ausgangsbasis zunächst in Form eines Steckbriefs festgehalten (Bild 4-20). Der SoS-Grobkonzeptsteckbrief umfasst vier Bereiche. In der **Kurzbeschreibung** können die wesentlichen Ziele und Anwendungsfälle textuell festgehalten werden. Die **Einordnung in SoS-Netzwerke** dient der Umfeldanalyse und der ersten Eingrenzung des Betrachtungsbereichs. Innerhalb des **funktionalen Konzepts** werden die relevanten Funktionalitäten aus den zuvor genannten Anwendungsfällen abgeleitet und wenn möglich in Beziehung gesetzt. Dabei sollten sie auf die wesentlichen Funktionen beschränkt werden, die für die Umsetzung des SoS-Zwecks erforderlich sind. Die Beziehungen geben Aufschlüsse darüber, welche Rollen oder Systeme im SoS benötigt werden. Das **logische Konzept** erfasst die Rollen oder Systeme und setzt diese in Beziehung. Hieraus wird deutlich, ob bereits Systeme existieren, die die jeweiligen Funktionen realisieren können oder ob neue Systeme entwickelt werden müssen, um das Konzept umzusetzen. Außerdem kann anhand dieser Überlegungen abgeschätzt werden, welche Unternehmen an der Umsetzung beteiligt sein müssen. Die gesammelten Informationen werden genutzt, um anhand von **drei**

Fragen die Einstufung als SoS zu überprüfen. Die Fragen adressieren die wesentlichen Kriterien, die ein Systemprojekt von einem SoS-Projekt differenzieren. Diese sind nach MAIER die Managementunabhängigkeit, die operative Unabhängigkeit und das emergente Verhalten [Mai15, S. 13]. Sind alle drei Fragen mit „Ja“ beantwortet, wird der Übergang zu Phase 2 des Vorgehensmodells empfohlen. Ist eine Frage mit „Nein“ beantwortet, tendiert das Projekt zu einem Systementwicklungsprojekt und sollte durch Entwicklungsansätze, wie sie bspw. durch die ISO 15288 beschrieben sind, durchgeführt werden (vgl. Kapitel 2.3.1).

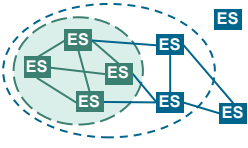
SoS-Grobkonzept Name Kurzbeschreibung (Welche Ziele hat das SoS? Welche Anwendungsfälle werden realisiert? ...) Ziele ▪ ... Anwendungsfälle ▪ ...	Funktionales Konzept (Welche Funktionalitäten werden benötigt und wie hängen diese zusammen?)
Einordnung in SoS-Netzwerke (Welcher Kontext ist zu betrachten? Welche Systeme und SoS sind von Interesse?)   Zusammenhängende SoS ▪ ...  SoS von Interesse ▪ ...  Eigenständige Systeme ▪ ...	Logisches Konzept (Welche eigenständigen Systeme sind beteiligt und wie stehen diese in Beziehung?) 1. Werden die beteiligten Systeme von unterschiedlichen Unternehmen betrieben oder gemanagt? ☐ Ja ☐ Nein 2. Haben mindestens zwei der beteiligten Systeme einen eigenen Zweck und bieten einen Mehrwert außerhalb des SoS-Konzepts? ☐ Ja ☐ Nein 3. Wird ein Mehrwert durch die Interaktion der Systeme erzielt, der nicht durch einzelne Systeme erreicht werden kann? ☐ Ja ☐ Nein

Bild 4-20: SoS-Grobkonzeptsteckbrief

Phase 2 des Vorgehensmodells dient der Auswahl eines passenden Gestaltungskonzepts. Dazu wird der Anwender anhand eines Fragebaums zu einem passenden Gestaltungskonzept geleitet. Die Fragen geben die Möglichkeit, das Grobkonzept zu reflektieren und durch vorgegebene Antwortmöglichkeiten den Anwender zu einem passenden Gestaltungskonzept zu lenken. Insgesamt wurden sieben Fragen entwickelt, die im Folgenden aufgelistet und kurz erläutert werden:

- **Frage 1:** Sind bereits wesentliche Strukturen zur Umsetzung des Grobkonzepts festgelegt? (Ja | Nein)

Ergänzende Informationen zu Frage 1: Als wesentliche Strukturen werden bestehende Systeme und die relevante Infrastruktur aufgefasst. Ist ein Großteil der notwendigen Systeme bereits etabliert und verfügt über die notwendige Kommunikationsstrukturen, so kann auf diese aufgebaut werden und die Frage ist mit „Ja“ zu beantworten.

- **Frage 2:** Ist das SoS bereits in Betrieb oder bestehen bereits wesentliche Interaktionsmechanismen für das SoS? (Ja | Nein)

Ergänzende Informationen zu Frage 2: Interaktionsmechanismen sind die grundsätzlichen Verhaltensweisen, die die Interaktion innerhalb eines Systemverbunds steuern. Bspw. kann innerhalb einer Lagerstätte der Grundsatz bestehen, dass Objekte, die als erstes eingelagert wurden, auch als erstes wieder entnommen werden sollen (First-In-First-Out-Prinzip). Diese Grundsätze steuern die grundsätzliche Interaktion und sind ein Hinweis für ein bereits bestehendes SoS.

- **Frage 3:** Gibt es Standards oder Protokolle für Schnittstellen, die eine Kommunikation zwischen den beteiligten Systemen ermöglichen können? (Ja | Nein)

Ergänzende Informationen zu Frage 3: Standardisierte Schnittstellen sind die Grundlage für die Interoperabilität zwischen unabhängigen Systemen. Kann ein Großteil der notwendigen Systeme bereits miteinander interagieren da ein Schnittstellenstandard etabliert ist, so kann auf diesem aufgebaut werden. Ein Beispiel ist OPC UA im Kontext der vernetzten Produktion (vgl. Kapitel 2.2.3).

- **Frage 4:** Wird das Projekt aus Sicht eines einzelnen Unternehmens durchgeführt oder hat ein einzelnes Unternehmen die Kontrolle über das SoS? (Ja | Nein)

Ergänzende Informationen zu Frage 4: Ist der Anwender des Fragebogens ein einzelnes Unternehmen, welches ein SoS-Konzept entwickelt, ist die Frage mit „Ja“ zu beantworten. Auch wenn ein einzelnes Unternehmen die wesentlichen Systeme innerhalb des SoS kontrolliert ist die Frage mit „Ja“ zu beantworten. Dies ist vor allem bei Unternehmen mit einer breiten Produktpalette und einer starken Marktposition möglich.

- **Frage 5:** Sind die Unternehmen, die zur Umsetzung des SoS-Konzepts notwendig sind, bekannt und/oder festgelegt? (Ja | Nein)

Ergänzende Informationen zu Frage 5: Die notwendigen Systeme für die Umsetzung des SoS werden von unterschiedlichen Unternehmen gemanagt. Sind diese bereits durch vorgelagerte Entscheidungen bspw. durch das Management festgelegt, ist die Antwort „Ja“ zutreffend. Wurde die Frage mit „Nein“ beantwortet, wird im Rahmen der anschließenden Gestaltung eine Analyse für ein passendes Zusammenarbeitsmodell zwischen Unternehmen durchgeführt.

- **Frage 6:** Sind die einzelnen Systeme bekannt und können diese beeinflusst bzw. angepasst werden? (White-, Grey-Box | Black-Box | Unbekannt)

Ergänzende Informationen zu Frage 6: Der Einflussgrad wird anhand der Zugriffstiefe der beteiligten Systeme beschrieben. Sind das interne Verhalten und Funktionen der beteiligten Systeme bekannt und können durch beteiligte Unternehmen oder ein SoS-Team angepasst werden, so liegt die Stufe White- bis Grey-Box vor. Dies bedeutet, dass in das Innere eines Systems geschaut werden kann. Sind lediglich das äußere Verhalten und die externen Schnittstellen bekannt, so wird von einer Black-Box gesprochen. Bei einer Black-Box Betrachtung wird die konkrete Lösung im Inneren des Systems nicht betrachtet und kann auch nicht angepasst werden. Sind das Verhalten und die Funktionen der beteiligten Systeme gänzlich unbekannt, ist die Antwort „Unbekannt“ zutreffend.

- **Frage 7:** Welches Ziel wird verfolgt? (1 | 2 | 3 | 4 | 5 | Unbekannt)
 1. Zusammenarbeit unter Know-How-Schutz
 2. Optimierung einzelner Leistungsparameter
 3. Etablierung von Standards innerhalb einer Branche oder eines Anwendungsbereichs
 4. Kompatibilität zwischen möglichst vielen Systemen (offene Standards)
 5. Maximaler Kundennutzen durch breite Anwendbarkeit
 6. Unbekannt

Ergänzende Informationen zu Frage 7: Die hier aufgeführten Ziele entsprechen den Zielen, die aus den analysierten SoS-Beispielen abgeleitet wurden. Die Ziele erheben keinen Anspruch auf Vollständigkeit, sondern dienen der ersten Einschätzung eines passenden Gestaltungskonzepts. Alternativ werden dem Anwender die Gestaltungskonzepte 2, 3 und 4 vorgestellt und auf Basis der Beschreibung ein passendes Gestaltungskonzept ausgewählt.

Die Fragen bauen aufeinander auf und bedingen sich gegenseitig. Die Logik der Fragenreihenfolge in Abhängigkeit der Antworten wird anhand eines Entscheidungsbaums visualisiert (Bild 4-21). Durch die Beantwortung der Fragen wird der Anwender zu einem passenden Gestaltungskonzept geführt.

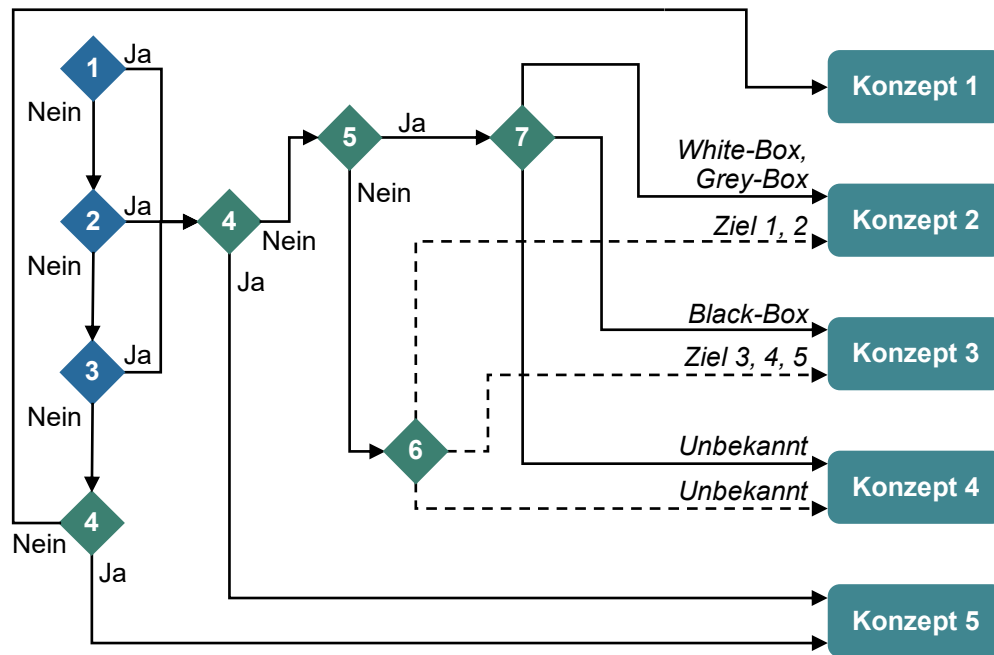


Bild 4-21: Fragebaum zur Ableitung eines SoS-Gestaltungskonzepts

Nach Beantwortung der Fragen wird entsprechend des Auswahlmechanismus ein Gestaltungskonzept vorgeschlagen. Die Gestaltungskonzepte 1 bis 5 entsprechen den in Kapitel 4.3.1 beschriebenen SoS-Gestaltungskonzepten.

Das SoS-Gestaltungskonzept und die entsprechenden Methoden sind die Ausgangsbasis für **Phase 3** des Vorgehensmodells. Im Rahmen der Ausarbeitung werden die entsprechenden Leitlinien und Hinweise des Gestaltungskonzepts berücksichtigt und die verwiesenen Methoden angewandt. Die Durchführung der jeweiligen Tätigkeiten kann zu Erkenntnissen führen, die eine Anpassung der zuvor definierten Inhalte zur Folge haben. Sowohl dem Vorgehen als auch den jeweiligen Methoden liegt ein iterativer Charakter zugrunde, der eine erneute Durchführung der Tätigkeiten ermöglicht.

Um darüber hinaus die Tätigkeiten bei der Gestaltung von SoS in Phase 3 zu unterstützen, wird dem Anwender zusätzlich Gestaltungswissen bereitgestellt. Dieses unterstützt bzgl. der Herausforderungen von SoS, die sich in den charakteristischen Merkmalen zeigen (vgl. Kapitel 4.2.1). Das Gestaltungswissen sowie die Möglichkeiten zur Nutzung des Gestaltungswissens werden in Kapitel 4.4 vorgestellt und ausführlich erläutert.

4.3.4 Erweiterte Nutzung des Gestaltungsrahmens

Der Gestaltungsrahmen gibt Organisationen Hinweise zur Bewältigung von Herausforderungen innerhalb des jeweiligen Gestaltungskonzepts. Dabei ergeben sich verschiedene Entscheidungsmöglichkeiten, die einen Wechsel des Gestaltungskonzepts zur Folge haben können. Bild 4-22 gibt einen Überblick von typischen Übergängen zwischen den Gestaltungskonzepten.

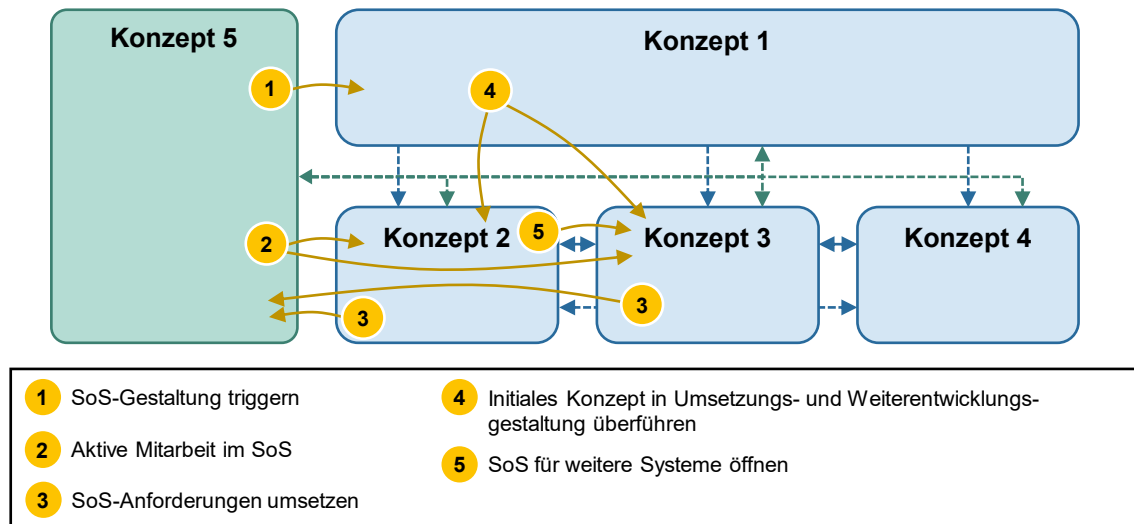


Bild 4-22: Typische Übergänge zwischen SoS-Gestaltungskonzepten

Ein wichtiger Zusammenhang besteht zwischen der Gestaltung aus Unternehmenssicht (Konzept 5) und der Gestaltung auf SoS-Ebene (Konzept 1 bis 4). Diese finden häufig parallel statt und bedingen sich gegenseitig. Dennoch werden durch bestimmte Analysen und Entscheidungen Wechsel zwischen den Konzepten notwendig. Durch Voranalysen eines Unternehmens kann bspw. der **Übergang 1** getriggert werden. Ein einzelnes Unternehmen entscheidet sich in diesem Fall, eine neue SoS-Lösung zu entwickeln und diese zusammen mit weiteren Unternehmen im Rahmen von Konzept 1 zu initiieren. Wie in Kapitel 4.3.1.5 beschrieben, muss ein Unternehmen hierfür zunächst die Rahmenbedingungen und eigenen Ziele abwägen, um anschließend mit geeigneten Unternehmenspartnern in den Austausch zu gehen. Zusammen interagieren die Unternehmen dann innerhalb von Gestaltungskonzept 1 und entwickeln eine initiale SoS-Lösung.

Übergang 2 beschreibt die aktive Mitarbeit eines Unternehmens innerhalb eines bestehenden SoS. Hat ein Unternehmen ein Interesse an einer Anpassung innerhalb eines SoS, so ist eine aktive Mitarbeit in Kooperationsbeziehungen oder Gremien notwendig. Auch wenn ein SoS bereits in Betrieb ist, ist die kontinuierliche Mitarbeit essenziell, um die Auswirkungen von Weiterentwicklungen auf Systeme des eigenen Unternehmens nachzuziehen und ggf. im eigenen Interesse beeinflussen zu können.

Hieraus können sich wiederum auch neue Anforderungen an die eigenen Systemen ergeben, sodass ein Übergang zwischen der SoS-Sicht und der Unternehmenssicht notwendig wird. Wie in Bild 4-22 visualisiert, kann der **Übergang 3** sowohl aus Gestaltungskonzept 2 als auch 3 relevant werden. Innerhalb von Gestaltungskonzept 5 müssen dabei die Anforderungen aus der SoS-Betrachtung in die Systemebene überführt und umgesetzt werden.

Übergang 4 ist grundsätzlich notwendig für eine erfolgreiche SoS-Umsetzung und folgt dem Wellenmodell nach DAHMANN et al. (vgl. Kapitel 3.2.1.1) [DRL+11, S. 214f.]. Im

Anschluss an eine SoS-Initialisierung erfolgt demnach in kontinuierlichen Entwicklungszyklen die Weiterentwicklung des SoS. Dies kann sowohl durch eine direkte Unternehmenskooperation (Konzept 2) als auch im Rahmen von Gremien (Konzept 3) erfolgen.

Eine typische Gestaltungsstrategie stellt der **Übergang 5** dar. Dabei wird zunächst anhand eines eingegrenzten Umfangs die Interaktion zwischen eigenständigen Systemen innerhalb einer direkten Unternehmenskooperation definiert und in Betrieb genommen (Konzept 2). Sobald das SoS etabliert ist und ein entsprechender Bedarf besteht, kann das SoS anschließend für weitere Systeme geöffnet werden und der Übergang in ein gremiengesteuertes SoS erfolgen. Dies hat den Vorteil, dass die Unternehmen, die Teil der ersten Entwicklungsstufe waren, einen gewissen Entwicklungsvorsprung haben und ihre Marktposition gestärkt ist.

Die Übergänge geben einen Einblick in die Zusammenhänge zwischen den Gestaltungskonzepten. Insbesondere der Zusammenhang zwischen der SoS-Sicht und der System-sicht einzelner Unternehmen ist für eine erfolgreiche Etablierung und Evolution eines SoS relevant. Je formaler Modelle auf beiden Ebenen aufgebaut werden, desto besser lassen sich Zusammenhänge zwischen Systemen und SoS nachvollziehen. So können Unternehmen die Informationen aus der SoS-Gestaltung durchgängig mit der Systementwicklung verknüpfen und bspw. die Auswirkungen von neuen Anforderungen der SoS-Ebene auf die Systementwicklung analysieren.

4.4 Hilfsmittel für die wissensbasierte SoS-Gestaltung

Wissen ist notwendig, um Aufgaben zu lösen (vgl. Kapitel 2.4). Zahlreiche Herausforderungen bestehen bei der Gestaltung von SoS, die ein entsprechendes Wissen voraussetzen. Da im Gegensatz zu Systementwicklungsprozessen das Wissen nicht vollständig innerhalb einer Organisation liegt und häufig Wissenslücken aufgrund der vielfältigen Herausforderungen in Bezug auf SoS bestehen, muss ein unternehmensübergreifendes Wissensmanagement etabliert werden.

Eine geeignete und im Kontext von SoS bereits erprobte Form stellen sowohl Lösungsmuster als auch Designprinzipien dar (vgl. Kapitel 2.4 und 3.4). Diese sind in der Lage Wissen für die Gestaltung von SoS zu dokumentieren und bereitzustellen. Da sich das Wissen im Gegensatz zu klassischen Wissensmanagementansätzen nicht auf ein einzelnes Unternehmen bezieht, muss für die Identifikation dieser Wissens Elemente ein geeignetes Verfahren gefunden werden. Dieses muss über die Befragung und Externalisierung von Erfahrungswissen einzelner Mitarbeiter hinausgehen und auf das kollektive unternehmensübergreifende Wissen zurückgreifen.

Im Rahmen der Arbeit wurden zwei Verfahren genutzt, die sich jeweils auf publizierte Wissens Elemente (Designprinzipien und Lösungsmuster) stützen. Die Identifikation von etablierten Designprinzipien erfolgte anhand einer systematischen Literaturanalyse, die

in einer Vorveröffentlichung publiziert wurde [GSA+24]. Diese hat zum Ziel, Designprinzipien zu identifizieren, die die Herausforderungen von SoS adressieren. Die Literaturanalyse wurde entsprechend dem Vorgehen nach XIAO und WATSON durchgeführt (vgl. Kapitel 4.2) [XW19]. Als Datenbank wurde SCOPUS ausgewählt. In dieser sind die relevanten Konferenzen und Journals aus dem Fachbereich SoS und Systems Engineering gelistet. Die Suchanfrage setzt sich aus den relevanten Begriffen mit entsprechenden Wortvariationen zusammen.

Suchanfrage (SCOPUS): TITLE-ABS-KEY (("system of systems" OR "systems of systems" OR "SoS") AND ("principle")) SRCTITLE ("systems engineering")

Insgesamt wurden 133 Literaturquellen ausgegeben, die anhand des in Bild 4-23 visualisierten Vorgehens auf zehn relevante Beiträge reduziert wurden. Diese sind die Ausgangsbasis für die in Kapitel 4.4.1 beschriebenen Designprinzipien für SoS.

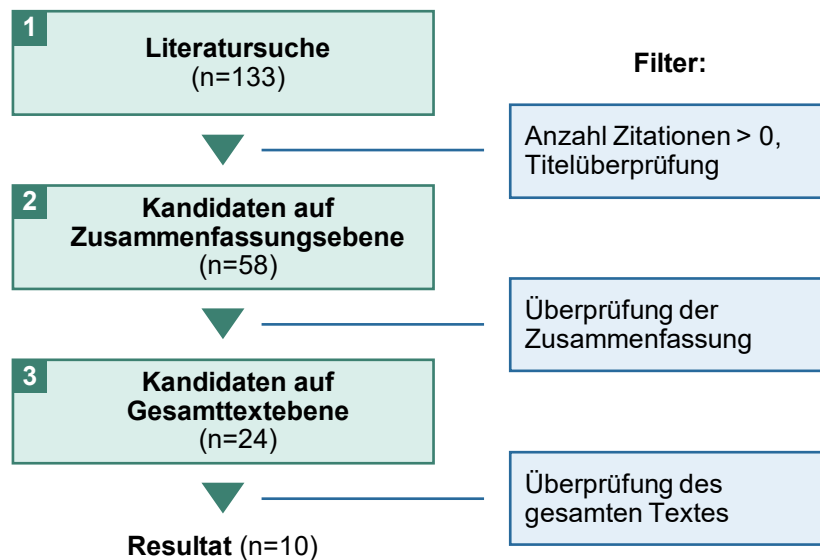


Bild 4-23: SLR zur Identifikation von Designprinzipien

Die Erarbeitung von Lösungsmustern erfolgte anhand der Action Design Research Methode nach SEIN et al. im Rahmen des Forschungsprojektes MoSyS (vgl. Kapitel 4.3.1) [SHP+11]. Hierzu wurde die Wissensbasis aus der Literatur genutzt und durch die Analyse, Anwendung und Reflexion zusammen mit den Anwendern weiterentwickelt. Als Literaturbasis wurden zum einen die im Stand der Forschung beschriebenen Lösungsmuster für SoS herangezogen, wie u. a. die Modellierungsmuster für SoS-Architekturen nach INGRAM et al. (vgl. Kapitel 3.3.1.1). Zum anderen wurde eine systematische Literaturanalyse zum Thema Lösungsmuster durchgeführt, die den grundsätzlichen Zusammenhang zwischen Lösungsmustern und SoS darlegt und entsprechende Lösungsmuster identifiziert. Das Vorgehen wird in Bild 4-24 zusammengefasst.

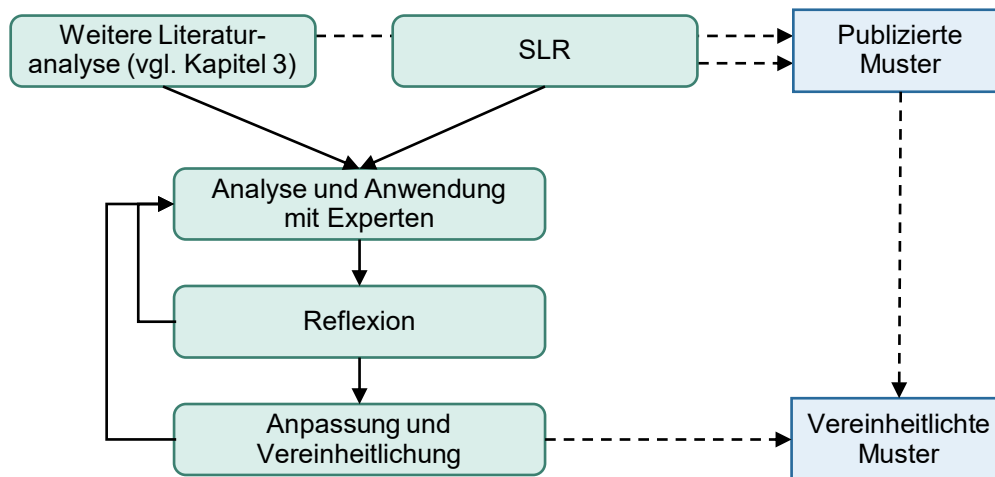


Bild 4-24: Vorgehen der Lösungsmusteridentifikation und -dokumentation

Die Literaturlbasis sowie die iterative Analyse, Anwendung und Reflexion der Inhalte führte zu zwei Erkenntnissen. Zum einen sind die in der Literatur beschriebenen Lösungsmuster teilweise eher generellen Handlungsanweisungen zuzuordnen. Daher wurde die Aufteilung der Wissensselemente in Designprinzipien und Lösungsmuster durchgeführt. Zum anderen sind die identifizierten Lösungsmuster auf unterschiedlichen Abstraktionsniveaus beschrieben, sodass eine Angleichung der Lösungsmuster notwendig wurde. Die Ergebnisse des Vorgehens werden in Kapitel 4.4.2 vorgestellt.

4.4.1 Designprinzipien

Designprinzipien legen bestimmte Handlungsweisen und Richtlinien dem Handeln zugrunde und sollen Hilfestellung bei der Gestaltung liefern. Aufgrund der zahlreichen Herausforderungen sind Designprinzipien auch im Kontext von SoS ein etabliertes Konzept, das den jeweiligen Anwender bei der Gestaltung unterstützen soll. Insgesamt konnten durch die SLR 65 Designprinzipien aus zehn Literaturquellen identifiziert werden. Diese adressieren unterschiedliche Herausforderungen der SoS-Gestaltung und sind teils sehr generisch. Um die Vielzahl an Designprinzipien zu strukturieren, wurde ein Mapping auf die acht charakteristischen Merkmale von SoS durchgeführt. Dies ermöglicht gleichzeitig Anwendern, Designprinzipien für eine spezifische Herausforderung zu identifizieren. Zusätzlich konnten durch das Mapping generische Designprinzipien identifiziert werden, sofern diese auf alle charakteristischen Merkmale zutreffen. Als Ergebnis wurden 26 generische und 39 spezifische Designprinzipien identifiziert (Bild 4-25).

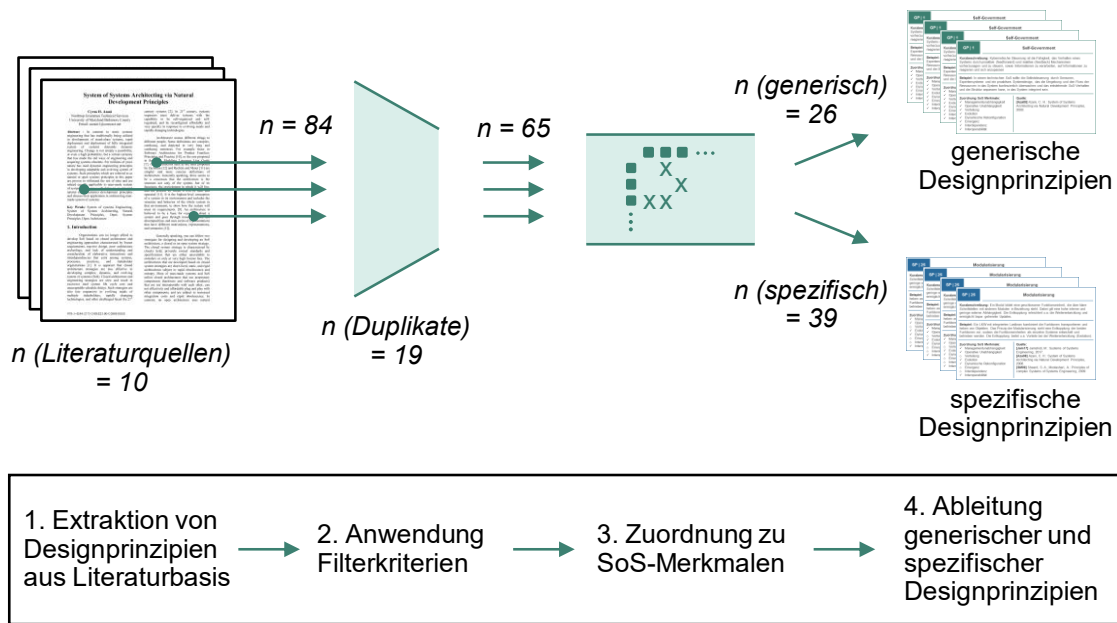


Bild 4-25: Vorgehen zur Identifikation von generischen und spezifischen Designprinzipien

Die generischen Designprinzipien gehen sehr allgemein auf die Herausforderungen von SoS ein, ohne Entwicklern konkrete Handlungsanweisungen zu geben. Ein Beispiel ist das von SHEARD und MOSTASHARI beschriebene Prinzip „Verzicht auf das letzte bisschen Optimierung“ [SM09, S. 304f.]. Während in klassischen Systementwicklungsprojekten die Zusammenhänge und Rahmenbedingungen weitestgehend bekannt sind und auf Einzelheiten eingegangen werden kann, ist dies im SoSE nicht der Fall. Zahlreiche unbekannte Faktoren ergeben sich bspw. durch Emergenzen und dynamische Systeminteraktionen. Daher wird den Beteiligten geraten, die SoS-Gestaltung nicht bis ins letzte Detail zu optimieren und sich zunächst auf die wesentlichen Elemente zu konzentrieren. Die generischen Designprinzipien helfen Entwicklern somit die richtige Einstellung zu finden und gewisse Rahmenbedingungen zu schaffen, die eine möglichst erfolgreiche SoS-Gestaltung fördern. Tabelle 4-4 zeigt einen Auszug aus den identifizierten generischen Designprinzipien, die in den jeweiligen Quellen zum Teil ausführlich beschrieben sind. Die vollständige Liste ist dem Anhang A2.2 zu entnehmen.

Tabelle 4-4: Übersicht (Auszug) von generischen Designprinzipien

Nr.	Quelle	Prinzip
1	[SM09]	Verwenden Sie einen Entscheidungsbaum, um festzustellen, welche Designansätze (Bottom-up, Top-Down, Analogie oder interaktive Evolution) am besten funktionieren
2		"Verzicht auf das letzte bisschen Optimierung"
3		Erstellung umfassender mentaler Modelle zum Verständnis des Problembereichs
11	[Ada11]	Pareto
12		Erforderliche Sparsamkeit
13		Erforderliche Prägnanz
14		Minimale kritische Spezifikation
15	[Hes12]	Prinzip der Multifinalität
16		Zirkuläre Kausalität
17	[WJ11]	1 (Bescheidenheit)
18		2 (Umfassende Systemperspektive)
19		4 (Transdisziplinäre Integration)
24	[MSB09]	Nutzung von Systemigrammen
25		Paradoxes Denken
26	[Cho16]	Einsatz der richtigen Informationstechnologie
27	[UM15]	Eigenschaften auf Systemebene verbessern
28		Einsatz von Menschen
29		Präventive Korrektur
30		Kombination von Resilienzprinzipien

Die spezifischen Designprinzipien wurden im Rahmen des Mappings den acht SoS-Merkmalen zugeordnet Tabelle 4-5. Das Verständnis der Merkmale sowie die Nummerierung folgt der Auflistung in Kapitel 4.2.1. Sämtliche SoS-Merkmale werden demnach durch Designprinzipien unterstützt. Dabei werden einige Prinzipien durch mehrere Autoren aufgegriffen, wodurch ihre Relevanz hervorgehoben wird. Diese wurden entsprechend mit einem (*) gekennzeichnet. Ein Beispiel ist Modularisierung, welches von drei Autoren als geeignetes Designprinzip vorgestellt wird. Modularisierung gilt als vielversprechendes Prinzip im Rahmen der SoS-Gestaltung, wie u. a. AZANI hervorhebt [Aza08, S. 5]. Durch die Aufteilung in unabhängige und austauschbare Module, die jeweils Funktionen bündeln, werden Herausforderungen wie die Managementunabhängigkeit, Evolution oder auch die Interoperabilität durch standardisierte Schnittstellen zwischen den Modulen adressiert.

Tabelle 4-5: Übersicht (Auszug) von spezifischen Designprinzipien und zugeordnete SoS-Merkmale

Nr.	Quelle	Prinzip	SoS-Merkmal							
			1	2	3	4	5	6	7	8
1	[Mai98]	Stabile Zwischenformen				x				
2		Policy Triage				x				
3		Hebelwirkung an den Schnittstellen								x
4		Sicherstellung der Zusammenarbeit	x							
5	[SM09]	Denken Sie eher an die "Entwicklung" des Systems als an die "Gestaltung" des Systems				x				
6		Suchen Sie nach lokalen Aktionen, die globale Auswirkungen haben können.						x		
7		Mehrere Möglichkeiten lebensfähig halten	x	x			x	x	x	x
8		Dem System explizit Vielfalt auferlegen		x				x		x
21	[Jam17], [Aza08]	Standardisierte Schnittstellen								x
22		Selbstverwaltung	x			x	x			
23		Konservierung				x			x	
24		Neugestaltung					x			x
25		Symbiose	x							
26		Modularität	x	x		x	x			x
27	[HPC+09]	Bedingung der Emergenz						x		
28		Emergentes Verhalten ist umgekehrt proportional zum Grad der Bindung zwischen Systemen						x		
29		Emergentes Verhalten ist nichtlinear						x		
30		Emergentes Verhalten ist selbstorganisiert						x		
31	[WJ11]	3 - Systemisches Gleichgewichtsprinzip	x	x						
32		9 - Vertrauenszentrierte Zusammenarbeit	x							
33		12 - Die Philosophie des einfachen Elementdesigns		x		x	x		x	x
34		13 - Geschichtete, lose gekoppelte Architektur		x		x	x		x	x
35	[MSB09]	Integrierte Produktteams	x							
36		Globales Risikomanagement und Entscheidungsanalyse	x		x	x				
37		Schnittstellenmanagement								x
38		Agile Entwicklungskonzepte: Plug-n-Play-Architekturskalierbarkeit					x	x		
39	[Cho16]	Unsicherheiten einbeziehen					x	x		
40	[UM15]	Physikalische Redundanz				x	x			
41		Funktionale Redundanz				x	x			
42		Reparierbarkeit								
46	[RRR41]	Nutzung von Vorfahren				x				
47		Disruptive architektonische Neugestaltung				x				
48		Nachahmung				x				
49		Sinnvolle Adaptierung				x				

Neben den Ergebnissen der SLR werden zusätzlich die im Stand der Forschung beschriebenen Ansätze nach UDAY und MARAIS sowie RICCI et al. in den Listen aufgeführt (vgl.

Kapitel 3.3.1.5 und 3.3.1.7) [UM15], [RRR14]. Insgesamt ergeben sich somit 30 generische und 57 spezifische Designprinzipien, die sich entsprechend dem Ansatz nach GAMMA et al. in Form eines Katalogs strukturieren lassen (vgl. Kapitel 3.3.2.1). Neben der Einordnung in generische und spezifische Lösungsmuster ermöglicht auch die Zuordnung zu den charakteristischen Merkmalen die einfache Identifikation von Lösungswissen. Auch alternative bzw. sich ergänzende Prinzipien können durch die erfolgte Strukturierung erkannt werden.

Die wichtigsten Informationen der Prinzipien wurden in Form von Kurzsteckbriefen festgehalten. Diese umfassen eine kurze Beschreibung des Prinzips, wenn möglich ein Beispiel sowie die Zuordnung zu den charakteristischen Merkmalen. Für weitere Informationen wird auf die jeweilige Literatur verwiesen. Bild 4-26 zeigt einen Kurzsteckbrief anhand des spezifischen Designprinzips „Modularisierung“.

SP 26	Modularisierung
Kurzbeschreibung: Ein Modul bildet eine geschlossene Funktionseinheit, die über klare Schnittstellen mit anderen Modulen in Beziehung steht. Dabei gilt eine hohe interne und geringe externe Abhängigkeit. Die Entkopplung erleichtert u.a. die Weiterentwicklung und ermöglicht bspw. getrennte Updates.	
Beispiel: Ein LKW mit integriertem Lastkran kombiniert die Funktionen „transportieren“ und „heben von Objekten“. Das Prinzip der Modularisierung sieht eine Entkopplung der beiden Funktionen vor, sodass die Funktionseinheiten als einzelne Systeme entwickelt und betrieben werden. Die Entkopplung bietet u.a. Vorteile bei der Weiterentwicklung (Evolution).	
Zuordnung SoS Merkmale: ✓ Managementunabhängigkeit ✓ Operative Unabhängigkeit ○ Verteilung ✓ Evolution ✓ Dynamische Rekonfiguration ○ Emergenz ○ Interdependenz ✓ Interoperabilität	Quelle: [Jam17] Jamshidi, M.: Systems of Systems Engineering, 2017. [Aza08] Azani, C. H.: System of Systems Architecting via Natural Development Principles, 2008. [SM09] Sheard, S. A.; Mostashari, A.: Principles of complex Systems of Systems Engineering, 2009

Bild 4-26: Kurzsteckbrief des spezifischen Designprinzips „Modularisierung“

4.4.2 Lösungsmuster für die SoS-Gestaltung

Da nicht immer alle Informationen vollständig vorhanden sind, werden weitere Hilfsmittel im Rahmen der Gestaltung benötigt. In diesem Kontext sind Lösungsmuster wiederkehrende Vorlagen zur Lösung von Problemen, die sich bereits in existierenden Umsetzungen bewährt haben. Diese können von Entwicklerinnen und Entwicklern angewendet und für konkrete Problemstellungen adaptiert werden. Es existieren bereits zahlreiche SoS, in denen Lösungen für unterschiedliche Problemstellungen gefunden wurden. Dabei zeigen sich Ähnlichkeiten zwischen den Lösungen. Dieses Wissen über wiederkehrende Lösungen in Form von Lösungsmustern hilft Unternehmen bei der Gestaltung von SoS. Zusätzlich können Muster die Etablierung von Standards fördern und dadurch die Interoperabilität zwischen interagierenden Systemen ermöglichen (vgl. Kapitel 2.4.2).

Wie in Kapitel 3.3.1 aufgezeigt, bestehen bereits Lösungsmusteransätze, um einzelne Herausforderungen von SoS zu adressieren. Diese sind jedoch unterschiedlich strukturiert und nicht einheitlich beschrieben. Um die im Rahmen dieser Arbeit diskutierten Muster einheitlich zu dokumentieren, erfolgt in Anlehnung an INGRAM et al. und auf Grundlage von ALEXANDER et al. die Strukturierung anhand von vier wesentlichen Kategorien: *Name*, *Kontext*, *Problem* und *Lösung* (vgl. Kapitel 3.3.1.1) [IPP+14], [AIS+77]. Diese werden im Rahmen dieser Arbeit für die unterschiedlichen Problem-Lösungspaare der SoS-Gestaltung ausgeprägt. Durch die vielfältigen Gestaltungstätigkeiten und SoS-Dimensionen wird daher im Folgenden eine Orientierungshilfe für die einzelnen Kategorien gegeben. Das als Grundlage dienende Lösungsmustertemplate wird in Bild 4-27 visualisiert und im Folgenden detailliert erläutert.

Muster Name		Kontext	
Problembeschreibung Welches Ziel, welche Aufgabe oder welche Funktionalität soll gelöst werden? (Bsp. Ziel, Anwendungsfall, Funktion, ...)		Gestaltungskonzeptmerkmale	
Lösungskonzept - Welches wiederkehrende Muster kann zur Lösung des Problems genutzt werden? (Bsp. funktionales oder logisches Konzept) (ggf. inkl. Grafik)		Managementunabh. abhängig unabhängig <input type="range"/>	Operative Unabhängigkeit abhängig unabhängig <input type="range"/>
		Emergentes Verhalten gering stark <input type="range"/>	Räumliche Verteilung lokal global <input type="range"/>
		Interoperabilität hoch gering <input type="range"/>	Evolution gering stark <input type="range"/>
Referenzen Literaturangabe		Beziehung zu weiteren Mustern Zusammenhängende Muster: Gibt es Muster, die im Zusammenhang zu diesem Muster stehen? Alternative Muster: Gibt es alternative Muster, die nicht gleichzeitig oder in Kombination angewendet werden können?	

Bild 4-27: SoS-Lösungsmustertemplate

Neben dem **Namen** ist insbesondere der **Kontext** wichtig, um ein Wissenselement zu identifizieren und den Einsatzzweck zu verstehen. Daher wurden vor dem Hintergrund einer geeigneten Einordnung der Wissens Elemente die charakteristischen Merkmale von SoS mit in den Steckbrief aufgenommen. Die Einordnung der Lösungsmuster anhand der Merkmale hat das Ziel, die Strukturierung von Lösungswissen in Form eines Katalogs entsprechend dem Ansatz nach GAMMA et al. zu unterstützen (vgl. Kapitel 3.3.2.1) [GHJ+93]. Ein Lösungsmuster soll demnach anhand der charakteristischen Merkmale bewertet werden und somit einen Bezug zu der adressierten Herausforderung herstellen. Dabei muss nicht jedes Lösungsmuster auf alle Herausforderungen eingehen, bzw. diese

im Rahmen des Steckbriefs bewerten. Einzelne Lösungsmuster können spezielle Lösungen für ein einzelnes Merkmal darstellen und dabei die anderen Kategorien unbewertet lassen. Dies gibt einen Hinweis auf die Anzahl an Herausforderungen, die unterstützt werden und somit gleichzeitig auf den Spezialisierungs- bzw. Generalisierungsgrad der wiederkehrenden Lösung. Die gestaltungskonzeptrelevanten Merkmale ermöglichen zusätzlich eine Einordnung in die fünf Konzepte der SoS-Gestaltung. Sind diese entsprechend dem Merkmalsprofil der Gestaltungskonzepte ausgeprägt, ist eine Verwendung in diesem Gestaltungskonzept möglich (vgl. Kapitel 4.3.1.1 bis 4.3.1.4).

Die Kategorien **Problembeschreibung** und **Lösungskonzept** sind bewusst einfach gehalten. Dies hängt mit den unterschiedlichen Problemstellungen und den damit einhergehenden vielfältigen Problem-Lösungspaaren im Rahmen der SoS-Gestaltung zusammen. Einen Ansatzpunkt bieten die in Kapitel 4.3.2 beschriebenen Methodensteckbriefe. Diese beschreiben jeweils Verfahren, wie eine entsprechende Fragestellung bearbeitet werden kann. Dabei werden die Tätigkeiten entsprechend dem Problemlösungszyklus nach HABERFELLNER et al. angeordnet und somit inhärent ein Zusammenhang zwischen problemorientierten und lösungsorientierten Artefakten gegeben (vgl. Kapitel 2.3.1) [HWF+19, S. x]. Diese können die Grundlage zur Identifikation und Beschreibung der Lösungsmuster sein (Bild 4-28). Wird die Methode durch eine spezifische Sprache unterstützt, sollte diese auch zur Beschreibung der Lösungsmuster verwendet werden. Zusätzlich können durch den Vergleich von problemorientierten Artefakten innerhalb von Gestaltungsvorgängen bereits beschriebene Lösungsmuster anhand der Problembeschreibung identifiziert werden.

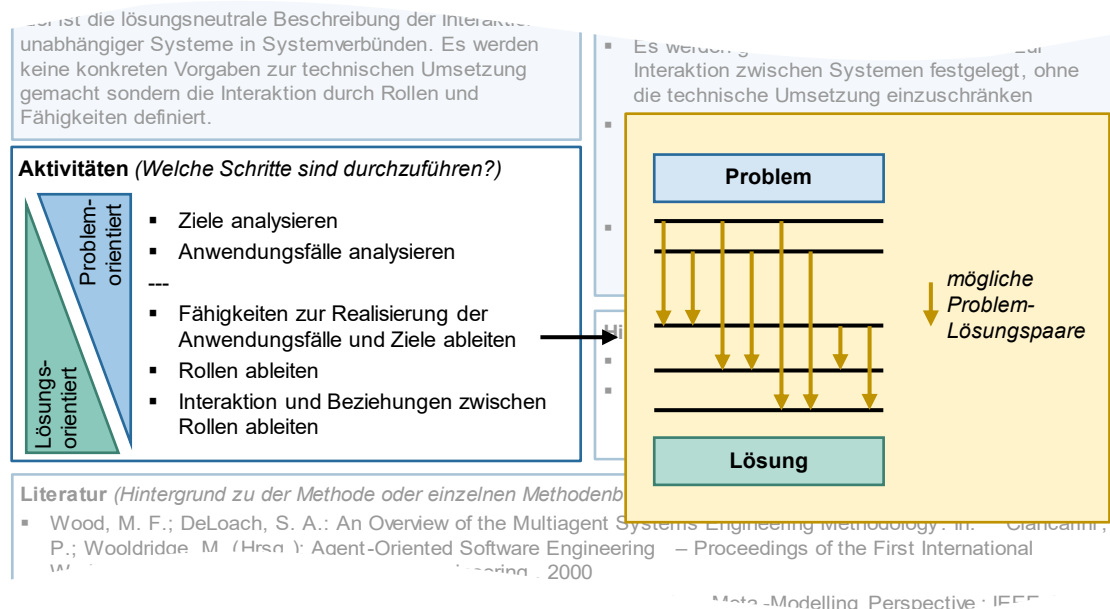


Bild 4-28: Identifikation von Problem-Lösungspaaren anhand der Methodensteckbriefe

Entsprechend dem zuvor in Bild 4-19 beschriebenen Vorgehen wurden insgesamt sieben Lösungsmuster auf Basis des Lösungsmustertemplates ausgearbeitet. Diese stellen Lösungsmuster für unterschiedliche Herausforderungen dar und haben keinen Anspruch auf

Vollständigkeit. Vielmehr soll Anwendern eine erste Basis an Wissen bereitgestellt werden, welches sich in Anwendungsbeispielen als sinnvoll gezeigt hat. Ein Beispiel ist das **Zentrale Architekturmuster**, welches u. a. durch INGRAM et al. beschrieben wurde (Bild 4-29). Dieses kann auf unterschiedliche Art und Weise für konkrete Anwendungen adaptiert werden, wie z. B. für die zentrale Energieversorgung. Dabei wird das Energielevel eines Netzwerks über einen zentralen Knoten verwaltet, in dem sämtliche Verbraucher und Erzeuger von einer zentralen Einheit gesteuert werden [YNF19, S. 3].

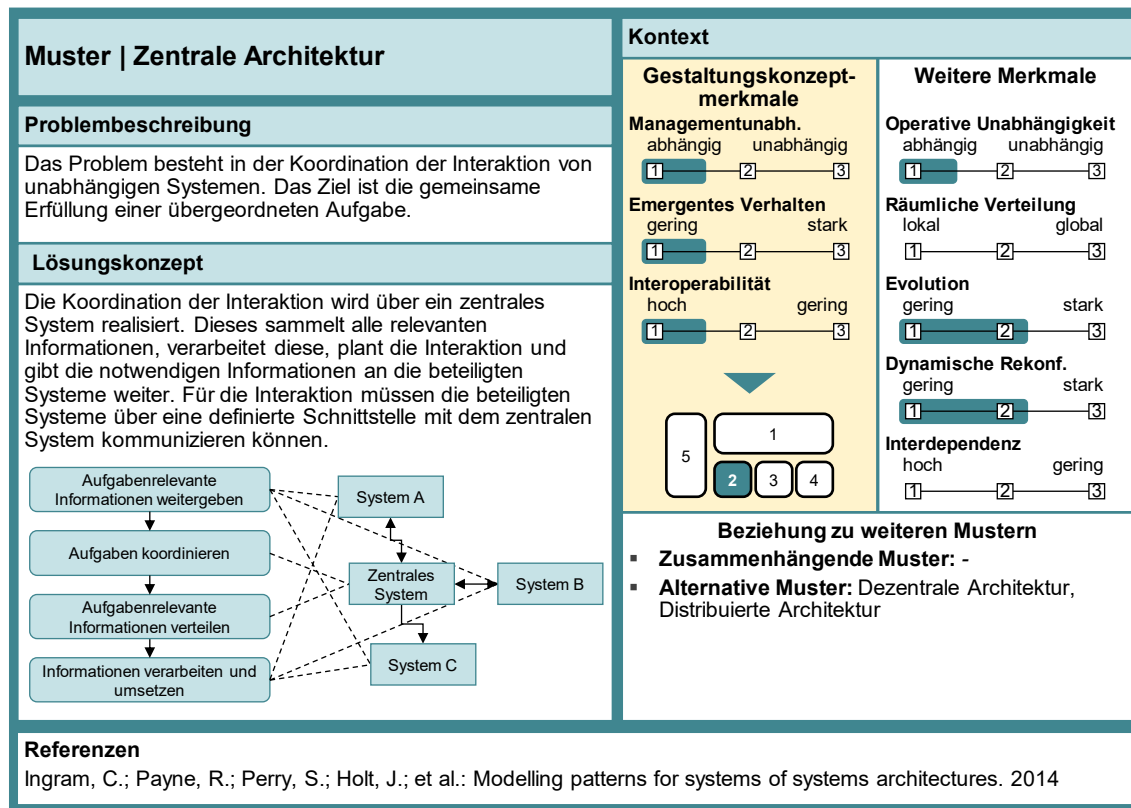


Bild 4-29: SoS-Lösungsmuster am Beispiel „Zentrale Architektur“

Weitere Lösungsmuster, die im Rahmen der Arbeit ausgearbeitet wurden, werden im Folgenden in Kurzform beschrieben. Die vollständigen Lösungsmuster sind im Anhang A2.3 aufgeführt.

Dezentrale Architektur: Die dezentrale Architektur gilt als Alternative zur zentralen Architektur. Das Lösungsmuster charakterisiert eine operative Unabhängigkeit und Managementunabhängigkeit durch dezentral organisierte Cluster von eigenständigen Systemen. Das Muster eignet sich daher für Problemstellungen aus dem Gestaltungskonzept 3. Auch dieses Muster kann als Beispiel auf Problemstellungen der Energieversorgung angewendet werden und zeigt sich in einem dezentral gemanagten Energienetz.

Verteilte Architektur: Das Muster der verteilten Architektur beschreibt eine Struktur von verteilten lokalen Steuerungen, die sich dynamisch untereinander vernetzen, um ihre Interaktion zu regeln. Sie stellt eine weitere Alternative zur zentralen und dezentralen Architektur dar und ist insbesondere für SoS mit unabhängigen Strukturen relevant.

Leader-Follower (Master-Client) Interaktion: Die Leader-Follower Interaktion beschreibt eine Rollenverteilung innerhalb eines SoS, in dem ein System die Rolle des Leaders einnimmt und Vorgaben macht, die von weiteren Systemen befolgt werden. Sie geht häufig einher mit einer zentralen Architektur in einem Kommunikationsnetzwerk, zeigt sich aber auch in dem Beispiel der Flottenfahrt.

Publish-subscribe Interaktion: Dieses Muster beschreibt eine kommunikationsbasierte Interaktionen im SoS, in der ein System Informationen bereitstellt oder Fähigkeiten anbietet, die von weiteren Systemen nachgefragt und genutzt werden. Diese Rollenverteilung bietet sich insbesondere in Gestaltungskonzept 3 an und wird durch das Muster der Vertragsdefinition nach FALDIK et al. gestützt (vgl. Kapitel 3.3.1.4).

Autorisierung: Das Autorisierungsmuster beschreibt einen Mechanismus, um eine sichere Interaktion innerhalb eines Systemverbunds zu realisieren. Durch einen Vergleich mit einer Liste an autorisierten Systemen kann eine Interaktion ermöglicht oder im Fall eines unautorisierten Zugriffs verhindert werden.

Over-the-Air-Update: Das Muster beschreibt das grundsätzliche Konzept, ein System im Betrieb über ein Softwareupdate an veränderte Rahmenbedingungen anzupassen oder neue Fähigkeiten zu ermöglichen. Dies funktioniert im Zusammenspiel mit dem verwaltenden Unternehmen und ist insbesondere im Rahmen der SoS-Gestaltung aus einem Unternehmen heraus (Konzept 5) relevant.

4.4.3 Musterbasierte Gestaltung

Um die SoS-Gestaltung mit Lösungswissen zu unterstützen, muss ein geeigneter Wissensmanagementprozess etabliert werden. Im Rahmen der Arbeit werden drei wesentliche Handlungsoptionen aufgegriffen, mit Lösungswissen zu interagieren. Diese sind vor, während und nach Gestaltungsaktivitäten relevant. Die Handlungsoptionen sind Ergänzungen zu den Tätigkeiten der Gestaltung. Sie können voneinander getrennt genutzt werden. Für einen größtmöglichen Nutzen sollten jedoch alle drei Handlungsoptionen berücksichtigt und vor, während und nach der Gestaltung eingesetzt werden. Bild 4-30 ordnet die drei Handlungsoptionen im Rahmen der musterbasierten Gestaltung ein.

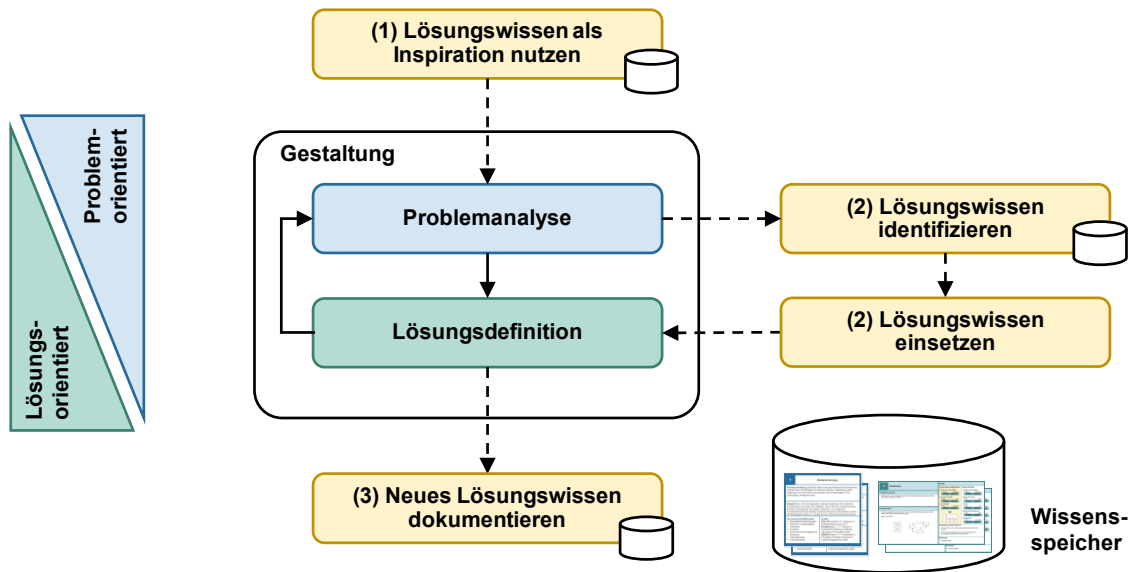


Bild 4-30: Wissensmanagement zur Unterstützung der Gestaltung

Handlungsoption (1) beschreibt die Nutzung von Lösungswissen als Inspiration für Gestaltungstätigkeiten. Diese können neue Ideen für SoS bieten oder neue Handlungsweisen vorschlagen, die zu einem bestimmten Ergebnis führen. Hierzu werden die verfügbaren Wissens Elemente analysiert und für eine mögliche Nutzung geprüft.

Handlungsoption (2) beschreibt die Nutzung von Lösungswissen während der Gestaltung. Entsprechend dem Vorgehen nach ANACKER ist die Grundlage für die Identifikation von Lösungswissen eine initiale Problembeschreibung (vgl. Kapitel 3.3.2.3) [Ana15]. Neben den daraus resultierenden Problemstellungen ist auch das entsprechende Gestaltungskonzept, in dem sich die Gestaltung einordnet, ein Indikator für die Suche nach relevantem Lösungswissen. Die eingeführte Strukturierung und Zuordnung des Lösungswissens zu den einzelnen Merkmalen von SoS helfen dabei, für die jeweiligen Herausforderungen relevantes Lösungswissen zu identifizieren. Sowohl Designprinzipien als auch Lösungsmuster können für die unterschiedlichen Herausforderungen eine Grundlage der Lösungsfindung und -definition darstellen. Sie können unterschiedliche Lösungswege aufzeigen und zu alternativen Umsetzungsoptionen führen, die es im Anschluss gegeneinander zu bewerten gilt. Neben der Adaption von Lösungswissen kann auch die Kombination von Lösungswissen zu konkreten Lösungen führen.

Für eine nachhaltige Etablierung des Musteransatzes ist die **Handlungsoption (3)** essenziell. Diese beschreibt die Dokumentation von neuem Lösungswissen, welches sich durch die spezifischen Gestaltungstätigkeiten ergibt. Dabei können Vergleiche mit weiteren Umsetzungen wiederkehrende Lösungen aufzeigen. Um das Wissen für eine breite Masse an Problemstellungen nutzbar zu machen, sollte es entsprechend generalisiert werden. Für die Dokumentation dienen das SoS-Lösungsmustertemplate oder der Designprinzipien-Kurzsteckbrief als Grundlage.

Ein weiterer Erfolgsfaktor zur Etablierung des musterbasierten Ansatzes ist der Wissenspeicher. Dieser muss die Wissens Elemente aufnehmen, speichern und auf Anfrage bereitstellen. Der Austausch mit der Wissensbasis erfolgt anhand des beschriebenen Wissensmanagementansatzes (Bild 4-31). Dabei ist insbesondere die Aufnahme von neuen Wissens Elementen wichtig für die Etablierung und zukunfts fähige Nutzung des Wissensspeichers. In einer ersten Ausbaustufe erfolgte die Publikation von Lösungsmustern und Designprinzipien im Rahmen von Konferenzbeiträgen. Diese sind als öffentliche Beiträge unternehmensübergreifend verfügbar und erfüllen somit die Anforderung einer unternehmensübergreifenden Wissensbasis im Rahmen von SoS. Literaturdatenbanken werden kontinuierlich mit neuem Wissen befüllt. Ein Beispiel sind die Resilienz-Prinzipien nach AGARWAL et al., die die Auswirkungen von äußeren Einflussfaktoren reduzieren [AEL+23, S. 3]. Ein Nachteil ist die uneinheitliche Strukturierung der Wissens Elemente in einer Literaturdatenbank. Eine sinnvolle nächste Ausbaustufe stellt daher ein öffentlich zugängliches Wiki dar, welches der einheitlichen Strukturierung der Lösungsmuster und Designprinzipien folgt. Auch eine mitgliedsgeführte Datenbank innerhalb von Gremien wäre ein sinnvolles Format. Im Rahmen dieser Arbeit wurden diese Optionen aufgrund von begrenzten Ressourcen nicht weiter verfolgt.

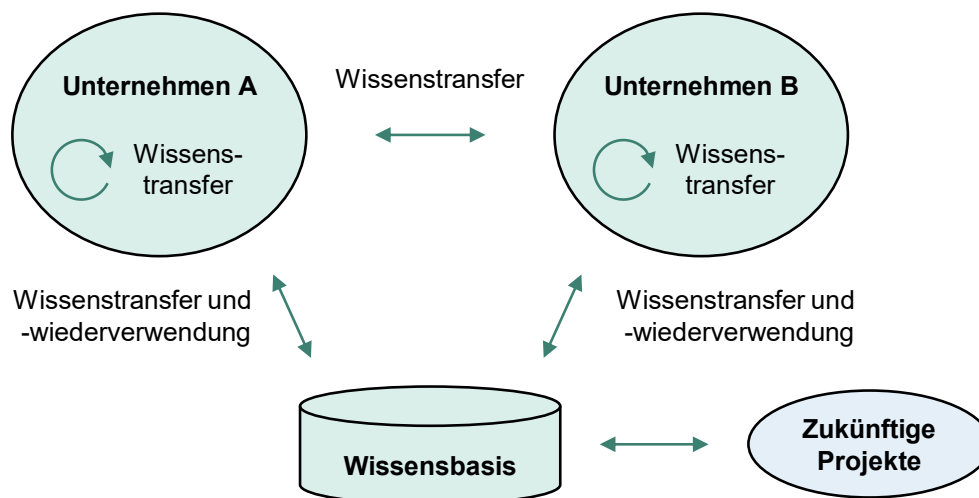


Bild 4-31: Zusammenhang zwischen Gestaltungstätigkeiten und der Wissensbasis (in Anlehnung an KAHLERT et al. [KLM+04, S. 425])

5 Demonstration und Evaluation

In diesem Kapitel wird die Anwendung des in Kapitel 4 vorgestellten *Rahmenwerks zur musterbasierten Gestaltung von System of Systems* anhand von zwei Praxisbeispielen demonstriert und anhand der in Kapitel 2.6 gestellten Anforderungen evaluiert. Hierzu wurde entsprechend dem Vorgehen nach PEFFERS et al. die Lösung angewandt und anhand von Feedback der Anwender qualitativ evaluiert, ob die Anforderungen erfüllt werden und die Anwendbarkeit und Effektivität der Lösung gegeben ist (vgl. Kapitel 1.3). Um die Unterstützung des Rahmenwerks für unterschiedliche Klassen von SoS zu überprüfen, wurde das Rahmenwerk in drei unterschiedlichen Beispielen angewandt, von denen im Folgenden auf zwei Beispiele detailliert eingegangen wird. Diese weisen unterschiedliche Charakteristika von SoS auf und demonstrieren unterschiedliche Ausprägungen des Rahmenwerks. Das dritte Beispiel wird im Rahmen der Bewertung anhand der zuvor gestellten Anforderungen aufgegriffen und kurz zusammengefasst. Die Anwendung und Evaluation erfolgte im Rahmen von interaktiven Workshops mit Unternehmensvertretern.

In **Kapitel 5.1** wird das Beispiel eines *automatisierten Überladevorgangs im Systemverbund aus Mähdrescher und Traktor mit Überladewagen* beschrieben. **Kapitel 5.2** beschreibt das Beispiel einer *systemübergreifenden Fehlerkommunikation innerhalb einer Industrie 4.0-Fertigungslinie*. Abschließend erfolgt in **Kapitel 5.3** eine zusammenfassende Betrachtung und Bewertung des Lösungsansatzes anhand der Anforderungen.

5.1 Anwendungsbeispiel 1: Automatisierter Überladevorgang im Smart Farming

Das erste Anwendungsbeispiel betrachtet die automatisierte Interaktion von Systemen innerhalb eines Ernteprozesses. Ziel ist es, die Arbeitsprozesse in der Landwirtschaft und Intralogistik der Erntegüter durch die Optimierung der Zusammenarbeit von mehreren Systemen zu erreichen. Die Implementierung eines intelligenten Managements der Transportkette auf und abseits des Feldes soll die Auslastung der Erntemaschinen und damit die Effizienz des Ernteprozesses erhöhen. Im konkreten Beispiel wurde der automatisierte Überladevorgang betrachtet, bei dem Mähdrescher sowie Traktor mit Überladewagen (Feldtransporter) in Abstimmung miteinander optimale Überladevorgänge durchführen sollen. Die Ergebnisse werden aus Sicht der an der Gestaltung beteiligten Unternehmen vorgestellt. Dabei wird insbesondere auf die Entscheidungen eines Landmaschinenherstellers sowie den direkten Kooperationspartnern eingegangen. Zum besseren Verständnis und aus Geheimhaltungsgründen werden einzelne Details abstrahiert und auf die wesentlichen Zusammenhänge beschränkt. Dies erhöht gleichzeitig das Verständnis zur Anwendung des Rahmenwerks. Die nachfolgende Darstellung präsentiert ein potenzielles Umsetzungskonzept und stellt keine tatsächlich am Markt erhältliche Lösung der Unternehmen dar.

Für den Überladevorgang muss der Feldtransporter parallel an den Mähdrescher heranfahren, um das Erntegut vom Mähdrescher aufzunehmen. Dies geschieht bei voller Fahrt, während der Mähdrescher den Ernteprozess kontinuierlich fortführt. Dabei müssen zahlreiche Faktoren berücksichtigt werden, wie z. B. das Ladevolumen und der jeweilige Füllzustand der beteiligten Maschinen oder auch die Erntefläche und die geplante Fahrtrajektorie des Mähdreschers. Zusätzlich kann die Anzahl der an dem Prozess beteiligten Mähdrescher und Transportsysteme variieren. In diesem dynamischen Systemverbund soll daher der Überladevorgang auf Basis der jeweiligen situativen Begebenheiten optimal realisiert werden. Hierzu sollen die jeweils eigenständigen Systeme um SoS-Funktionen erweitert werden, sodass diese im Verbund interagieren können. Bild 5-1 visualisiert ein erstes Konzeptbild in einem frühen Entwicklungsstadium für das Zusammenspiel der Systeme auf dem Feld.

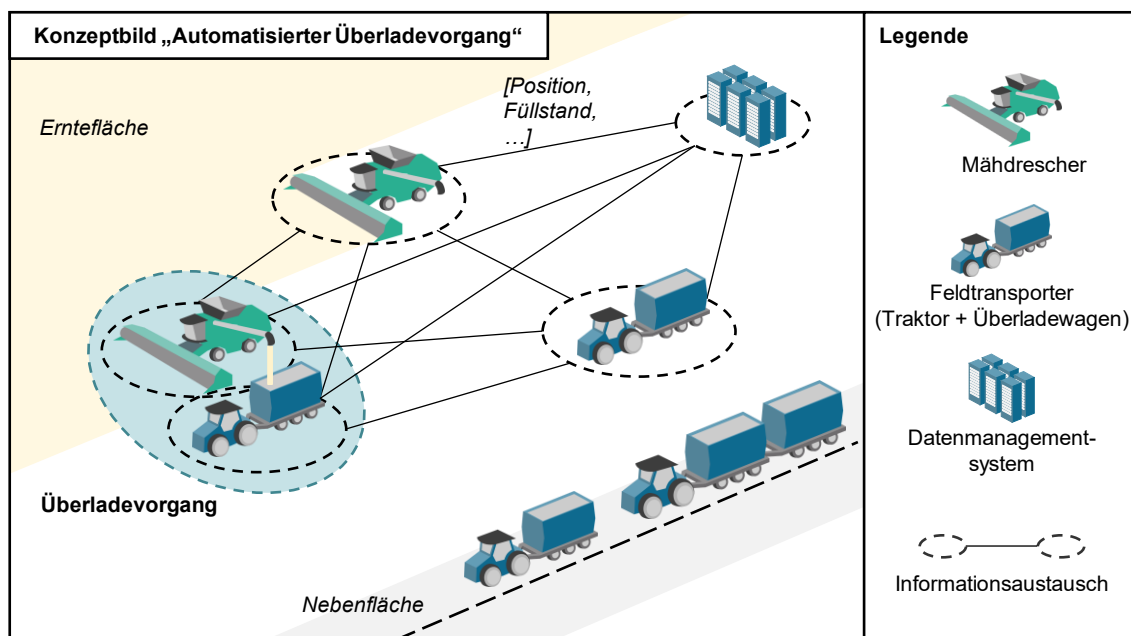


Bild 5-1: Übersicht Anwendungsbeispiel „Automatisierter Überladevorgang“

Im Folgenden wird die Anwendung des Rahmenwerks zur musterbasierten Gestaltung von System of Systems anhand der Konzeptentwicklung des automatisierten Überladevorgangs gezeigt.

5.1.1 Phase 1: Grobanalyse

Ausgangspunkt für die musterbasierte Gestaltung sind die initialen Informationen und identifizierten Bedarfe von Produktmanagern sowie erste Ideen und Konzepte aus Vorprojekten und Forschungsarbeiten, die in Kooperation mit mehreren beteiligten Unternehmen entwickelt wurden. Hieraus wurde ein Zielbild abgeleitet, welches die Optimierung des Überladevorgangs zwischen Mähdreschern und Feldtransportern zum Ziel hat. Konkret sollen drei Ziele verfolgt werden. Zum einen soll die *Zeit und Auslastung der*

einzelnen Systeme optimiert werden. Mähdrescher haben hohe Anschaffungs- und Betriebskosten und kommen in nur wenigen Wochen eines Jahres zum Einsatz. Das Zeitfenster ist eng, da das Getreide zu einem passenden Zeitpunkt abgeerntet werden muss, ohne Qualitäts- und Ernteverluste zu erleiden. Gleichzeitig ist die Ladekapazität des Korntanks begrenzt, was bedeutet, dass das Erntegut bereits nach kurzer Zeit vom Mähdrescher abgetankt und über Transportsysteme verteilt werden muss. Die effiziente Nutzung von Mähdreschern während der Erntezeit setzt daher voraus, dass ein möglichst effizienter Austausch von Erntegut stattfindet. Dies geschieht in der Regel bei laufender Fahrt ohne Unterbrechung des Erntevorgangs. Zum anderen sollen die *Fahrer der Landmaschinen entlastet* werden. Zeit- und Kostendruck sowie die herausfordernden Tätigkeiten während des Ernteprozesses sorgen für steigende Müdigkeit, was die Produktivität der beteiligten Akteure mindert. Die Verbesserungen durch mehr Automatisierung fördern die Effizienz und entlasten Landwirte. Darüber hinaus führt der Fachkräftemangel auch in der Landwirtschaft zu Schwierigkeiten, qualifiziertes Personal für den Betrieb zu finden. Die Abstimmung zwischen Feldtransportern und Mähdreschern beim Überladevorgang setzt viel Erfahrung und Know-How voraus. Durch die Automatisierung des Vorgangs soll auch *weniger qualifiziertes Personal* eingesetzt werden können.

Um das Vorhaben zu skizzieren, werden zunächst die vorhandenen Informationen sowie die ersten Konzeptideen aus den Vorarbeiten im Grobkonzeptsteckbrief festgehalten (Bild 5-2). Dazu werden u. a. die initial beteiligten Systeme aufgelistet, die notwendig sind, um die Anwendungsfälle und die damit einhergehenden Funktionen zu realisieren. Die gesammelten Informationen sind eine geeignete Basis zur Beantwortung der drei im Konzeptsteckbrief gestellten Fragen. Wie in dem Grobkonzept festgehalten, sind neben Mähdreschern auch Traktoren und Überladewagen beteiligt, die von unterschiedlichen Unternehmen hergestellt und gemanagt werden. Die Systeme können zusätzlich außerhalb des automatisierten Überladevorgangs eingesetzt werden und erfüllen somit jeweils eigene Zwecke. Der Mehrwert wird durch die optimale Abstimmung der Systeme und die Reduzierung der Standzeiten erreicht. Dabei können Effizienzsteigerungen erreicht werden, die nicht durch die Optimierung des Betriebs eines einzelnen Systems erzielt werden können. Im Fall des *Automatisierten Überladevorgangs* sind somit die Fragen mit „Ja“ zu beantworten. Hieraus wird abgeleitet, dass die wesentlichen Merkmale eines SoS erfüllt sind. Im Folgenden wird der Anwendungsfall mithilfe der musterbasierten Gestaltung weiterentwickelt.

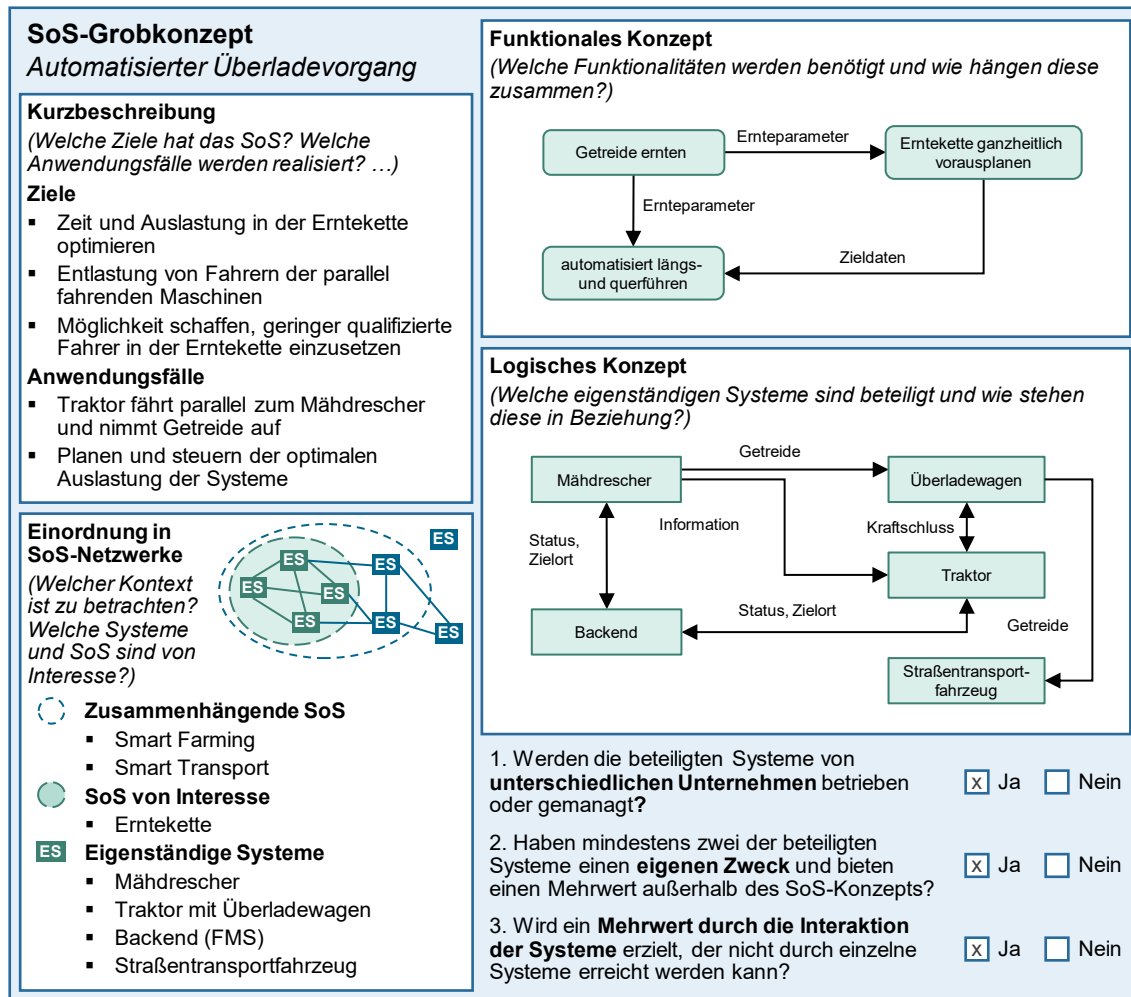


Bild 5-2: SoS-Grobkonzept „Automatisierter Überladevorgang“

5.1.2 Phase 2: Ermittlung SoS-Gestaltungskonzept

Das Grobkonzept ist die Ausgangsbasis zur Ermittlung eines passenden SoS-Gestaltungskonzepts. Hierzu wird ein SoS-Quick-Check mithilfe der in Kapitel 4.3.3 vorgestellten Leitfragen durchgeführt. Da die initialen Ideen lösungsoffen im Rahmen von Vorprojekten diskutiert wurden, waren keine Strukturen vorgegeben und gleichzeitig mehrere Strukturen denkbar. Auch die Fähigkeit der Automatisierung war nicht vollständig bekannt. Daher werden zunächst die Fragen 1 bis 3 mit „Nein“ beantwortet. Auf Basis des Grobkonzepts wird deutlich, dass mehrere Unternehmen bei der Umsetzung des Konzepts benötigt werden und keines dieser Unternehmen die alleinige Kontrolle über die beteiligten Systeme hat. Das Konzept wird initial von einem Hersteller für Mähdrescher und Traktoren sowie Herstellern für Überladewagen vorangetrieben, die in enger Kooperation stehen. Die Frage 4 ist folglich mit „Nein“ zu beantworten. Entsprechend dem Entscheidungsbaum ist bereits eine eindeutige Auswahl eines passenden Gestaltungskonzepts

möglich und daher die Fragen 5 bis 7 irrelevant für die Entscheidungsfindung. Ein Überblick über die Leitfragen und die Beantwortung für den *Automatisierten Überladevorgang* wird in Bild 5-3 dargestellt.

SoS-Quick-Check (Leitfragen zur Konzeptauswahl)		
1. Sind bereits wesentliche Strukturen zur Umsetzung des Grobkonzepts festgelegt?	☐ Ja ☒ Nein	Wenn Ja, welche? -
2. Ist das SoS bereits in Betrieb oder bestehen bereits wesentliche Interaktionsmechanismen für das SoS?	☐ Ja ☒ Nein	Wenn Ja, welche? -
3. Gibt es Standards oder Protokolle für Schnittstellen, die eine Kommunikation zwischen den beteiligten Systemen ermöglichen können?	☐ Ja ☒ Nein	Wenn Ja, welche? -
4. Wird das Projekt aus Sicht eines einzelnen Unternehmens durchgeführt oder hat ein einzelnes Unternehmen die Kontrolle über das SoS?	☐ Ja ☒ Nein	Wenn Ja, welche? -
5. Sind die beteiligten Unternehmen bei der Umsetzung des Konzepts bekannt und festgelegt?	☐ Ja ☐ Nein	→ Wenn Ja, weiter mit Frage 6 → Wenn Nein, weiter mit Frage 7
6. Sind die einzelnen Systeme bekannt und können diese beeinflusst bzw. angepasst werden? ☐ Ja (White Box, Grey Box) ☐ Nein oder nur teilweise (Black Box) ☐ Unbekannt	7. Welches Ziel wird verfolgt?? ☐ Ziel 1 (Know-How Schutz) ☐ Ziel 2 (Leistungsoptimierung) ☐ Ziel 3 (Standardisierung) ☐ Ziel 4 (Kompatibilität) ☐ Ziel 5 (Kundennutzen) ☐ Ziel 6 (unbekannt)	

Bild 5-3: Leitfragen zur Ermittlung des SoS-Gestaltungskonzepts für den „Automatisierten Überladevorgang“

Die Beantwortung der Fragen ermöglicht die Auswahl eines für den Anwendungsfall passenden Gestaltungskonzepts. Anhand des Entscheidungsbaums wird das Gestaltungskonzept 1 „SoS-Initialisierungsgestaltung“ identifiziert und ausgewählt (Bild 5-4). Im Folgenden wird entsprechend den Leitlinien und Unterstützungen des Gestaltungskonzepts 1 der *Automatisierte Überladevorgang* weiter ausgearbeitet.

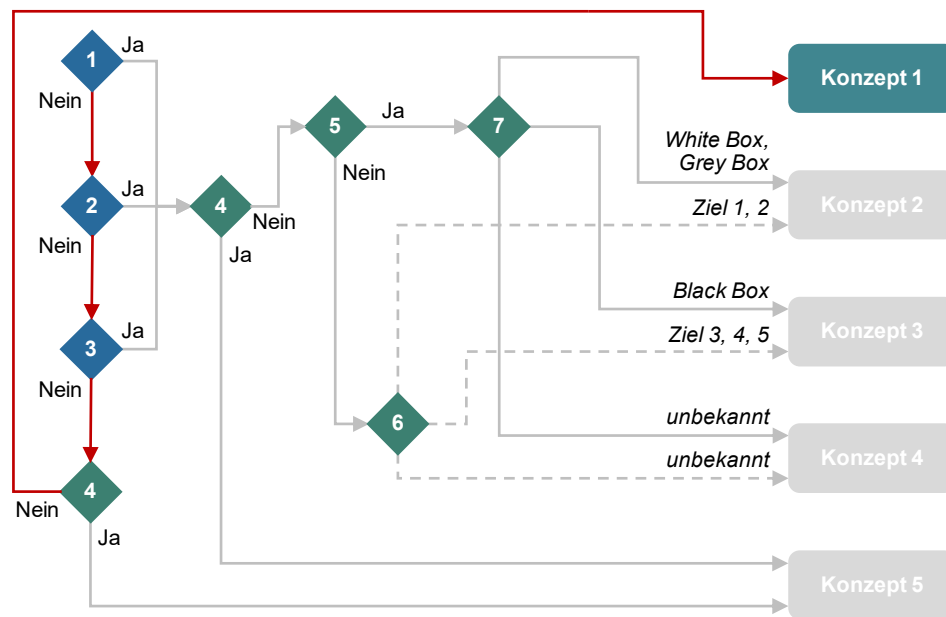


Bild 5-4: Anwendung des Entscheidungsbaums für den „Automatisierten Überladevorgang“

5.1.3 Phase 3: Musterbasierte Gestaltung

Das Ziel von Gestaltungskonzept 1 ist die Ideenfindung und initiale Konzipierung einer neuen SoS-Lösung. Dafür wird die initiale Idee systematisch anhand der Eingangsinformationen verfeinert und ein erstes Konzept für die Umsetzung abgeleitet. Hierzu wird die Methode „Initiales SoS-Konzept“ (M1-1) angewandt, die entsprechend im Gestaltungskonzept 1 als empfohlene Methode vorgeschlagen wird (vgl. Kapitel 4.3.2).

Im ersten Schritt werden die **Ziele ermittelt** und ggf. erweitert oder angepasst. Die Ziele liegen häufig in übergeordneten Treibern begründet, sodass zunächst die wesentlichen Treiber im Kontext der Landwirtschaft aufgenommen werden. Ein wesentlicher Treiber in der Landwirtschaft ist die *ökologische Nachhaltigkeit*. Es wird u. a. angestrebt, die natürlichen Ressourcen zu schonen, die Biodiversität zu fördern und negative Auswirkungen auf die Umwelt zu minimieren. Auch der *Fachkräftemangel* spielt als treibende Kraft für Veränderungen in der Landwirtschaft eine Rolle. Da qualifizierte Arbeitskräfte in der Landwirtschaft oft knapp sind, ist es wichtig, Prozesse zu vereinfachen oder zu optimieren, um den Output pro Arbeitskraft zu erhöhen. Der *steigende Nahrungsmittelbedarf* ist ein weiterer treibender Faktor, der beachtet werden muss. Mit der wachsenden Weltbevölkerung steigt auch der Bedarf an Nahrungsmitteln. Es ist daher von großer Bedeutung nachhaltige und effiziente Anbau- und Erntemethoden zu entwickeln, um den steigenden Nahrungsmittelbedarf zu decken, ohne dabei die natürlichen Ressourcen übermäßig zu belasten.

Aus den drei Treibern lassen sich die Ziele begründen, die auch in der Voranalyse bereits hervorgehoben wurden. Die Lösung soll zum einen die *Durchsatzkapazität erhöhen* und

damit sowohl dem Fachkräftemangel entgegenwirken als auch dem steigendem Nahrungsmittelbedarf gerecht werden. Aus der ökologischen Nachhaltigkeit wird das Ziel abgeleitet, die *Bodenverdichtung zu minimieren*. Ein verdichteter Boden verringert die Aufnahmefähigkeit von Wasser und verhindert ein gutes Pflanzenwachstum. Die Lösung muss daher Gewichtungsfaktoren im Zusammenhang mit der ökologischen Nachhaltigkeit beachten. Als letztes Ziel wird die *Reduzierung der Personaler müdung* angestrebt. Dies kann sowohl durch die Verkürzung der Erntedauer aufgrund einer höheren Durchsatzmenge pro Maschine als auch durch die Vereinfachung von Tätigkeiten erreicht werden.

Im Kontext des automatisierten Überladevorgangs gilt es mehrere **Stakeholder zu analysieren**. Zunächst werden die Interessen des Endkunden betrachtet. Im Anwendungsfall ist dies der Landwirt, der u. a. wirtschaftliche Interessen verfolgt und die Ernte möglichst effizient einfahren möchte. Der Landwirt kann gleichzeitig als Fahrer der Landmaschinen aktiv am Anwendungsfall beteiligt sein und die Ernte durchführen. Diese Nutzerrolle kann aber auch durch beauftragte Fahrer oder Lohnunternehmer eingenommen werden. Fahrer sind an einer komfortablen und gleichzeitig leicht zu bedienenden und effizienten Lösung interessiert, um die Ernteaufgabe gut durchführen können. Darüber hinaus sind als Stakeholder die Hersteller für Landmaschinen und Transportlösungen zu betrachten. Diese haben ein Interesse daran, den Kunden zufriedenzustellen und gleichzeitig einen Gewinn zu erwirtschaften. Dabei müssen sie sich gegenüber Konkurrenten behaupten und Vorteile, wie eine hohe Qualität oder einen größeren Kundennutzen, bieten. Auch der Gesetzgeber tritt als Interessensvertreter auf und setzt durch Gesetze für Umweltschutz oder Betriebssicherheit die Rahmenbedingungen für technische Lösungen.

Durch die Analyse der Stakeholder und deren Interesse wird deutlich, dass insbesondere für Hersteller von Landmaschinen die Konkurrenzsituation für die Ausgestaltung des automatisierten Überladevorgangs hervorzuheben ist. Die *Wettbewerbsfähigkeit Mehrwert durch eigene Lösungen*, die sich von der Konkurrenz abheben sowie der *Schutz von Know-How* werden dabei als wichtige Faktoren angesehen. Dies führt dazu, dass zunächst das SoS-Konzept mit den initialen Unternehmenspartnern weiterverfolgt und lediglich ein Landmaschinenhersteller in der Analyse berücksichtigt wird.

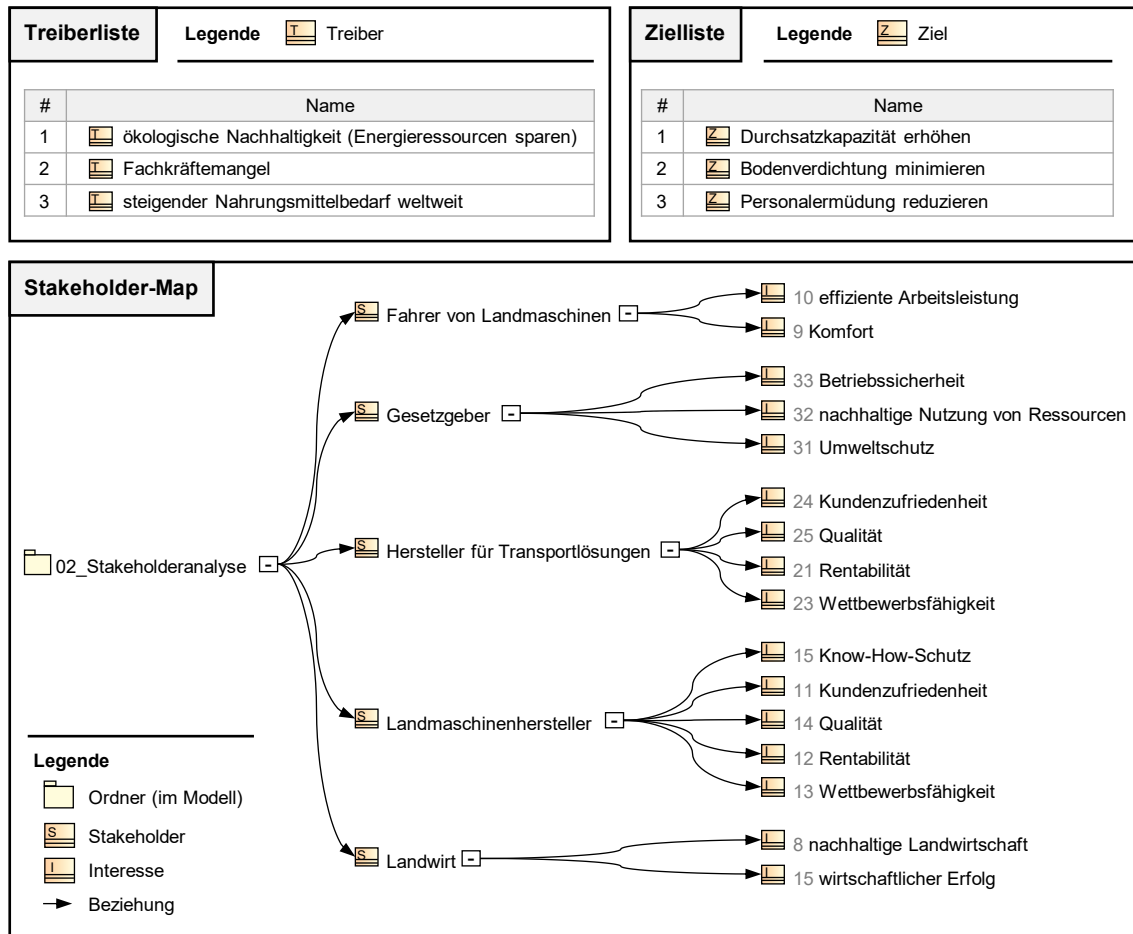


Bild 5-5: Ergebnisse der problemorientierten Aktivitäten

Auf Basis der problemorientierten Analyse wird im nächsten Schritt die Lösung weiterentwickelt. Um die **Lösungsfindung** zu unterstützen, wird entsprechend dem Vorgehen der musterbasierten Gestaltung der in Kapitel 4.4.2 eingeführte Lösungsmusterkatalog herangezogen. Die dort beschriebenen Lösungsmuster beschreiben Lösungen, die sich bereits in etablierten SoS bewiesen haben. Anhaltspunkte zur **Identifikation von geeigneten Lösungsmustern** geben die bereits identifizierten Rahmenbedingungen aus den Voruntersuchungen, wie u. a. die *Begrenzung der beteiligten Unternehmen* auf einen Landmaschinenhersteller und einer geringen Zahl an Herstellern für Transportlösungen sowie die *Beachtung der Sicherheitsrichtlinien* im Zusammenhang mit Landmaschinen. Dabei dienen die zuvor genannten Rahmenbedingungen als Indikatoren zur Auswahl eines passenden Musters. Die auszugestaltende Lösung orientiert sich an einem *geringen Managementabhängigkeitsgrad* sowie einer möglichst *geringen Emergenz*, um negative und ggf. sicherheitskritische Situationen zu vermeiden. Dieses Profil spiegelt auch das Lösungsmuster **Zentrale Architektur** wider (vgl. Kapitel 4.4.2). Durch die klare Zuweisung zentraler Planungsaufgaben an ein zentrales System können u. a. Emergenzeffekte vermieden werden. Diese können in dezentralen Systemen durch die verteilte lokale Planung zwischen eigenständigen Systemen auftreten. Durch die lokale Optimierung werden

globale Effekte wahrscheinlicher, da keine ganzheitliche Informationsbasis als Entscheidungsgrundlage dient. Das geplante Merkmalsprofil des *Automatisierten Überladevorgangs* stimmt demnach größtenteils mit der Merkmalsausprägung der zentralen Architektur überein und wird daher als Grundprinzip für den automatisierten Überladevorgang ausgewählt (Bild 5-6).

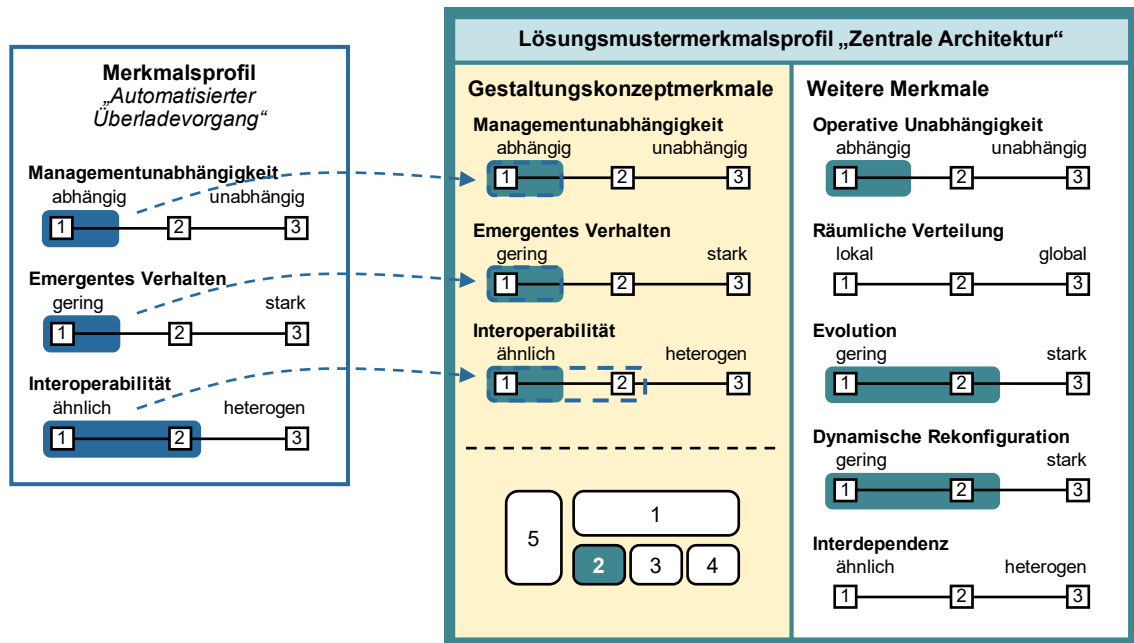


Bild 5-6: Identifikation eines geeigneten Lösungsmusters auf Basis der Merkmalsausprägung des „Automatisierten Überladevorgangs“

Zur **Umsetzungsplanung** werden im Folgenden die wesentlichen Systeme und Zusammenhänge sowie die zur Umsetzung notwendigen Unternehmen festgelegt. Als zentrales System soll ein Backendsystem die Betriebsinformationen, wie Füllstand und Position, der beteiligten Systeme zusammentragen und auswerten. Die Informationen stellen die Basis dar, auf dessen Grundlage u. a. der optimale Zeitpunkt und Ort für den Überladevorgang zwischen einem Mähdrescher und einem Feldtransporter festgelegt und an die entsprechenden Systeme weitergegeben wird. Die zur Bestimmung des optimalen Überladevorgangs notwendigen Algorithmen werden in Zusammenarbeit der beteiligten Unternehmen entwickelt. Hierzu wird eine direkte Kooperation vereinbart, um spezifisches Know-How in Zusammenhang mit Ernteverfahren und Verhaltensparametern zu schützen. Zusätzlich soll mit einem IT-Dienstleister zusammengearbeitet werden, der die entsprechende Infrastruktur zur Umsetzung des Planungssystems zur Verfügung stellt. Für den Datenaustausch soll die Kommunikation über etablierte Kommunikationsinfrastrukturen (WLAN, Mobilfunk) erfolgen und ein spezifisches Datenprotokoll etabliert werden, welches die notwendigen Informationen, wie Positionsdaten und Füllstände, übermittelt.

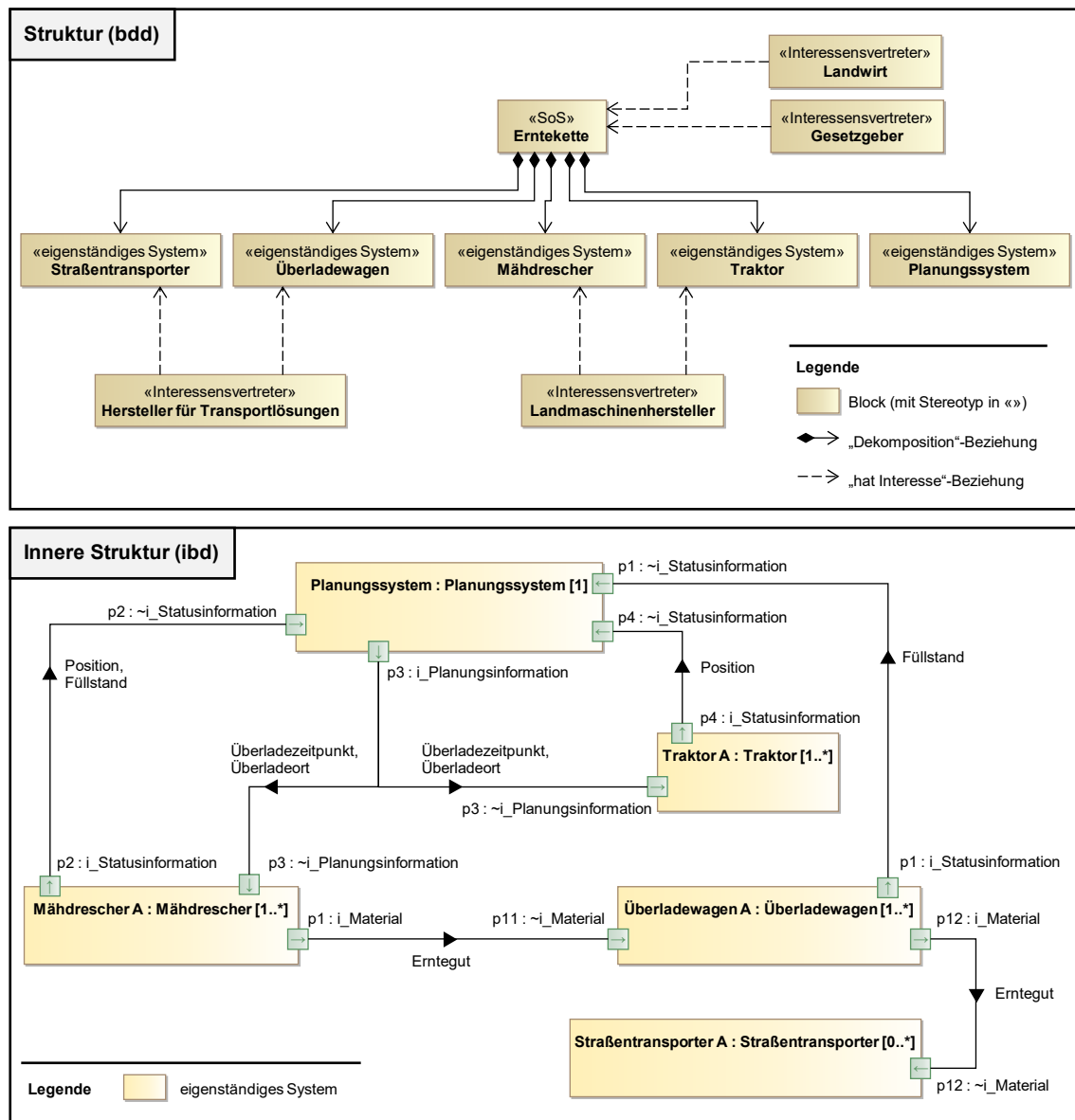


Bild 5-7: Umsetzungskonzept für den „Automatisierten Überladevorgang“

Das vorgestellte Umsetzungskonzept basiert auf dem zentralen Architekturprinzip, in dem das Planungssystem die Informationen der am SoS beteiligten Systeme erfasst und mit dem Ziel eines möglichst effizienten Ernteprozesses auswertet. Das Planungssystem bestimmt den optimalen Überladeort und -zeitpunkt aus den Füllstands- und Positionsdaten, die von den beteiligten Systemen übermittelt werden. Dies soll skalierbar für das Zusammenspiel von mehreren Mähdreschern und mehreren Feldtransportern umgesetzt werden. Dabei kann das Planungssystem optimale Überladevorgänge aus sämtlichen Betriebsparametern planen und bspw. frühzeitige Überladevorgänge durchführen lassen, um ein langfristiges Optimum zu erzielen.

5.1.4 Erweiterte Betrachtung

Die zuvor beschriebene Lösung stellt das initiale Konzept nach Gestaltungskonzept 1 dar. Im Folgenden wird dieses weiter ausgearbeitet. Dabei erfolgt entsprechend der Leitlinien in Gestaltungskonzept 1 der Übergang zu Gestaltungskonzept 2. Grund ist die zuvor getroffene Entscheidung zur direkten Kooperation zwischen Unternehmen in einem geschlossenen SoS. Dementsprechend wird im weiteren Verlauf das Konzept der *Managementgesteuerten SoS-Gestaltung* angewandt.

Durch die Vorarbeit sind bereits erste Entscheidungen getroffen worden, die es im nächsten Schritt weiter auszuarbeiten gilt. Hierzu wird die Methode zur *Konzipierung von abgrenzbaren SoS* (M2-1) angewandt. Diese schreibt zunächst eine fundierte Problemanalyse vor. Wie in Kapitel 4.3.2 beschrieben, kann die Problemanalyse einer Methode auf der Lösungsbeschreibung einer vorangegangenen Methode aufbauen. Dementsprechend werden die **Treiber-, Stakeholder- und Feldanalysen** aus Kapitel 5.1.4 als Basis genutzt und durch weitere Untersuchungen ergänzt. Hierzu wird die in Methode M2-1 vorgeschlagene **Analyse der Anwendungsfälle** durchgeführt. Diese gibt Aufschluss über notwendige funktionale Zusammenhänge der beteiligten Systeme. Bild 5-8 zeigt einen Überblick über die untersuchten Anwendungsfälle.

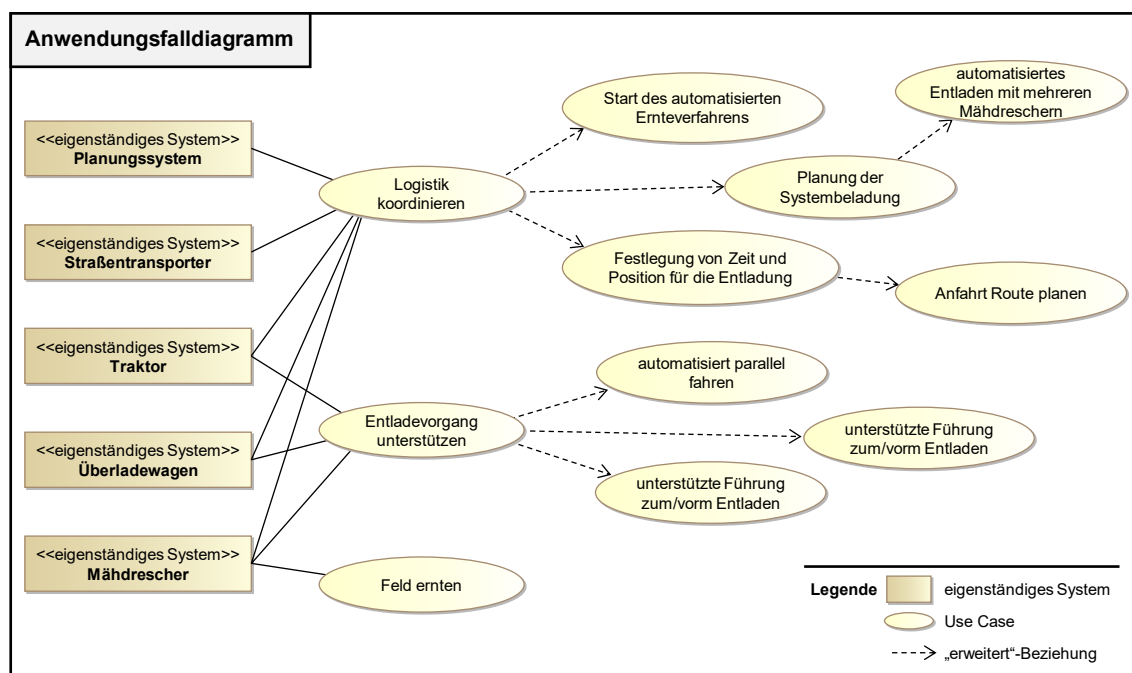


Bild 5-8: Anwendungsfallanalyse des „Automatisierten Überladevorgangs“

Für die Umsetzung des *Automatisierten Überladevorgangs* sind mehrere Anwendungsfälle zu betrachten. Bspw. stellt das Vorgewende¹ eine besondere Situation dar, die ein verändertes Verhalten der Systeme bedarf, indem u. a. der Überladevorgang unterbrochen

¹ Als Vorgewende wird der Randbereich eines Feldes bezeichnet, in dem die Landmaschinen wenden.

werden muss. Um die Komplexität der folgenden Ausführung zu verringern und das Verständnis zu erhöhen, wird auf die wesentlichen Zusammenhänge bis zum Start des Überladevorgangs eingegangen. Teilkonzepte, wie das *Wenden am Vorgewende*, werden nicht weiter ausgeführt.

Das Konzept erfordert, dass die beteiligten Systeme in der Lage sind, automatisiert oder teilautomatisiert zu fahren und eine *automatisierte Parallelfahrt* von Mähdrescher und Feldtransporter unterstützen können. Hierzu müssen die beteiligten Systeme die Längs- und Querführung in gegenseitiger Abstimmung regeln, um während des Überladevorgangs eine zielgenaue Positionierung vom Erntegut im Überladewagen zu realisieren. Führender Anwendungsfall ist der eigentliche *Ernteprozess* des Mähdreschers. Dieser soll kontinuierlich durchgeführt werden, um eine möglichst hohe Auslastung zu gewährleisten. Daher ist auch die *Planung der Anfahrtroute des Feldtransporters* hin zu einem bestimmten Überladeort zu einem bestimmten Zeitpunkt ein wichtiges Element.

Um die Anwendungsfälle zu realisieren, wird im nächsten Schritt das **funktionale Konzept abgeleitet**. Dabei werden etablierte Funktionen berücksichtigt und durch neue Funktionen erweitert, um den automatisierten Überladevorgang möglichst effizient umzusetzen und dabei die Entwicklungskosten der Hersteller niedrig zu halten. Dies hängt mit den wirtschaftlichen Interessen der beteiligten Unternehmen zusammen (vgl. Kapitel 5.1.3). Das funktionale Konzept umfasst sowohl Funktionen zur Initialisierung der beteiligten Systeme und Rahmenbedingungen auf dem Feld als auch Funktionen zur Planung und Durchführung des Überladevorgangs. Diese stehen über Eingangs- und Ausgangsgrößen in Beziehungen zueinander. Zur Festlegung der Überladeposition werden bspw. aktuelle Positionsdaten, die geplanten Routen und die verbleibende Erntefläche ausgewertet, die wiederum Ausgangsgrößen anderer Funktionen sind. So ergibt sich ein funktionales Konzept, welches die grundlegenden Zusammenhänge für den automatisierten Überladevorgang aufzeigt (Bild 5-9).

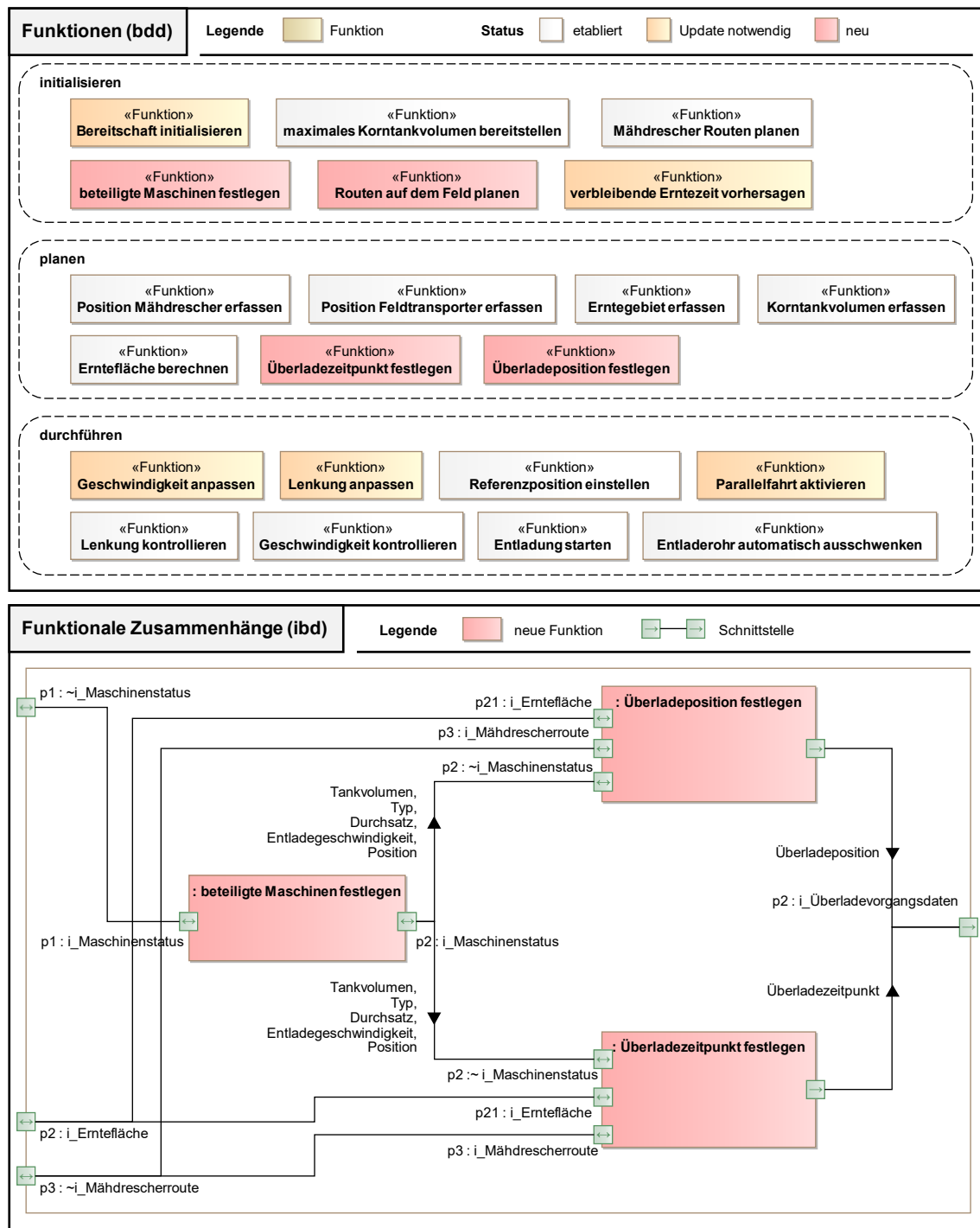


Bild 5-9: Funktionales Konzept inkl. funktionaler Zusammenhänge (Auszug)

Das funktionale Konzept beschreibt Zusammenhänge unabhängig von den zur Realisierung notwendigen Systemen. Daher wird im nächsten Schritt das **logische Konzept entwickelt** und die Funktionen den Systemen zugeordnet (Bild 5-10). Bei der Verortung muss darauf geachtet werden, dass die Systeme in der Lage sind, die Funktion den Vorgaben entsprechend umzusetzen. Ist dies nicht der Fall, wird ein Update notwendig oder die Funktion muss auf einem anderen System verortet werden. Entsprechend gekennzeichnete *etablierte Funktionen* sind bereits auf Systemen vorhanden und können durch

diese realisiert werden. Darüber hinaus wird das Prinzip der zentralen Architektur verfolgt, sodass eine Verortung der Planungsfunktionen auf das zentrale System erfolgen sollte. Das zentrale System führt relevante Informationen zusammen und managt die Interaktion der Systeme. Daher werden die Funktionen *Überladezeitpunkt festlegen* und *Überladeort festlegen* auf dem Planungssystem verortet. Die Verortung der Funktionen *Geschwindigkeit* und *Lenkung anpassen* auf die Systeme Traktor und Mähdrescher hat zur Folge, dass Systeme ausgeschlossen werden, die keine adaptive Längs- und Querverführung umsetzen können. Sollen ältere Modelle in dem SoS partizipieren, müssten sie durch ein Update der Soft- und/oder Hardware entsprechend nachgerüstet werden.

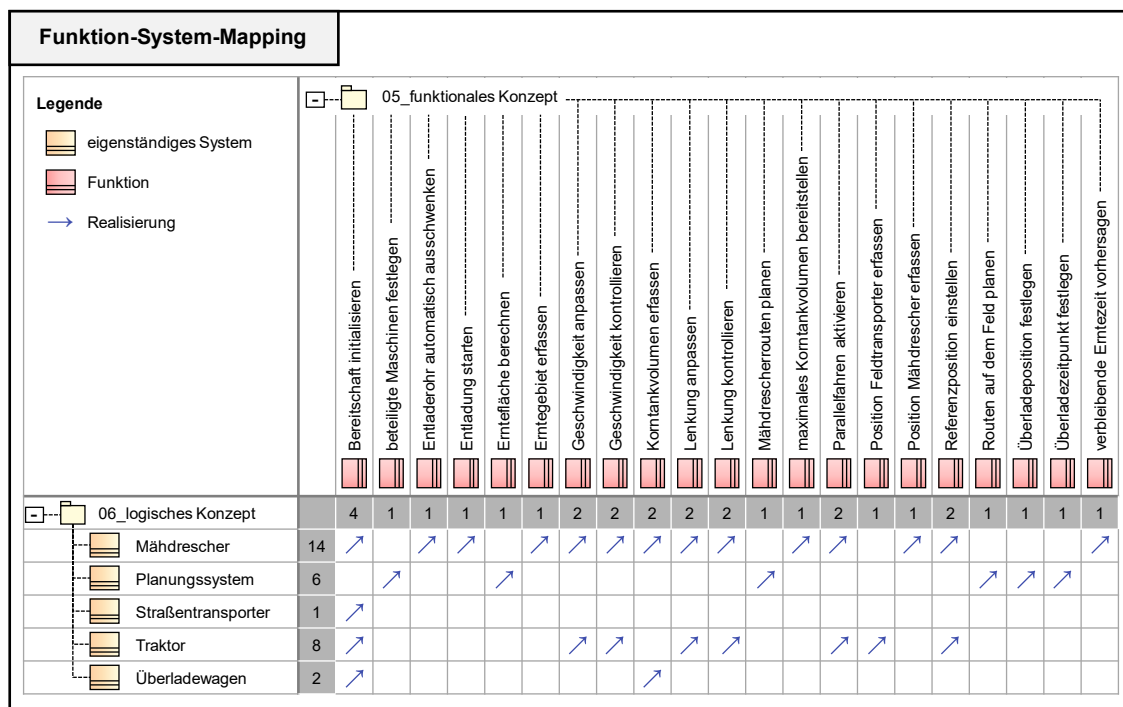


Bild 5-10: Logisches Konzept - Verortung der Funktionen auf die beteiligten Systeme

Auf Basis der Zuordnung können nun Anforderungen an die einzelnen Systeme abgeleitet werden. Insbesondere die neuen Funktionen und durch Updates zu erweiternden Funktionen gilt es durch die beteiligten Unternehmen umzusetzen. Die Zuordnung der Planungsfunktionen auf das zentrale Planungssystem erfordert, dass das System die Vielzahl an Informationen verarbeiten können muss, die durch die beteiligten Systeme bereitgestellt werden. Hierzu ist eine entsprechende Rechenkapazität notwendig, die die Vielzahl an Informationen auswerten kann, um einen optimalen Überladevorgang berechnen zu können. Des Weiteren führt die Verortung der Funktionen auf unterschiedliche Systeme notwendigerweise zu einem Austausch von Informationen, Energie und Material. Die Auslegung der proprietären Schnittstelle für den *Automatisierten Überladevorgang* innerhalb des geschlossenen SoS kann auf Basis der beschriebenen Zusammenhänge abgeleitet werden. Die Schnittstellenspezifikation muss anschließend von den beteiligten Systemen umgesetzt werden, um den Austausch der unterschiedlichen Medien zu ermöglichen.

Die Teilinformationen der einzelnen Schritte stehen im Zusammenhang und können in einem ganzheitlichen Modell zusammengeführt werden. Dies kann insbesondere für die beteiligten Unternehmen einen ganzheitlichen Blick auf den Anwendungsfall bieten. Als Grundlage dienen die in Kapitel 4 beschriebenen Zusammenhänge zwischen den Konzepten. In Anlehnung an BAEK et al. lässt sich ein Informationsmodell ableiten, welches die einzelnen Entwicklungsartefakte in einen übergeordneten Zusammenhang bringt (vgl. Kapitel 3.2.1.6) [BMS+20, S. 338]. Für den automatisierten Überladevorgang werden die Artefakte aus den Überlegungen von Gestaltungskonzept 1 (vgl. Kapitel 5.1.3) und Gestaltungskonzept 2 in Bild 5-11 dargestellt. Zusätzlich wird der Übergang zur unternehmensspezifischen Sicht innerhalb des Gestaltungskonzepts 5 skizziert, auch wenn dieser nicht Teil des Gestaltungskonzepts 2 ist. Die unternehmensspezifische Sicht ist für die Auslegung eines einzelnen Systems relevant und stellt den Bezug zu den zuvor skizzierten Anforderungen an die beteiligten Systeme des SoS her.

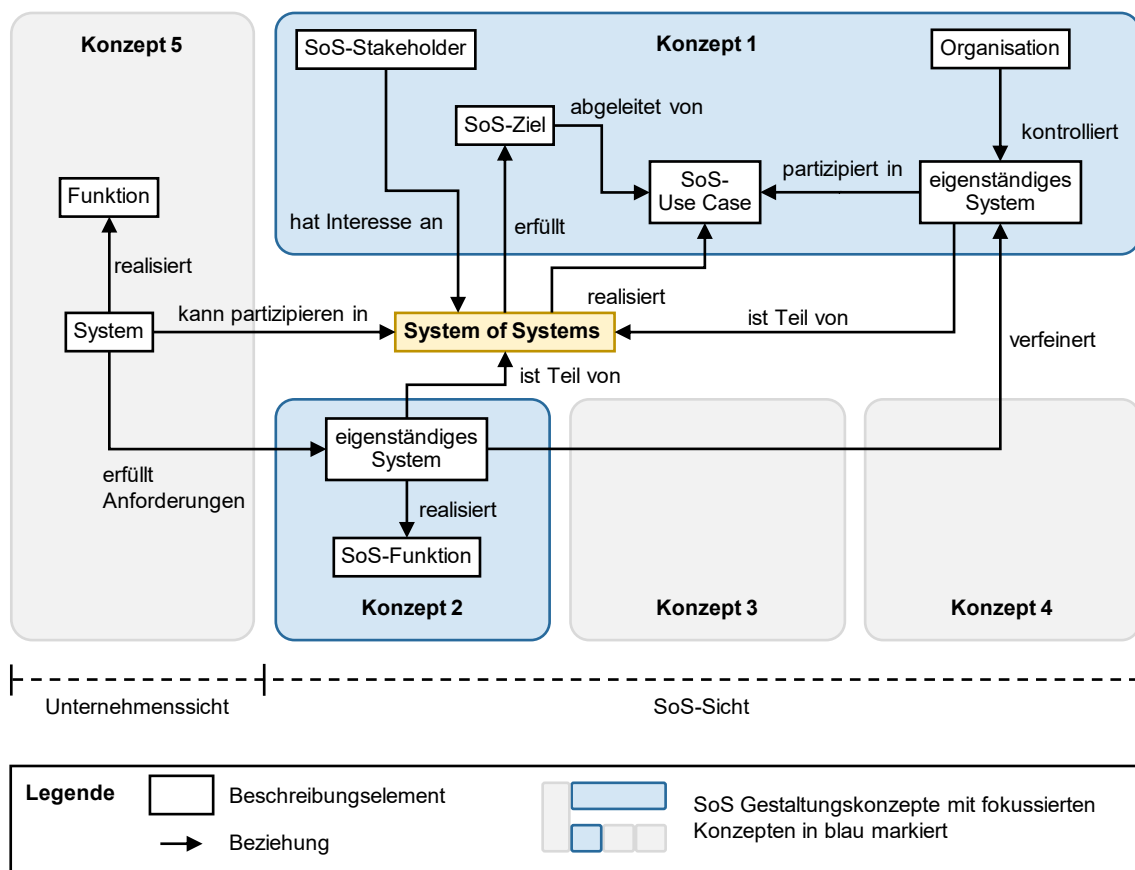


Bild 5-11: Informationsmodell – Zusammenhang der Entwicklungsartefakte

Im Rahmen des Anwendungsbeispiels wurde das Informationsmodell prototypisch umgesetzt. Dazu wurden die Inhalte aus Gestaltungskonzept 1 und 2 in einem übergreifenden SoS-Modell modelliert und in einen entsprechenden Zusammenhang gebracht. Anschließend wurde der Bezug zu einem spezifischen Mähdrescher-Modell hergestellt. Hieraus konnten wesentliche Entwicklungsbedarfe abgeleitet und nachvollzogen werden.

5.2 Anwendungsbeispiel 2: Fehlerkommunikation innerhalb einer Verpackungslinie

Im zweiten Anwendungsbeispiel wird die Entwicklung von Automatisierungslösungen betrachtet, die zum Beispiel in Verpackungslinien eingesetzt werden können. Hierzu wurde in mehreren aufeinander aufbauenden Workshops zusammen mit einem Automatisierungslösungshersteller die Entwicklung von Automatisierungslösungen für Verpackungslinien begleitet. Die Ergebnisse der Workshops basieren zum Teil auf vorgelagerten Arbeiten und enthalten Annahmen, die den Ergebnissen aus Vorprojekten entsprechen. Vorgelagerte Arbeiten sind sowohl Ergebnisse aus Forschungsprojekten, die zusammen mit anderen Organisationen durchgeführt wurden, als auch Erfahrungen aus der Zusammenarbeit mit weiteren Organisationen im Rahmen von Kooperationen. Hierzu gehört insbesondere die Kooperation innerhalb der Interessensverbünde Plattform Industrie 4.0 sowie der OPC Foundation.

Automatisierungslösungen werden an unterschiedlichen Stellen innerhalb von Verpackungslinien eingesetzt, um die Maschinen in Bewegung zu versetzen und die Bewegungen untereinander zu koordinieren. Im Verbund mit anderen Maschinen werden einzelne Teilaufgaben erfüllt, die bei der Verpackung von Objekten notwendig sind. Dabei ist in den letzten Jahren insbesondere die Flexibilität und Anpassung an unterschiedliche Verpackungsabläufe für unterschiedliche Objekte ein wesentlicher Faktor für produzierende Unternehmen. Schwankende Absatzzahlen und Individualisierung sind treibende Kräfte, die häufige Anpassungen erfordern. Der Verbund aus unterschiedlichen Systemen muss sich an die unterschiedlichen Aufgaben anpassen können und dabei übergreifend interagieren.

Im konkreten Beispiel soll eine übergreifende Fehlerkommunikation umgesetzt werden, die es ermöglicht, Fehlermeldungen zwischen benachbarten Maschinen weiterzugeben. Dies erhöht die Transparenz von Fehlerfällen und ermöglicht gezielte Wartungsprozesse. Da innerhalb von Verpackungslinien Maschinen von unterschiedlichen Herstellern eingesetzt werden, ist die Interoperabilität zwischen unabhängigen Maschinen notwendig. Eine mögliche Konstellation einer solchen Verpackungslinie, wird in Bild 5-12 visualisiert.

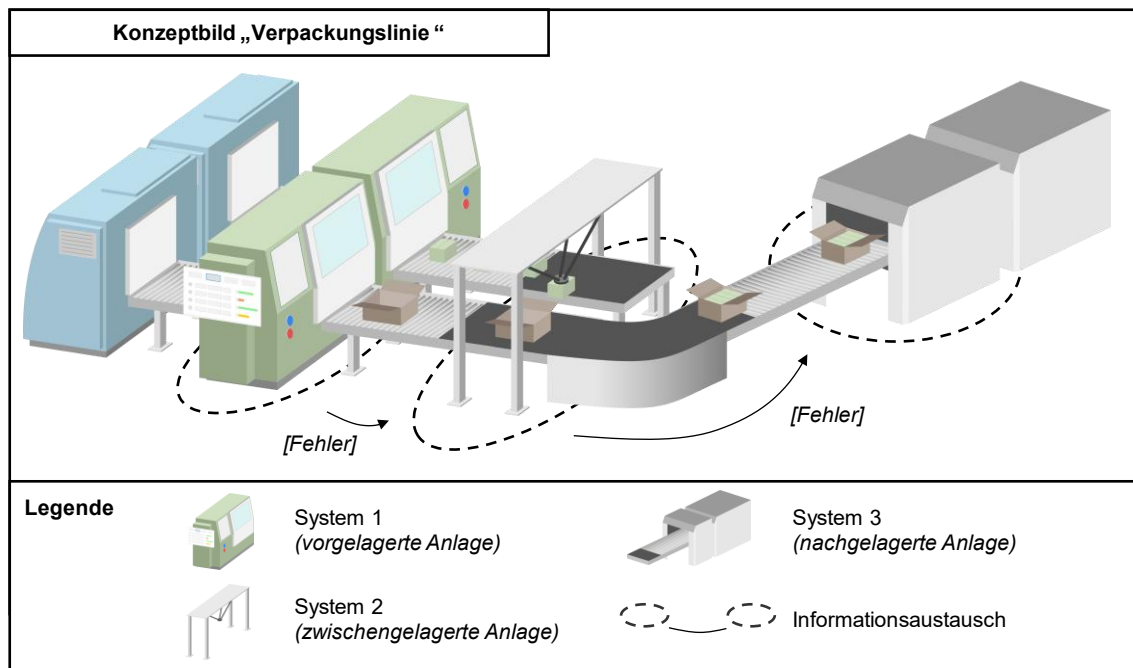


Bild 5-12: Übersicht Anwendungsbeispiel „Verpackungslinie“

5.2.1 Phase 1: Grobanalyse

Entsprechend dem Vorgehen zur musterbasierten Entwicklung wurde zunächst eine erste Analyse mithilfe des SoS-Grobkonzeptsteckbriefes durchgeführt. Als Input wurden die Ergebnisse und Handlungsbedarfe von vorherigen Projekten und vorgelagerten Analysen u. a. von Projektmanagern herangezogen.

Das übergeordnete Ziel ist die *Interoperabilität* zwischen unabhängigen Systemen, die eine Kommunikation von Fehlern in *flexiblen Produktionslinien* zulässt. Dies soll die *Transparenz von Fehlerfällen* erhöhen, indem die Fehlerinformationen über den Maschinenverbund weiterkommuniziert werden und somit auch Folgefehler nachvollzogen werden können. Für Endanwender erhöht sich der Nutzen, indem *Wartungs- und Instandhaltungsprozesse planbarer und zielgerichteter* durchgeführt werden können. Insgesamt sollen drei Anwendungsfälle unterstützt werden. Neben dem Hauptanwendungsfall *Fehlerkommunikation über Maschinen*, soll zusätzlich eine *herstellerunabhängige Auswertung der Fehler* ermöglicht werden. Dies erhöht zum einen die Flexibilität und ermöglicht zum anderen nachfolgenden Maschinen die Anpassung an vorgelagerte Fehlerfälle. Werden in einem vorgelagerten Schritt bspw. Objekte nicht im richtigen Winkel ausgegeben, kann eine nachgelagerte Maschine durch Anpassung des Greifwinkels auf die fehlerhafte Stellung der Objekte reagieren. Als weiteren Anwendungsfall soll das *Plug-and-Play von Maschinen* unterstützt werden, indem Maschinen so vorbereitet sind, dass sie durch einfache Aufgabenübertragung und wenig Zusatzprogrammierung ihre zugewiesene Aufgabe in der Verpackungslinie umsetzen können.

Das Konzept gliedert sich in den Gesamtverbund aus vernetzten Produktionslinien (Industrie 4.0) ein und hat Schnittstellen u. a. mit der vorgelagerten Produktion von Objekten. Dennoch soll der Fokus auf die Verpackung von Objekten gelegt werden, um in einem abgegrenzten Umfang die Anwendungsfälle umzusetzen. Daher werden insbesondere die Verpackungs- und Handlingssysteme von unterschiedlichen Maschinen- und Automatisierungslösungsherstellern betrachtet. Bild 5-13 zeigt das konsolidierte Ergebnis der Grobkonzeptbeschreibung, indem der Umfang der Initiative und ein grobes funktionales und logisches Konzept skizziert wird. Die drei ergänzenden Fragen wurden jeweils mit „Ja“ beantwortet. Dies begründet sich zum einen in dem Ziel, eine herstellerunabhängige Lösung zu finden und dabei unterschiedliche Systeme einzusetzen. Zum anderen wird auch ein Mehrwert erzielt, indem Fehlerzusammenhänge entdeckt und ausgewertet werden können, die aus einer isolierten Betrachtung der einzelnen Systeme nicht erfolgen könnte. Somit liegt ein Konzept vor, welches mithilfe der musterbasierten SoS-Gestaltung weiter ausgearbeitet werden kann.

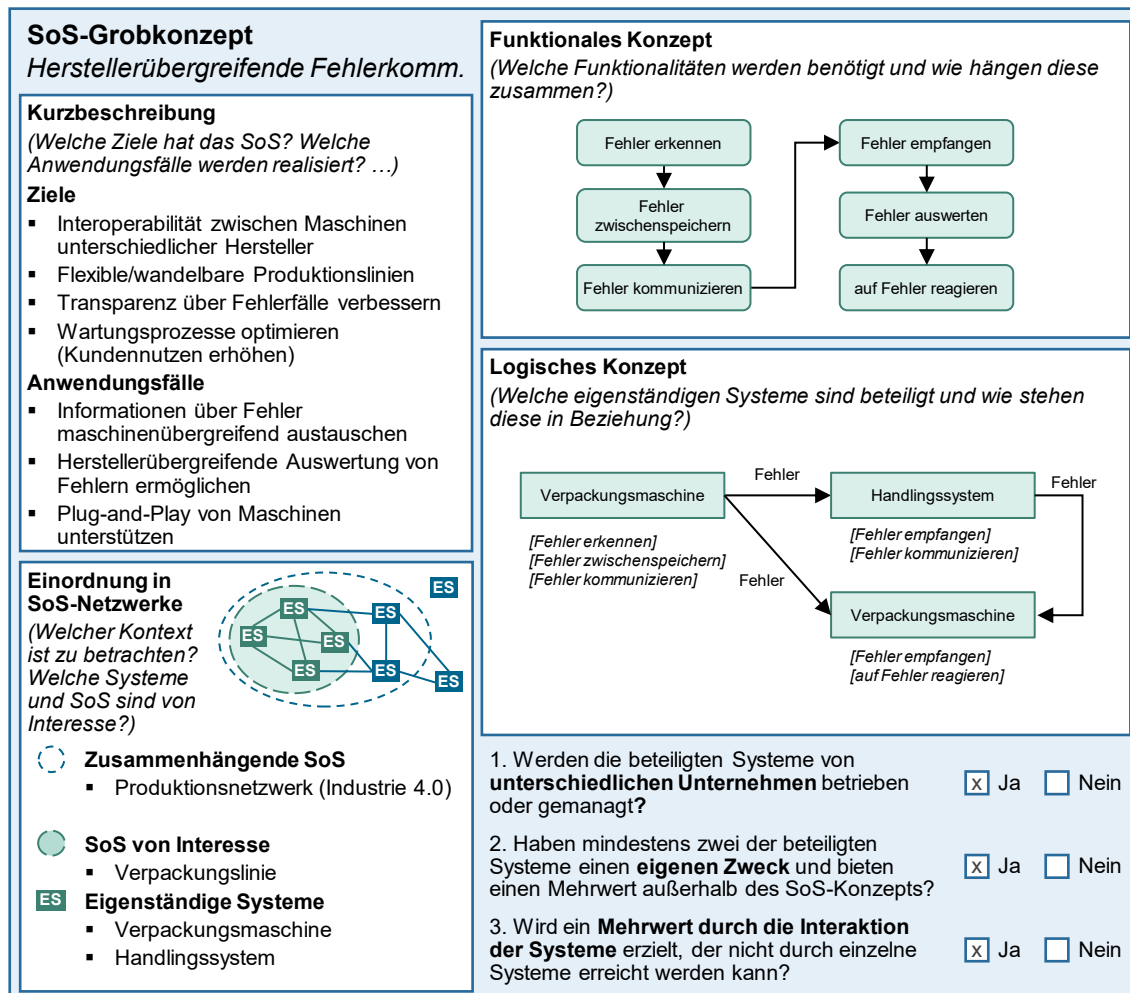


Bild 5-13: SoS-Grobkonzept „Herstellerübergreifende Fehlerkommunikation“

5.2.2 Phase 2: Ermittlung SoS-Gestaltungskonzept

Um ein passendes SoS-Gestaltungskonzept zu ermitteln, wurden zunächst die im Vorgehen definierten **Leitfragen** beantwortet. Im Gegensatz zu Anwendungsbeispiel 1 (vgl. Kapitel 5.1.2) lagen für die geplante Umsetzung bereits erste Grundstrukturen vor, die für die Fehlerkommunikation genutzt werden können. Dazu gehören u. a. etablierte Kommunikationskonzepte, die durch Initiativen wie die Plattform Industrie 4.0 und weitere Interessensverbünde zur Vernetzung von Produktionssystemen bereits entwickelt wurden. Im Speziellen wurde die OPC Unified Architecture (OPC UA) als Standard identifiziert. OPC UA bietet eine Grundlage für den Datenaustausch zwischen unabhängigen Maschinen (vgl. Kapitel 2.2.3). In der OPC Foundation organisieren sich unterschiedliche Unternehmen, wobei die Beteiligung am Gremium für Unternehmen offen ist und durch eine Mitgliedschaft ermöglicht wird. Innerhalb der Foundation werden jeweils Interessengruppen für einzelne Branchen oder Anwendungsgebiete gebildet, um die Interoperabilität in den spezifischen Bereichen zu ermöglichen. Daher sind die Unternehmen zu Beginn einer Initiative nicht vollständig bekannt und mussten auch für die Fehlerkommunikation innerhalb einer flexiblen Verpackungslinie initial zusammengebracht werden. Entscheidend dabei ist ein gemeinsames Ziel, welches die Unternehmen im Anschluss gemeinsam verfolgen. Innerhalb der OPC UA Foundation ist dies allen voran die herstellerübergreifende Interoperabilität durch Standardisierung für einen größtmöglichen Kundennutzen.

Die Beantwortung der Leitfragen und Anwendung der entwickelten **Auswahllogik** resultiert in einem Konzeptvorschlag für den beschriebenen Anwendungsfall (Bild 5-14). Für die herstellerübergreifende Fehlerkommunikation wurde eine Gestaltung anhand des Konzepts 3 „Gremiengesteuerte SoS-Gestaltung“ vorgeschlagen, die im Anschluss für den Anwendungsfall angewendet wurde.

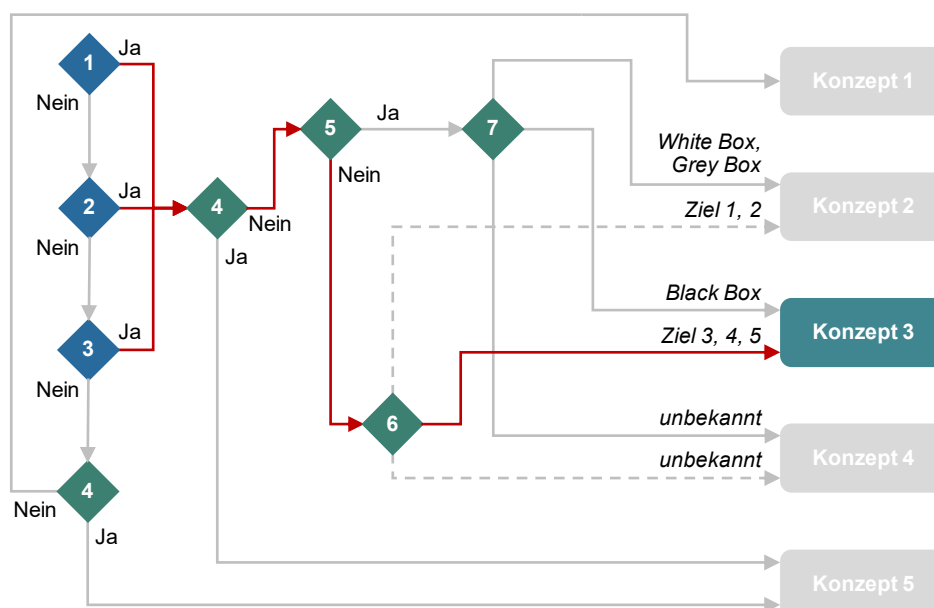


Bild 5-14: Auswahlpfad zur Identifizierung eines SoS-Gestaltungskonzepts für die herstellerübergreifende Fehlerkommunikation

5.2.3 Phase 3: Musterbasierte Gestaltung

Das **Gestaltungskonzept 3** beschreibt die Weiterentwicklung von bestehenden Systemverbünden innerhalb einer Interessengemeinschaft. Dabei werden unterschiedliche Systeme und Interessen berücksichtigt und als Grundlage für die Auslegung der Schnittstellen und des SoS-Verhaltens genutzt. Die zuvor in den Leitfragen identifizierten Grundstrukturen gelten als Basis für das zu entwickelnde Konzept. Zur Gestaltung der herstellerübergreifenden Fehlerkommunikation im Rahmen des Gestaltungskonzepts 3 wird die vorgeschlagene Methode 3-1 „rollenbasierte Konzipierung“ angewandt.

Im ersten Schritt werden entsprechend der Methode die **Ziele und Anwendungsfälle** innerhalb der Interessengemeinschaft analysiert und festgelegt. Wie bereits vorab beschrieben, ist das primäre Ziel die Standardisierung von Schnittstellen und Interaktionsmechanismen, um die Interoperabilität unabhängiger Systeme innerhalb einer Verpackungslinie zu realisieren. Das gemeinschaftliche Ziel wird der Gestaltung und den zu treffenden Gestaltungsentscheidungen als Prämisse zugrunde gelegt. Es sollen insgesamt fünf Anwendungsfälle unterstützt werden, die im Zusammenhang mit der herstellerübergreifenden Fehlerkommunikation stehen. Bild 5-15 fasst die Ergebnisse der Ziel- und Anwendungsfallanalyse zusammen.

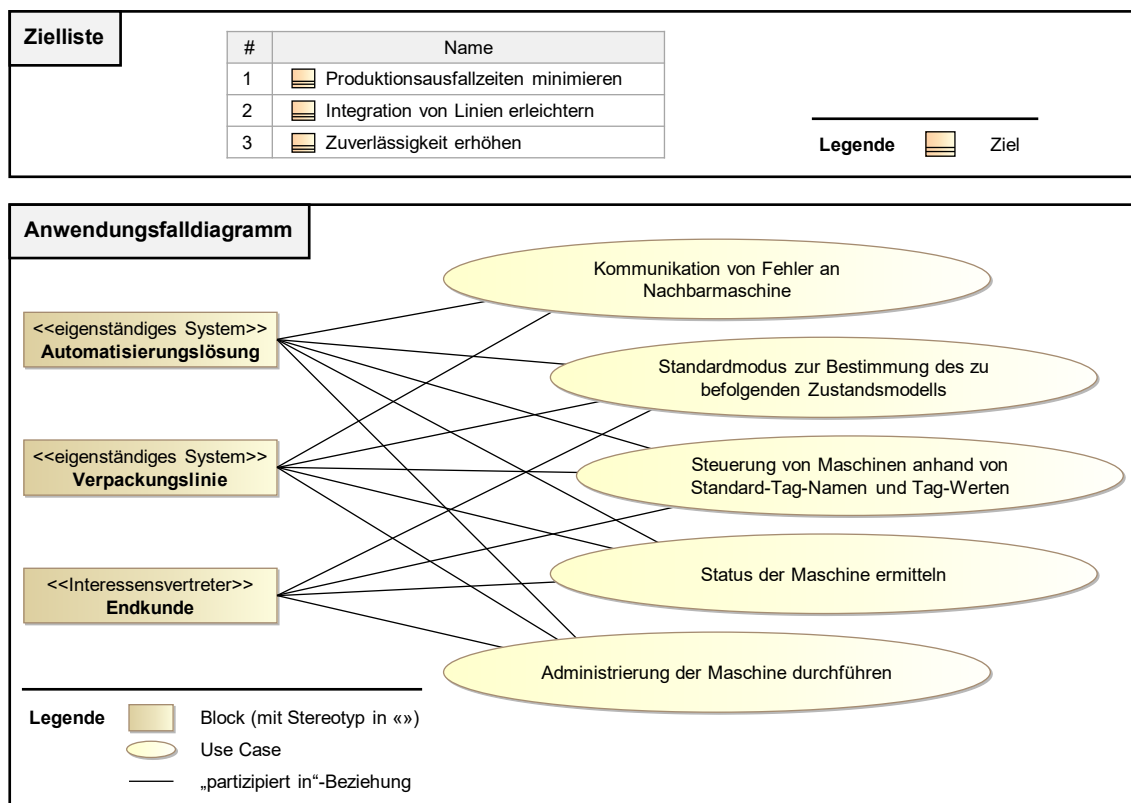


Bild 5-15: Ziele und Anwendungsfälle im Rahmen der „herstellerübergreifende Fehlerkommunikation“

Auch wenn bereits erste Kommunikationsstandards in der Industrie bestehen und auch Fehler kommuniziert werden können, werden die Anforderungen an Flexibilität, Interoperabilität und übergreifende Auswertung der Fehler noch nicht vollständig erfüllt. Um die Ziele zu erreichen und die Anwendungsfälle umsetzen zu können, wird ein konkretes Interaktionskonzept benötigt, welches die abstrakten Standards für die jeweilige Anwendungsdomäne überträgt und verfeinert. Daher werden im Anschluss auf Basis der Anwendungsfälle die zur Umsetzung notwendigen **Fähigkeiten** und **Rollen abgeleitet** sowie der Informationsaustausch im Detail definiert. Dabei liegt der Fokus in den nachfolgenden Ausführungen auf der *Kommunikation von Fehlern an Nachbarmaschinen*, um die grundsätzliche Herangehensweise zu demonstrieren und die Komplexität zu reduzieren. Grundlegende Fähigkeiten zur Fehlerkommunikation, wie die *Erfassung von Fehlern* oder *Bereitstellung von Fehlern*, müssen von jeder Maschine erfüllt werden. Darüber hinaus gibt es Fähigkeiten, die insbesondere den nachfolgenden Maschinen zugeordnet werden. Beispielsweise muss eine erste Maschine in einer Verpackungslinie nicht auf Fehler einer vorgelagerten Maschine reagieren können. Aus der Analyse des Anwendungsfalls und der identifizierten Fähigkeiten lassen sich drei Rollen ableiten: *vorgelagerte*, *zwichengelagerte* und *nachgelagerte Maschine*.

Insgesamt lassen sich elf Fähigkeiten ableiten, von denen sechs Fähigkeiten bestimmten Rollen zugewiesen werden. Durch die Zuordnung ergeben sich gleichzeitig die Austauschbeziehungen sowie die daraus resultierenden Informationsflüsse zwischen den Rollen. Dabei kann die Zusammensetzung innerhalb einer Verpackungslinie flexibel erweitert werden, indem innerhalb der Linie weitere zwischengelagerte Maschinen ergänzt werden. Hieraus resultiert der Bedarf, wesentliche Elemente des Informationsaustausches, wie den Fehlercode, herstellerübergreifend festzulegen. Anhand der Informationsflüsse wird deutlich, welche Informationen in Form von standardisierten Kommunikationsprotokollen festgelegt werden müssen, die von allen beteiligten Maschinen erfüllt werden. Bild 5-16 fasst die Ergebnisse der Fähigkeiten- und Rollenbeschreibung zusammen.

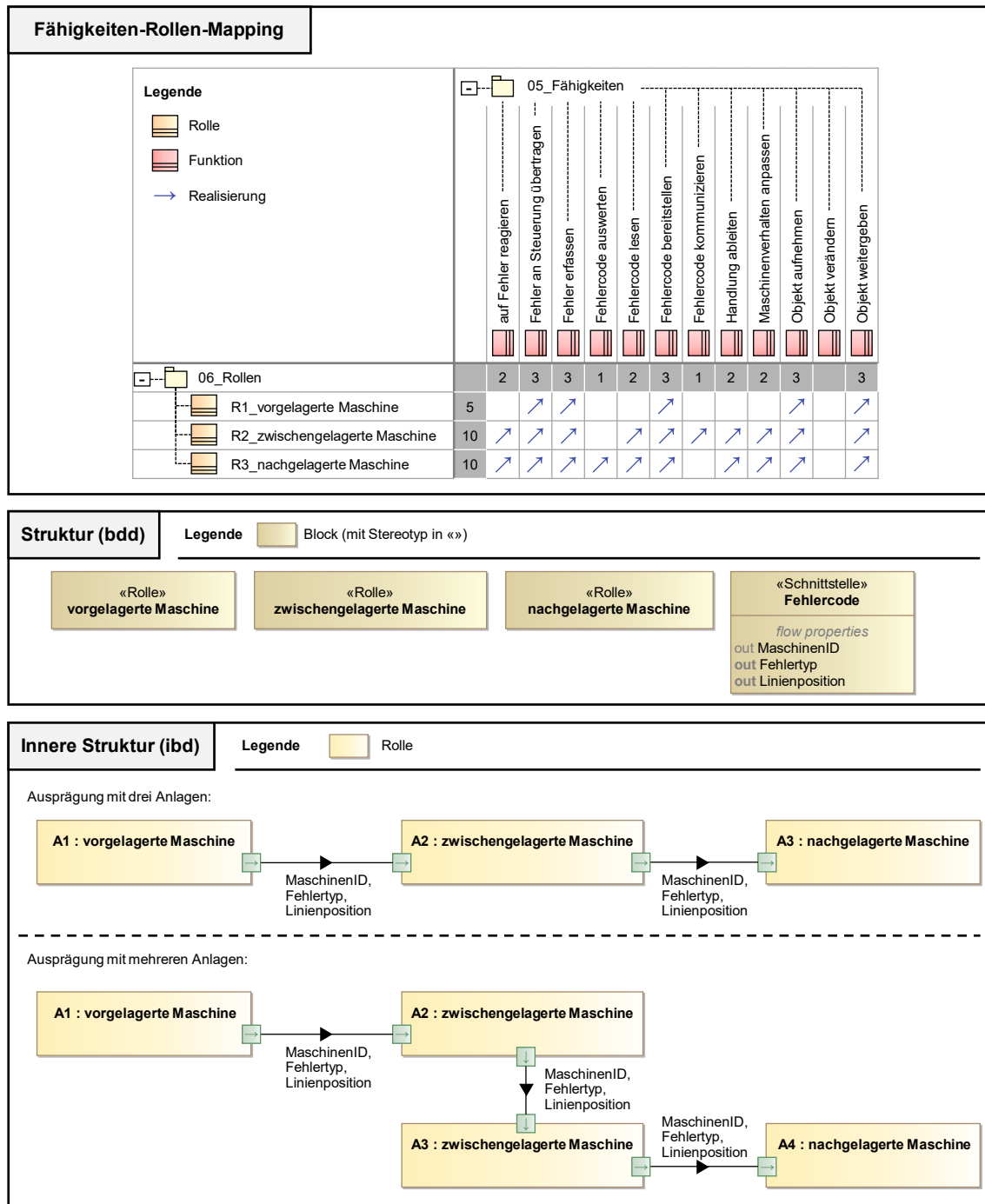


Bild 5-16: Interaktionsspezifikation mithilfe von Rollen und Fähigkeiten

Die definierten Rollen mit den jeweiligen Fähigkeiten sind die Basis für ein erfolgreiches Zusammenwirken der Systeme innerhalb einer Verpackungslinie. Jede Maschine nimmt dabei eine Rolle ein und muss die Fähigkeiten und Informationen, die für die Rolle definiert sind, bereitstellen. Wird bspw. ein Handlingssystem mit Automatisierungslösung als zwischengelagerte Maschine eingesetzt, so muss das System u. a. die Fähigkeit besitzen, einen *Fehlercode zu lesen* und *weiter zu kommunizieren*. Die Verknüpfung der grundsätz-

lichen Interaktionsprinzipien im SoS und die konkrete Ausprägung einer Verpackungslinie durch unterschiedliche Maschinen erfüllt dabei die Anforderungen an eine flexible und herstellerunabhängige Zusammenstellung von Maschinen in einer Verpackungslinie.

Der informationstechnische Zusammenhang zwischen der unternehmensspezifischen Systemsicht und der zuvor beschriebenen SoS-Sicht wird in Bild 5-17 dargestellt. Im konkreten Fall der Automatisierungslösung repräsentiert diese ein System eines Unternehmens, welches eine bestimmte Rolle übernimmt und somit spezifische Fähigkeiten bereitstellen muss. Dabei interagiert die Automatisierungslösung im SoS und trägt zur Erfüllung des SoS-Ziels bei.

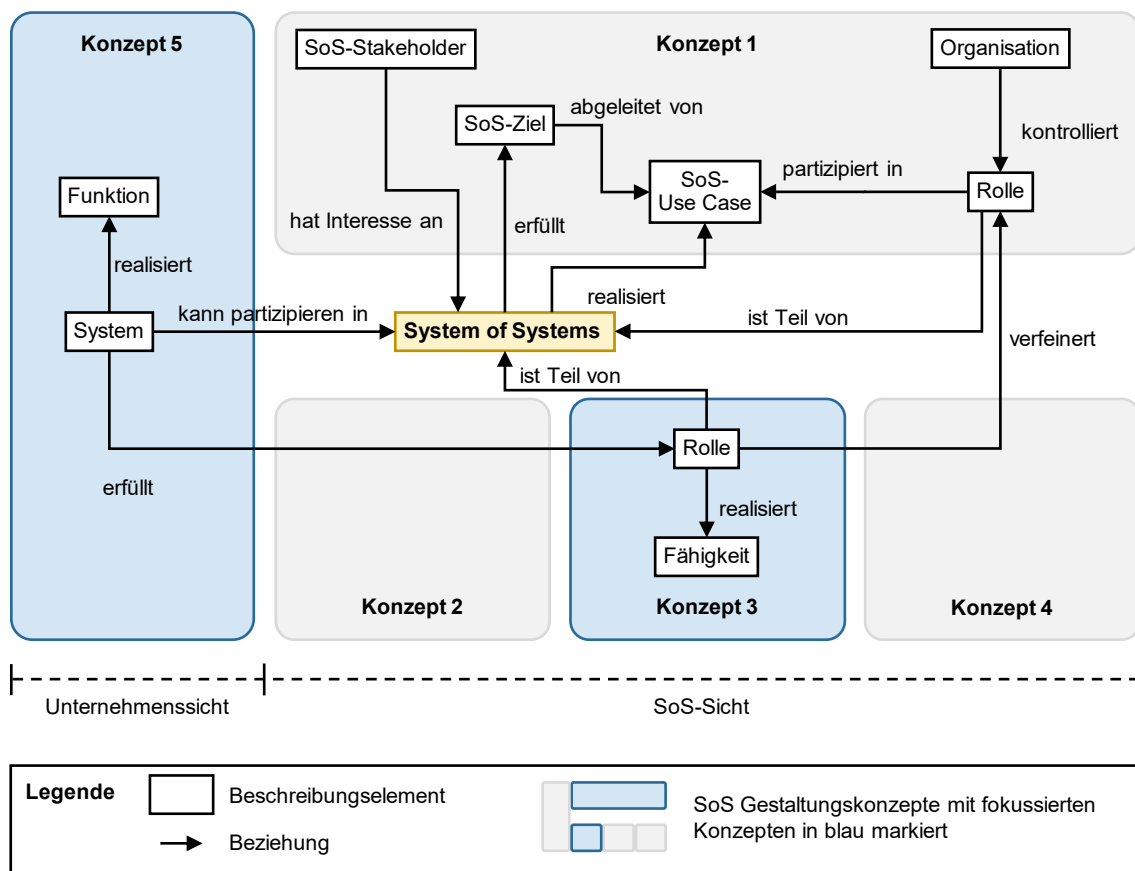


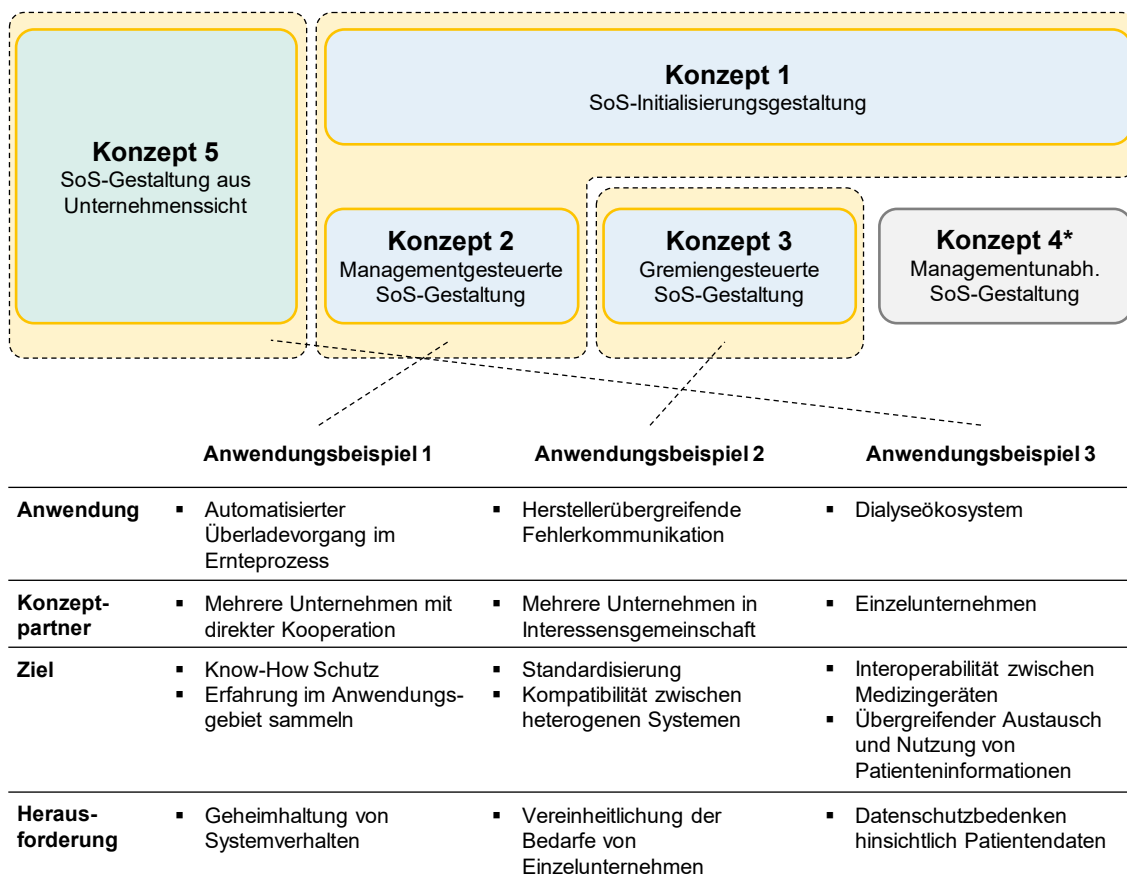
Bild 5-17: Zusammenhang zwischen System- und SoS-Sicht für das Anwendungsbeispiel „herstellerübergreifende Fehlerkommunikation“

5.3 Bewertung der Anforderungserfüllung des Rahmenwerks

Im Folgenden wird das Rahmenwerk anhand der Kriterien Anwendbarkeit und Effektivität (Kapitel 5.3.1) sowie der Erfüllung der in Kapitel 2.6 gestellten Anforderungen (Kapitel 5.3.2) bewertet.

5.3.1 Bewertung der Anwendbarkeit und Effektivität auf Basis der Anwendung in der Praxis

Neben den zwei beschriebenen Anwendungsbeispielen in Kapitel 5.1 und 5.2 stützt sich die Bewertung des Rahmenwerks auf den Erkenntnissen und dem Feedback einer weiteren Anwendung. Die dritte Anwendung wurde gemeinsam mit einem Medizingerätehersteller durchgeführt und hatte das Ziel, ein Dialyseökosystem zu gestalten. Dabei wurde zunächst ein initiales Konzept aus Unternehmenssicht entwickelt und entsprechend des Gestaltungskonzepts 5 ausgearbeitet. Die Untersuchungen auf Basis der Leitlinien und Hilfsmittel führten zu der Erkenntnis, dass weitere Fähigkeiten durch Systeme bereitgestellt werden müssen, um das Ziel einer bestmöglichen Patientenversorgung mithilfe eines zu entwickelnden SoS zu erreichen. Bspw. werden Analysefähigkeiten zur Bestimmung von physiologischen Daten (u. a. Blutdruck und Gewicht) benötigt, die nicht im Zugriffsbereich des Anwendungspartners liegen. Auf Basis der ersten Erkenntnisse sollte im nächsten Schritt entsprechend der Vorgaben im Gestaltungskonzept 5 die Suche nach passenden Unternehmenspartnern beginnen und durch Kooperationen mit weiteren Unternehmen die Gestaltung anhand des Gestaltungskonzepts 1 weitergeführt werden. Eine Zusammenfassung der Anwendungsbeispiele und deren Zuordnung zu den angewandten Gestaltungskonzepten wird in Bild 5-18 visualisiert.



*Kein direkter Anwender (vgl. Kapitel 4.1)

Bild 5-18: Übersicht der Anwendungsbeispiele im Rahmen der Evaluation

Die drei analysierten Beispiele unterstreichen die Vielfalt von SoS und zeigen gleichzeitig die **Anpassungsfähigkeit** des Rahmenwerks an unterschiedliche Rahmenbedingungen. Hierbei wurden unterschiedliche Maßnahmen ergriffen, um die Anwendung zielgerichtet für unterschiedliche Unternehmen zu ermöglichen ohne die Komplexität des Rahmenwerks zu groß werden zu lassen. Ein Beispiel ist der SoS-Quick-Check. Nachdem in einer ersten Iterationsschleife acht Leitfragen sukzessive beantwortet werden mussten, konnte dieser durch einen angepassten Fragebaum auf sieben Fragen reduziert werden, wobei je nach Antwort eine dynamische Reduzierung auf bis zu vier Fragen erfolgt. Ein weiteres Beispiel sind die kompakten Gestaltungskonzeptsteckbriefe, die einer klaren Strukturierung in managementbezogene und technische Hinweise folgen. Dabei wurden allgemeingültige und unspezifische Hinweise entfernt, um die wesentlichen Herausforderungen zu adressieren. Diese Maßnahmen führten schließlich zu einer größeren Akzeptanz und Zufriedenheit bei der Nutzung des Rahmenwerks.

Die Vielfältigkeit von SoS spiegelt sich auch in den umfassenden Themen wider, die mit SoS einhergehen und Teil der Diskussionen mit den Anwendungspartnern waren. Neben der Betrachtung von strategischen Planungsthemen spielt auch die Etablierung von Serviceleistungen durch Produkt-Service-Systeme eine wichtige Rolle. Ein Beispiel ist die Flottenfahrt. Hierbei muss ein vorausfahrender LKW einen Ausgleich, bspw. in Form einer Bezahlung der nachfolgenden LKW, bekommen. Die Mitfahrt eines LKW in einer Flotte könnte z. B. durch ein Servicemodell angeboten werden. Dies würde den Gewinn durch die Energieeinsparung beim Fahren im Windschatten über das SoS durch eine Ausgleichszahlung verteilen. Die **Fokussierung auf technische SoS** im Rahmen des Lösungsansatzes stellt sich aus zwei Gründen als zielführend heraus. Zum einen bringt der Bereich der Servitisierung im SoS eine zusätzliche Komplexität mit sich, die es separat zu betrachten gilt. Zum anderen steht nicht immer die Monetarisierung an erster Stelle, wie z. B. der automatisierte Überladevorgang zeigt.

Insgesamt lässt das Feedback der Anwendungspartner auf eine Verbesserung der übergreifenden Systementwicklung schließen. Die in Kapitel 2 und 3 beschriebene Ausgangsbasis zeigt Forschungs- und Anwendungslücken, die auch durch die Teilnehmenden bestätigt wurden. Etablierte Ansätze fokussieren sich auf einzelne Systeme (klassische MBSE-Ansätze) oder gehen lediglich auf einen begrenzten Umfang der bei der Gestaltung von SoS relevanten Themen ein. Den Anwendungspartnern waren lediglich isolierte Ansätze zur Entwicklung von SoS bekannt, die nur unzureichend den Bedarfen der jeweiligen Anwendungsbeispiele genügen. Dem gegenüber steht die stetig steigende Vernetzung der Systeme der Anwendungspartner und die damit einhergehenden wachsenden Fragestellungen in Bezug auf die Gestaltung von SoS. Das im Rahmen dieser Arbeit vorgestellte Rahmenwerk unterstützt sowohl den Zugang als auch die weitere Begleitung von SoS über verschiedene Entwicklungsstufen.

Insbesondere eine umfassende Berücksichtigung der unterschiedlichen Typen von SoS sowie dazu passende konkrete Vorgehen und Hilfsmittel zur Gestaltung fehlten. Das Rahmenwerk wurde durch die Teilnehmenden als praktikabler Ansatz beschrieben und bietet

durch die bedarfsgerechten Hilfsmittel einen wesentlichen Mehrwert für die Entwicklung von SoS und den darin integrierten eigenständigen Systemen. Um die Anwendbarkeit weiter zu steigern, wurden Verbesserungen vorgeschlagen. Zum einen wurde die Erweiterung durch ergänzende Methoden genannt, um beispielsweise die Identifikation von passenden Unternehmenskooperationen anhand der Geschäftsmodelle zu unterstützen. Zum anderen ist auch die Pflege, Weiterentwicklung und Zugänglichkeit der Muster und Designprinzipien zur weiteren Steigerung des Nutzens benannt worden. Beide Vorschläge werden durch die vorgegebene Strukturierung sowohl der Methodensteckbriefe als auch der Wissenstemplates prinzipiell unterstützt, müssen jedoch durch die systematische Anwendung des Ansatzes und kontinuierliche Aufbereitung der neuen Erkenntnisse weiter vorangetrieben werden.

Zusammenfassend zeigen die Erfahrungen und Erkenntnisse im Rahmen der Evaluierung die **Anwendbarkeit** und **Nützlichkeit** des Rahmenwerks für unterschiedliche Arten von SoS sowie unterschiedliche Gestaltungsziele und Unternehmenskonstellationen.

5.3.2 Bewertung der Anforderungserfüllung des Rahmenwerks

Das Rahmenwerk soll die in Kapitel 2 herausgearbeiteten Herausforderungen im Umgang mit SoS adressieren und Unternehmen gezielt bei der Gestaltung von SoS unterstützen. Im Folgenden werden die an das Rahmenwerk gestellten Anforderungen im Einzelnen bewertet.

Handlungsfeld 1: Definition und Strukturierung von SoS

A1) Definition von SoS im Kontext von technischen Systemen: Für ein einheitliches Verständnis von SoS wird eine konsistente *Definition von SoS* mit Fokus auf technische Systeme gegeben (Kapitel 4.2.1). Die Ausführung konsolidiert die Grundlagenliteratur und stützt sich auf charakteristischen Merkmalen, die SoS spezifizieren und von dem allgemeinen Systembegriff differenzieren.

A2) Berücksichtigung der charakteristischen Merkmale von SoS: Die in der Definition konsolidierten *charakteristischen Merkmale* sind die Grundlage des Lösungsansatzes und werden an unterschiedlichen Stellen aufgegriffen und berücksichtigt. U. a. werden sie implizit in den Leitfragen zur Auswahl eines Gestaltungskonzepts (vgl. Kapitel 4.3.3) sowie in den entsprechenden Gestaltungskonzepten selbst berücksichtigt. Zusätzlich dienen die charakteristischen Merkmale der Strukturierung und Identifizierung von *Wissenselementen* als Hilfsmittel für die Gestaltung (vgl. Kapitel 4.4.1 und 4.4.2).

A3) Klassifizierung von SoS: Eine ausführliche Analyse und Bewertung von Anwendungs- und Literaturbeispielen anhand der charakteristischen Merkmale sind die Grundlage der *Klassifizierung von SoS* (vgl. Kapitel 4.2.2). Diese erfüllt die Anforderungen an eine prägnante und gleichzeitig sparsame Klassifizierung, indem drei spezifische Klassen von SoS voneinander abgegrenzt sind. Zusätzlich ist die Klassifizierung Grundlage des

Gestaltungsrahmens (vgl. Kapitel 4.3.1), indem die Klassifizierung um eine projektspezifische Dimension ergänzt und den jeweiligen Klassen bedarfsgerechte Methoden und Hilfsmittel bereitstellt.

Handlungsfeld 2: Unternehmensübergreifende Gestaltung von SoS

A4) Berücksichtigung von SoS- und Systemperspektiven: Mit Hilfe des *Gestaltungsrahmens* (vgl. Kapitel 4.3.1) werden unterschiedliche Rahmenbedingungen, Ziele und Unternehmensbeziehungen berücksichtigt und passende Leitlinien und Methoden bereitgestellt. Dabei fokussieren Gestaltungskonzepte 1 bis 4 die SoS-Perspektiven, während Gestaltungskonzept 5 auf die Unternehmenssicht eingeht und explizit die Systemperspektive betrachtet. Zusätzlich wird durch die beschriebenen Übergänge im Rahmen der *Erweiterten Nutzung des Gestaltungsrahmens* (vgl. Kapitel 4.3.4) der Wechsel zwischen Perspektiven beschrieben.

A5) Gestaltungsmethoden strukturieren und bereitstellen: Die Vielfalt an SoS resultiert in zahlreichen Ansätzen, die spezifische Bedarfe bei der SoS-Gestaltung adressieren. Durch die im Gestaltungsrahmen abgegrenzten Gestaltungskonzepte konnten die im Stand der Forschung identifizierten Methoden bestimmten Bedarfen aus den Gestaltungskonzepten zugeordnet werden (vgl. Kapitel 4.3.2). Zusätzlich wurden spezifische Methoden für die Gestaltungskonzepte entwickelt und in Form von Methoden-Steckbriefen bereitgestellt (vgl. Kapitel 4.3.2.1 bis 4.3.2.3).

A6) Anwendbarkeit sicherstellen: Um die Anwendbarkeit sicherzustellen, wurde ein *Vorgehen* (vgl. Kapitel 4.3.3) entwickelt, welches anhand von drei Schritten eine bedarfsgerechte Anwendung unterschiedlicher Hilfsmittel sicherstellt. Ein vorstrukturierter Grobkonzept-Steckbrief sowie ein *Leitfragenkatalog* inkl. ergänzender Informationen und Beispiele führen anhand einer *Auswahllogik* zu einem passenden Gestaltungskonzept. Über Hinweise und referenzierte Methoden wird anschließend die Anwendung des jeweiligen *Gestaltungskonzepts* unterstützt. Für die Gestaltungskonzepte 1 bis 5 exklusive Gestaltungskonzept 4 wurde die Anwendbarkeit entsprechend nachgewiesen (vgl. Kapitel 5.3.1).

Handlungsfeld 3: Wissensmanagement zur Unterstützung der Gestaltung von SoS

A7) Wissenstransfer ermöglichen: Designprinzipien und Lösungsmuster sind etablierte Ansätze zur Strukturierung und Kommunikation von Wissen. Die *Designprinzipien-* und *Lösungsmustertemplates* bieten eine Möglichkeit, Wissen zu beschreiben und für unternehmensübergreifende Kooperationen zugänglich zu machen (vgl. Kapitel 4.4.1 und 4.4.2).

A8) Bereitstellung von SoS-Lösungswissen: Designprinzipien und Lösungsmuster beschreiben Lösungen, die für unterschiedliche Probleme angewendet werden können. Insgesamt werden 30 *generische* und 57 *spezifische Designprinzipien* (vgl. Kapitel 4.4.1) sowie sieben *Lösungsmuster* (vgl. Kapitel 4.4.2) bereitgestellt, die bei unterschiedlichen

Problemstellungen im Rahmen der SoS-Gestaltung unterstützen. Durch die Charakterisierung anhand der SoS-Merkmale können diese für konkrete Herausforderungen der SoS-Gestaltung eingesetzt werden.

6 Zusammenfassung und Ausblick

Die durch die Digitalisierung hervorgerufene zunehmende Vernetzung von Systemen in Systemverbünden, ermöglicht neue Funktionen und hebt Potentiale, die einzelne Systeme nicht erreichen können. Solche Systemverbünde, auch System of Systems (SoS) genannt, bestehen aus unterschiedlichen eigenständigen Systemen und bilden sich dynamisch. Unternehmen sind gefordert über Unternehmensgrenzen hinweg neue Entwicklungsansätze zu verfolgen und unternehmensübergreifende Kooperationen einzugehen, um Systemverbünde gestalten zu können.

Kapitel 1 führt in die Problematik ein und begründet die Zielstellung der vorliegenden Arbeit, die sich mit der Gestaltung von System of Systems auseinandersetzt. Das übergeordnete Ziel besteht darin, Unternehmen dabei zu unterstützen, die Herausforderungen von SoS gezielt anzugehen. Vor diesem Hintergrund ist das Forschungsvorgehen aus dem Bereich der anwendungs- und gestaltungsorientierten Forschung gewählt. Im Rahmen der vorliegenden Arbeit wurde die Design-Science-Research-Methodology nach PEFFERS et al. angewendet und das Vorgehen entsprechend strukturiert [PTG+06].

Darauf aufbauend erfolgt in **Kapitel 2** eine detaillierte Problemanalyse. Betrachtungsgegenstand sind technische Systeme und die an deren Gestaltung beteiligten Unternehmen. Zunächst wird die Veränderung der Systeme von mechatronischen hin zu intelligenten vernetzten Systemen betrachtet. Hintergrund ist der steigende Anteil an Informations- und Kommunikationstechnologie, der sowohl die Funktionalität als auch die Vernetzungsfähigkeiten der Systeme erhöht. Die Interaktionsfähigkeit von Systemen ermöglicht neue Anwendungen und Synergien. Der Nutzen und die Effizienz innerhalb der Systemverbünde übersteigt die Ergebnisse, die durch die reine Optimierung eines einzelnen Systems erzielt werden können. Für Unternehmen ist die Erweiterung der Betrachtungsgrenzen in Form von SoS unumgänglich, um wettbewerbsfähig zu bleiben. Dies geht einher mit zahlreichen Herausforderungen, die aus Sicht von Unternehmen zu bewältigen sind. Im Kontext des Engineerings müssen Lösungen, wie die Interoperabilität zwischen Systemen, unternehmensübergreifend gelöst werden. Im Gegensatz zu reinen Systementwicklungsprojekten, sind mehrere Unternehmen bei der Ausarbeitung von Lösungen involviert und es gibt keine alleinige Entscheidungsbefugnis. Gleichzeitig sorgt die gestiegene Komplexität dafür, dass ein erweitertes Wissen benötigt wird, um Lösungen zu finden. Die Dynamik von SoS und die immer kürzeren Innovationszyklen lassen die Zeitspanne zur Entwicklung geeigneter Lösungen immer kürzer werden. Vor diesem Hintergrund werden Ansätze des Wissensmanagement im Hinblick auf die Herausforderungen von SoS analysiert und Handlungsbedarfe abgeleitet. Aus den beschriebenen Handlungsbedarfen werden insgesamt acht Anforderungen an ein zu entwickelndes Rahmenwerk definiert.

Auf Grundlage der Problemstellung werden im Stand der Forschung in **Kapitel 3** etablierte Ansätze im Kontext der Arbeit betrachtet und hinsichtlich der gestellten Anforderungen analysiert. Zunächst werden unterschiedliche Beschreibungsansätze für SoS analysiert und in die Bereiche Definition und Klassifizierung unterteilt. Anschließend folgt die Betrachtung von Ansätzen zur Gestaltung von SoS, die sich sowohl spezifisch mit SoS auseinandersetzen als auch aus angrenzenden Forschungsbereichen kommen und einzelne Aspekte von SoS aufgreifen. Darüber hinaus erfolgt eine Betrachtung von wissensbasierten Ansätzen, die eine Unterstützung von Entwicklungstätigkeiten zum Ziel haben. Eine abschließende Bewertung der Ansätze anhand der Anforderungen begründet einen ausstehenden Handlungsbedarf.

Das entwickelte Rahmenwerk stellt den Kern der Arbeit dar und wird in **Kapitel 4** vorgestellt. Das Rahmenwerk zur musterbasierten Gestaltung von System of Systems besteht aus drei wesentlichen Elementen:

- 1) Grundlage ist eine einheitliche **Definition und Klassifizierung von System of Systems**, die sich auf technische Systeme fokussiert. Diese orientiert sich an etablierten Literaturen und konsolidiert das Verständnis in einer einheitlichen Beschreibung anhand von acht charakteristischen Merkmalen. Darüber hinaus werden auf Basis einer Analyse von Anwendungs- und Literaturbeispielen drei unterschiedliche Klassen von SoS unterschieden. Diese dienen der zweckmäßigen Unterscheidung der vielfältigen Ausprägungen von SoS, um zielgerichtete Maßnahmen und Handlungsrichtlinien zu strukturieren.
- 2) Der **Gestaltungsrahmen für das SoS-Engineering** greift die Klassifizierung von SoS auf und erweitert diese um die Dimension der Entwicklungsphase. Hintergrund ist die vor allem für SoS-Entwicklungsmaßnahmen entscheidende Differenzierung zwischen der Initialisierungs- und Weiterentwicklungsphase. Insgesamt werden fünf Gestaltungskonzepte unterschieden, die je nach Rahmenbedingungen und SoS-Typ passende Leitlinien bereitstellen. Diese bieten Unternehmen eine Orientierung für die Gestaltung von SoS und gehen dabei auf wichtige Zusammenhänge zwischen den Organisationsstrukturen und den technischen Konzepten ein. Zusätzlich verweisen die Gestaltungskonzepte auf relevante Methoden, die bei der Umsetzung der Leitlinien Anwendung finden und in Form von Methodensteckbriefen bereitgestellt werden. Als Einstieg in den Gestaltungsrahmen und Identifizierung eines passenden Gestaltungskonzepts dient ein Vorgehen mit unterstützenden Hilfsmitteln.
- 3) Die **Hilfsmittel für die wissensbasierte SoS-Gestaltung** ergänzen den Gestaltungsrahmen und stellen für die unterschiedlichen Gestaltungsaktivitäten und Herausforderungen im Kontext von SoS Wissen bereit. Die Basis sind etablierte Prinzipien und Lösungsmuster, die in einer einheitlichen Struktur zur Verfügung gestellt werden und anhand der charakteristischen Merkmale von SoS bedarfsge-

recht identifiziert und angewendet werden können. Drei grundsätzliche Handlungsoptionen sind im Kontext der wissensbasierten Gestaltung relevant und mithilfe der vorgestellten Strukturierung möglich: die Nutzung von Wissen zur Inspiration, die Identifikation von Wissen für bestimmte Herausforderungen und die Dokumentation von neuem Wissen für zukünftige Problemstellungen.

Kapitel 5 beschreibt die Anwendung und Evaluation des vorgestellten Rahmenwerks. Das Rahmenwerk wurde anhand von insgesamt drei unterschiedlichen Anwendungsbeispielen mit Unternehmensvertretern evaluiert. Dabei wurde das Vorgehen jeweils durchlaufen und vier der fünf Gestaltungskonzepte des Rahmenwerks erprobt. Lediglich Gestaltungskonzept 4, welches sich an ein stark unabhängiges SoS richtet, in dem es keinen direkten Adressaten gibt, wurde aufgrund der Charakteristika nicht evaluiert. Die Anwendung zeigt, dass das Rahmenwerk für unterschiedliche Rahmenbedingungen anwendbar ist. Das Feedback der Unternehmen ist grundsätzlich positiv, da das Rahmenwerk auf unterschiedliche Herausforderungen und Rahmenbedingungen von SoS eingeht und entsprechende Hilfsmittel zur Verfügung stellt. Gleichzeitig werden Verbesserungen insbesondere hinsichtlich der Verstetigung der Ansätze in Unternehmensprozesse gesehen. Die Gestaltung von SoS betrifft viele Prozesse in Unternehmen, sodass eine weitere Vertiefung der Leitlinien und Erweiterung der Methoden bspw. für strategische Unternehmensprozesse oder auch Validierungs- und Verifikationstätigkeiten notwendig werden. Insgesamt erfüllt das Rahmenwerk die gestellten Anforderungen und wurde hinreichend hinsichtlich der Anwendbarkeit und Nutzbarkeit evaluiert.

Dennoch bestehen einige **Limitationen**, die im Folgenden erläutert werden. Zum einen ist die **Anzahl an Anwendungen begrenzt** und lässt keinen vollständigen Beweis der Nützlichkeit zu. Hierzu benötigt es neben weiteren Evaluationen zusätzlich einen Vergleich der Gestaltung von SoS mit Unterstützung durch den Lösungsansatz gegenüber SoS-Projekten ohne Unterstützung. Dies ist nicht zuletzt aufgrund von Zeitrestriktionen im Rahmen der Arbeit nicht möglich gewesen, da Entwicklungsprojekte im Vergleich zu Forschungsarbeiten häufig längere Zeit in Anspruch nehmen. Insbesondere in SoS-Gestaltungsprojekten erfordern Entscheidungen Zeit, wenn es z.B. um Unternehmenskooperationen geht. Dies hängt neben den unterschiedlichen Interessen und Rahmenbedingungen auch mit rechtlichen und strategischen Entscheidungen der beteiligten Unternehmen zusammen und führt dazu, dass eine Evaluation nur durch unterschiedliche Maßnahmen durchzuführen ist. Zum einen stützen sich die **Entscheidungen der Unternehmensvertreter in den Evaluationsanwendungen zum Teil auf Vorarbeiten**, die u. a. innerhalb von Gremien getroffen wurden, sowie auf Erkenntnissen aus bereits getroffenen Entscheidungen. Zum anderen ist das Thema Systems Engineering noch nicht vollständig in Unternehmen angekommen und für sich bereits nur mit entsprechendem Zeitaufwand zu vermitteln. Daher wurden **Anwendungspartner mit Vorkenntnissen und Erfahrungen** in diesem Themenfeld ausgewählt. Diese sind in gewisser Weise in ihrer Bewertung beeinflusst, da sie diesen Ansätzen tendenziell eher positiv gegenüberstehen.

Abschließend sind weitere Limitationen mit Fokus auf die bereitgestellten Wissensselemente aufzuführen. Diese wurden im Rahmen von wissenschaftlichen Publikationen aufbereitet und durch wissenschaftliche Experten begutachtet. Neben der vereinzelt Anwendung der Wissensselemente und dem grundsätzlich nachgewiesenen Mehrwert, erfolgte **keine vollständige Evaluation der Wissensselemente** in praktischen Anwendungen. Dies ist nicht zuletzt auf die Erweiterungsmöglichkeit und kontinuierliche Ergänzung von Wissensselementen im Verlauf der Arbeit zurückzuführen.

Aufbauend auf den Limitationen und Erkenntnissen im Rahmen der Forschungsarbeit bestehen weitere **Forschungsbedarfe**. **Kurzfristig** sollte zunächst eine umfassende Evaluation des Rahmenwerks erfolgen. Neben der Erhöhung der Anzahl sollten auch weitere Faktoren variiert werden. Dazu gehört neben unterschiedlichen Typen von SoS auch die Anzahl an beteiligten Vertretern und deren Beziehung untereinander. Sowohl die Betrachtung von Konkurrenzbeziehungen als auch Aspekte wie die Marktmacht können unterschiedliche Auswirkungen auf die Nutzung des Rahmenwerks haben. Eine Evaluation muss daher systematisch geplant und mit dem notwendigen Vorlauf umgesetzt werden.

Mittel- und langfristig gilt es, vier Forschungsbedarfe zu adressieren. (1) Ein großer Bedarf besteht in der Unterstützung der Entwicklung von SoS durch Modelle [RM21, S. 403]. Das Aufbrechen von Betrachtungsgrenzen durch dynamische Interaktionen in SoS erfordert übergreifende Modellierungsansätze, um einen unternehmensübergreifenden Austausch und unternehmerische Interoperabilität zu ermöglichen [Tek22, S. 6]. Die Ansätze im Rahmen der Gestaltungskonzepte bieten eine gute Basis, sind jedoch nicht auf eine formalisierte Herangehensweise ausgelegt. Um eine formale Modellierung zu ermöglichen wird u. a. eine gemeinsame Sprache benötigt, die den Anforderungen von SoS gerecht wird [MOD16, S. 637f.], [NLF+15, S. 24]. (2) Darüber hinaus erfordern die charakteristischen Merkmale von SoS eine gleichzeitige Betrachtung von strategischen Managementaktivitäten sowie Konzeptentscheidungen in der Architekturentwicklung, wie sie in den Leitlinien der SoS-Gestaltungskonzepte aufgeführt werden. Das Vier-Zyklen Modell nach GAUSEMEIER et al. beschreibt in diesem Zusammenhang den iterativen Charakter zwischen der strategischen Produktplanung und der Produktentwicklung [GDE+19, S. 89f.]. Es gilt, die Ansätze der SoS-Gestaltung auf die Tätigkeiten der strategischen Planung zu übertragen und in die Entscheidungsfindung u. a. für Unternehmenskooperationen und neue Geschäftsmodelle mit einzubeziehen. (3) Im Rahmen der Arbeit wurden insbesondere technische SoS betrachtet. Wie bspw. TEKINERDOGAN beschreibt, gibt es weitere Dimensionen von SoS [Tek19, S. 280]. Dazu gehören sowohl Cyber-physische Systeme mit dem Fokus auf Service und Daten als auch soziotechnische SoS. Insbesondere im Kontext dynamischer und emergenter Interaktionen von SoS ist der Mensch ein wesentlicher Einflussfaktor und sollte integrativ in die Analysen mit einbezogen werden. Ansätze, wie die rollenbasierte Modellierung von SoS (vgl. Kapitel 4.3.2.2), bieten vielversprechende Möglichkeiten, die weiter erforscht werden sollten. (4) Abschließend gilt es die Möglichkeiten durch generative KI im Hinblick auf die unzähl-

gen Unwägbarkeiten in der SoS-Entwicklung zu erforschen. So können bspw. KI-Agenten mit bestimmten Fähigkeiten trainiert werden und miteinander in Austausch gebracht werden. Die Interaktionen können hinsichtlich Emergenzen analysiert und frühzeitig im Rahmen der Gestaltung mitberücksichtigt werden.

Abkürzungsverzeichnis

bspw.	beispielsweise
CPS	Cyber-physische Systeme
DSRM	Design Science Research Methodology
ES	eigenständiges System
et al.	et alia (lat. „und andere“)
ggf.	gegebenenfalls
IoT	Internet of Things
iPeM	integriertes Produktentstehungsmodell
ISS	Internationale Raumfahrtstation
ITS	Intelligente Technische Systeme
KES	kollaborative eingebettete Systeme
KSG	kollaborative Systemgruppen
MBSE	Model-based Systems Engineering
MoSyS	Mensorientierte Gestaltung komplexer System of Systems (Forschungsprojekt)
OPC UA	Open Platform Communications Unified Architecture
SE	Systems Engineering
SLR	systematische Literaturanalyse (engl. „systematic literature review“)
SoI	System of Interest
SoS	System of Systems
SoSE	System of Systems Engineering
u. a.	unter anderem
USPS	United States Postal Service
vgl.	vergleich(e)
z. B.	zum Beispiel

Literaturverzeichnis

- [Abb06] ABBOTT, R.: Open at the Top; Open at the Bottom; and Continually (but Slowly) Evolving. In: IEEE (Hrsg.): Proceedings 2006 IEEE/SMC International Conference on System of Systems Engineering. 24.-26. April 2006, Los Angeles, California, USA, 2006, S. 41–46
- [ABD+16] ANDERL, R.; BAUER, K.; DIEGNER, B.; DIEMER, J.; FAY, A.; FRITZ, J.; GOERICKE, D.; GROTEPASS, J.; HILGER, C.; JASPERNEITE, J.; KALHOFF, J.; KUBACH, U.; LÖWEN, U.; MENGES, G.; JAN STEFAN, M.; MÜNCH, W.; PREIB, H.; REUS, O.; SCHMOLZE-KRAHN, R.; STEFFENS, E.-J.; STIEDL, T.; HOMPEL, M. TEN; ZEIDLER, C.: Plattform Industrie 4.0 - Aspekte der Forschungsroadmap in den Anwendungsszenarien – Ergebnisrapport, 2016
- [aca17-ol] ACATECH – DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN: Wegweiser Smart Service Welt – Smart Services im digitalen Wertschöpfungsnetz. Unter: https://www.acatech.de/wp-content/uploads/2018/03/acatech2017_SSW_Wegweiser_de_bf.pdf, 1. Oktober 2024
- [Ada11] ADAMS, K. M.: Systems principles: foundation for the SoSE methodology. In: Inderscience Publishers (Hrsg.): International Journal of System of Systems Engineering. 2(2-3), 2011, S. 120–155
- [ADK13] ACHESON, P.; DAGLI, C.; KILICAY-ERGIN, N.: Model Based Systems Engineering for System of Systems Using Agent-based Modeling. In: Paredis, C. J. J.; Bishop, C.; Bodner, D. (Hrsg.): Procedia Computer Science. 2013 Conference on Systems Engineering Research, 2013, S. 11–19
- [AEF23-ol] AEF - AGRICULTURAL INDUSTRY ELECTRONICS FOUNDATION E.V.: About the AEF. Unter: <https://www.aef-online.org/about-us/about-the-aef.html#/About>, 23. November 2023
- [AEL+23] AGARWAL, P.; ESTANGUET, R.; LEBLANC, F.; ERNADOTE, D.: Architecting System of Systems for Resilience using MBSE: 2023 18th Annual System of Systems Engineering Conference (SoSe). 14. - 16. Juni 2023, Lille, France, IEEE, 2023, S. 1–7
- [AES+12] ANDERL, R.; EIGNER, M.; SENDLER, U.; STARK, R. (Hrsg.): Smart Engineering – Interdisziplinäre Produktentstehung. acatech DISKUSSION, Springer Verlag, Berlin, Heidelberg, 2012
- [AFE19] AXELSSON, J.; FRÖBERG, J.; ERIKSSON, P.: Architecting systems-of-systems and their constituents: A case study applying Industry 4.0 in the construction domain: Systems Engineering. 22 (6), 2019, S. 455–470
- [AGW+22] ANACKER, H.; GUENTHER, M.; WYRWICH, F.; DUMITRESCU, R.: Pattern based engineering of System of Systems - a systematic literature review: 17th Annual System of Systems Engineering Conference (SOSE). 07. - 11. Juni 2022, Rochester, NY, USA, IEEE, 2022, S. 178–183
- [AIS+77] ALEXANDER, C.; ICHIKAWA, S.; SILVERSTEIN, M.; JACOBSON, M.; FIKSDAHL-KING, I.; ANGEL, S.: A pattern language – Towns, Buildings, Construction. 41st Edition, Oxford University Press, Vol. 2 New York, NY, 1977
- [AIS+95] ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M.; JACOBSON, M.; KING, I. F.; ANGEL, S.: Eine Muster-Sprache – Städte, Gebäude, Konstruktion. Löcker Verlag, Wien, 1995
- [AK11] ADAMS, K. M.; KEATING, C. B.: Overview of the systems of systems engineering methodology: International Journal of System of Systems Engineering. 2 (2-3), 2011, S. 112–119
- [AK18] AXELSSON, J.; KOBETSKI, A.: Towards a risk analysis method for systems-of-systems based on systems thinking: 2018 Annual IEEE International Systems Conference (Sys-Con). 23. - 26. April 2018, Vancouver, BC, 2018, S. 1–8

- [AMY+18] ALBERS, A.; MANDEL, C.; YAN, S.; BEHRENDT, M.: System of systems approach for the description and characterization of validation environments: DS92: Proceedings of the DESIGN 2018 15th International Design Conference, 2018, S. 2799–2810
- [Ana15] ANACKER, H.: Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme. Dissertation, Heinz Nixdorf Institut, Universität Paderborn, 2015
- [ARB+16] ALBERS, A.; REIB, N.; BURSAC, N.; RICHTER, T.: The integrated Product engineering Model (iPeM) in context of the product generation engineering: Proceedings 26th CIRP Design Conference. Stockholm, Schweden, 2016, S. 100–105
- [Axe19a] AXELSSON, J.: Business Models and Roles for Mediating Services in a Truck Platooning System-of-Systems: 2019 14th Annual Conference System of Systems Engineering (SoSE). 19. - 22. Mai 2019, Anchorage, AK, USA, IEEE, 2019, S. 113–118
- [Axe19b] AXELSSON, J.: A Refined Terminology on System-of-Systems Substructure and Constituent System States: 14th Annual Conference System of Systems Engineering (SoSE). IEEE, 2019, pp. 31–36
- [Axe22] AXELSSON, J.: What Systems Engineers Should Know About Emergence: INCOSE International Symposium. 32 (1), 2022, S. 1070–1084
- [Aza08] AZANI, C. H.: System of Systems Architecting via Natural Development Principles: International Conference on System of Systems Engineering (SoSE). 02. - 04. Juni 2008, Monterey, California, USA, IEEE, 2008, pp. 1–6
- [Aza09] AZANI, C.: A multi-criteria decision model for migrating legacy system architectures into open system and system-of-systems architectures: Defense AR Journal. 16 (3), 2009, 322–341
- [Bal11] BALZERT, H.: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. 3. Auflage, Lehrbücher der Informatik, Spektrum Akademischer Verlag, Heidelberg, 2011
- [Bar98] BARTER, R. H.: A Systems Engineering Pattern Language: INCOSE International Symposium. Vancouver, 1998, 350–353
- [BBK+21] BÖHM, W.; BROY, M.; KLEIN, C.; POHL, K.; RUMPE, B.; SCHRÖCK, S.: Model-Based Engineering of Collaborative Embedded Systems. Springer International Publishing, Cham, 2021
- [BBK16] BONDAVALLI, A.; BOUCHENAK, S.; KOPETZ, H. (Hrsg.): Cyber-Physical Systems of Systems – Foundations - A Conceptual Model and Some Derivations: The AMADEOS Legacy. Programming and Software Engineering Band 10099, Springer, Cham, 2016
- [Ber50] BERTALANFFY, L. VON: The Theory of Open Systems in Physics and Biology: Science. 111 (2872), 1950, S. 23–29
- [Ber72] BERTALANFFY, L. VON: The History and Status of General Systems Theory: Academy of management journal. 15 (4), 1972, S. 407–426
- [BHA+20-ol] BRANDL, D.; HUNKAR, P.; APOLITO, F.; SOEHNER, H.: OPC UA for PackML – Release 1.01. Unter: <https://opcfoundation.org/developer-tools/documents/view/209>, 20. November 2024
- [BMS+20] BAEK, Y.-M.; MIHRET, Z.; SHIN, Y.-J.; BAE, D.-H.: A Modeling Method for Model-based Analysis and Design of a System-of-Systems: 2020 27th Asia-Pacific Software Engineering Conference (APSEC). 01. - 04. Dezember 2020, Singapore, Singapore, IEEE, 2020, S. 336–345
- [BMS22] BOYD, M.; MAZZUCHI, T.; SARKANI, S.: Coevolutionary Network Model for Efficient Collaboration in Systems-of-Systems: 2022 17th Annual System of Systems Engineering Conference (SOSE). 07. - 11. Juni 2022, Rochester, NY, USA, IEEE, 2022, S. 113–118

- [BMW19-ol] BMWI - BUNDESMINISTERIUM FÜR WIRTSCHAFT UND ENERGIE: Leitbild 2030 für Industrie 4.0 – Digitale Ökosysteme global gestalten – Positionspapier. Unter: <https://www.plattform-i40.de/IP/Navigation/DE/Industrie40/Leitbild2030/leitbild-2030.html>, 20. November 2024
- [BOM+23] BEALE, D.; OOSTHUIZEN, R.; MCKINNEY, D.; SMITH, G.; WATSON, M.: Systems Engineering Heuristics for Complex Systems Revisited: 2023 18th Annual System of Systems Engineering Conference (SoSe). 14. - 16. Juni 2023, Lille, France, IEEE, 2023, S. 1–7
- [Bou56] BOULDING, K. E.: General systems theory - the skeleton of science: Management science. 2 (3), 1956, S. 197–208
- [Bri96] BRINKKEMPER, S.: Method engineering: engineering of information systems development methods and tools: Information and Software Technology. 38 (4), 1996, S. 275–280
- [BS06] BOARDMAN, J.; SAUSER, B.: System of Systems - the meaning of of: Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering. 24. - 26. April 2006, Los Angeles, California, USA, 2006, S. 118–123
- [BSS+17] BRÖRING, A.; SCHMID, S.; SCHINDHELM, C.-K.; KHELIL, A.; KABISCH, S.; KRAMER, D.; LE PHUOC, D.; MITIC, J.; ANICIC, D.; TENIENTE, E.: Enabling IoT Ecosystems through Platform Interoperability: IEEE Software. 34 (1), 2017, S. 54–61
- [CB18] CHAMI, M.; BRUEL, J.-M.: A survey on MBSE Adoption Challenges: INCOSE EMEA Sector Systems Engineering Conference. 05. - 07. November 2018, Berlin, 2018
- [CC05] CONWAY, S. R.; CONSIGLIO, M. C.: Towards a Functionally-Formed Air Traffic System-of-Systems: 2005 IEEE International Conference on Systems, Man and Cybernetics. 10. - 12. Oktober 2005, Waikoloa, HI, USA, Band 4, IEEE, 2005, S. 3167–3172
- [CHA21-ol] CHADEMO ASSOCIATION: CHAdEMO Association & Protocol – Brochure. Unter: https://www.chademo.com/wp2016/wp-content/uploads/2021/11/chademo_brochure_2021.pdf, 20. November 2024
- [Cho16] CHOI, T.-M.: Managing service supply chains in the big data era: A system of systems perspective. In: Choi, T.-M. (Ed.): Service supply chain systems – A systems engineering approach. Communications in cybernetics, systems science and engineering, 8, CRC Press, Boca Raton, 2016
- [Clo06] CLOUTIER, R. J.: Applicability of Patterns to Architecting complex Systems. Dissertation, Stevens Institute of Technology, 2006
- [CTE+23] CAMPO, K. X.; TEPPER, T.; EATON, C. E.; SHIPMAN, A. M.; BHATIA, G.; MESMER, B.: Model-based systems engineering: Evaluating perceived value, metrics, and evidence through literature: Systems Engineering. 26 (1), 2023, S. 104–129
- [CV07] CLOUTIER, R. J.; VERMA, D.: Applying the Concept of Patterns to Systems Architecture: Systems Engineering. 10 (2), 2007
- [Cza19] CZAJA, A.: Systematik zur Ableitung bedarfsgerechter Systems Engineering Leitfäden im Maschinenbau. Dissertation, Universität Paderborn, 2019
- [DAR+21] DUMITRESCU, R.; ALBERS, A.; RIEDEL, O.; STARK, R.; GAUSEMEIER, J. (Hrsg.): Engineering in Deutschland - Status quo in Wirtschaft und Wissenschaft. Ein Beitrag zum Advanced Systems Engineering. Paderborn, 2021
- [Dav94] DAVIS, A. M.: Fifteen principles of software engineering: IEEE Software. 11 (6), 1994, S. 94–96
- [DB08] DAHMANN, J. S.; BALDWIN, K. J.: Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering: SysCon 2008 - IEEE International Systems Conference. 07.04-10, Montreal, Canada, 2008, S. 1–7

- [DBB20] DRIDI, C. E.; BENZADRI, Z.; BELALA, F.: System of Systems Engineering: Meta-Modeling Perspective: IEEE 15th International Conference of System of Systems Engineering (SoSE). 02. - 04. Juni 2020, Budapest, Hungary, IEEE, 2020, S. 135–144
- [Del14] DELLIGATTI, L.: SysML distilled – A brief guide to the systems modeling language. Addison-Wesley, Upper Saddle River, NJ, 2014
- [DR16] DAHMANN, J.; ROEDLER, G.: System of Systems Engineering Standards: Insight. 19 (3), 2016, S. 23–26
- [DRL+10] DAHMANN, J.; REBOVICH, G.; LANE, J. A.; LOWRY, R.: System engineering artifacts for SoS: 2010 IEEE International Systems Conference. 05. - 08. April 2010, San Diego, CA, IEEE, 2010, S. 13–17
- [DRL+11] DAHMANN, J.; REBOVICH, G.; LANE, J.; LOWRY, R.; BALDWIN, K.: An implementers' view of systems engineering for systems of systems: IEEE International Systems Conference (SysCon). 04. - 07. April 2011, Montreal, QC, Canada, IEEE, 2011, S. 212–217
- [DRL+11] DAHMANN, J.; REBOVICH, G.; LANE, J. A.; LOWRY, R.: System engineering artifacts for SoS: IEEE Aerospace and Electronic Systems Magazine. 26 (1), 2011, S. 22–28
- [Dud23-ol] DUDEN: System. Unter: <https://www.duden.de/rechtschreibung/System>, 4. November 2024
- [Dum10] DUMITRESCU, R.: Entwicklungssystematik zur Integration kognitiver Funktionen in fortgeschrittene mechatronische Systeme. Dissertation, Universität Paderborn, 2010
- [EKM17] EIGNER, M.; KOCH, W.; MUGGEO, C.: Modellbasierter Entwicklungsprozess cybertronischer Systeme. Springer Vieweg, Berlin, Heidelberg, 2017
- [EM11] ENSTE, U.; MAHNKE, W.: OPC Unified Architecture – Die nächste Stufe der Interoperabilität: Automatisierungstechnik. 59 (7), 2011, S. 397–404
- [Est08] ESTEFAN, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies: INCOSE MBSE Initiative. 25 (8), 2008, S. 1–12
- [Eur22-ol] EUROPEAN COMMISSION: Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020. Unter: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52022PC0454>, 4. November 2024
- [FGD21] FORTE, S.; GÖBEL, J. C.; DICKOPF, T.: System of Systems Lifecycle Engineering Approach integrating Smart Product and Service Ecosystems: Proceedings of the International Conference on Engineering Design (ICED21). 16. - 20. August 2021, Gothenburg, Schweden, 2021, S. 2911–2920
- [FH19] FERNANDEZ, J. L.; HERNANDEZ, C.: Practical model-based systems engineering. Artech House, Boston, 2019
- [FMS12] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A practical guide to SysML – The systems modeling language. 2. Auflage, Morgan Kaufmann, Waltham MA, 2012
- [FPF+17] FALDIK, O.; PAYNE, R.; FITZGERALD, J.; BUHNOVA, B.: Modelling System of Systems Interface Contract Behaviour. In: Kofron, J.; Tumova, J. (Hrsg.): Formal Engineering approaches to Software Components and Architectures. Band 245, 2017, S. 1–15
- [FYW15] FU, K. K.; YANG, M. C.; WOOD, K. L.: Design Principles: The Foundation of Design: 27th International Conference on Design Theory and Methodology. 02. - 05. August 2015, Boston, Massachusetts, Band 7, ASME International, 2015
- [FZZ+23] FENG, Y.; ZHOU, C.; ZOU, Q.; LIU, Y.; LYU, J.; WU, X.: A goal-based approach for modeling and simulation of different types of system-of-systems: Journal of Systems Engineering and Electronics. 34 (3), 2023, S. 627–640

- [GAB+11] GARRETT, R. K.; ANDERSON, S.; BARON, N. T.; MORELAND, J. D.: Managing the Interstitials, a System of Systems Framework Suited for the Ballistic Missile Defense System: Systems Engineering. 14 (1), 2011, S. 87–109
- [GDE+19] GAUSEMEIER, J.; DUMITRESCU, R.; ECHTERFELD, J.; PFÄNDER, T.; STEFFEN, D.; THIELE-MANN, F.: Innovationen für die Märkte von morgen – Strategische Planung von Produkten, Dienstleistungen und Geschäftsmodellen. Hanser, München, 2019
- [GDZ+07] GAUSEMEIER, J.; DANGELMAIER, W.; ZIMMER, D.; SCHMIDT, A.; FRANK, U.; KLOPPER, B.: Using Active Patterns for the Conceptual Design of Self-Optimizing Systems Exemplified by an Air Gap Adjustment System: 27th Computers and Information in Engineering Conference. 04. - 07. September 2007, Las Vegas, Nevada, USA, ASME International, 2007, S. 935–944
- [GFS17] GFSE E.V.: INCOSE Systems Engineering Handbuch – Ein Leitfaden für Systemlebenszyklus-Prozesse und -Aktivitäten. GfSE e.V., München, 2017
- [GG98] GASKA, M. T.; GAUSE, D. C.: An approach for cross-discipline requirements engineering process patterns: Proceedings of IEEE International Symposium on Requirements Engineering: RE '98. 06. - 10. April 1998, Colorado Springs, CO, USA, IEEE, 1998, S. 182–189
- [GGH+23] GÜNTHER, M.; GÖLLNER, D.; HEIHOFF-SCHWEDE, J.; ANACKER, H.; DUMITRESCU, R.: Engineering und Management von Systems of Systems – Gestaltungskonzepte im SoS-Engineering. In: Koch, W.; Wilke, D.; Dreiseitel, S.; Kaffenberger, R. (Hrsg.): Tag des Systems Engineering. 15. - 17. November 2023, Würzburg, 2023, S. 137–143
- [GHJ+93] GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.: Design Patterns: Abstraction and Reuse of Object-Oriented Design: ECOOP '93 - Object-Oriented Programming. 7th European Conference, 26. - 30. Juli 1993, Kaiserslautern, Springer Berlin, Heidelberg, 1993
- [GP14] GAUSEMEIER, J.; PLASS, C.: Zukunftsorientierte Unternehmensgestaltung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. 2. Auflage, Hanser, München, 2014
- [GSB08] GOROD, A.; SAUSER, B.; BOARDMAN, J.: System-of-Systems Engineering Management: A Review of Modern History and a Path Forward: IEEE Systems Journal. 2 (4), 2008, S. 484–499
- [GSB08] GOROD, A.; SAUSER, B.; BOARDMAN, J.: Paradox: Holarchical view of system of systems engineering management: 2008 IEEE International Conference on System of Systems Engineering. 02. - 04. Juni 2008, Monterey, CA, USA, IEEE, 2008, S. 1–6
- [GTS+14] GAUSEMEIER, J.; TRÄCHTLER, A.; SCHÄFER, W.; ANACKER, H.: Semantische Technologien im Entwurf mechatronischer Systeme – Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten. Hanser, München, 2014
- [HB07] HSU, J. C.; BUTTERFIELD, M.: Modeling Emergent Behavior for Systems-of-Systems: INCOSE International Symposium. 17 (1), 2007, S. 1811–1821
- [Hen16] HENSHAW, MICHAEL J. DE C: Systems of Systems, Cyber-Physical Systems, The Internet-Of-Things... Whatever Next?: Insight. 19 (3), 2016, S. 51–54
- [Hes12] HESTER, P. T.: Why optimisation of a system of systems is both unattainable and unnecessary: International Journal of System of Systems Engineering. 3 (3-4), 2012, S. 268–276
- [HMG+91] HOFFMANN, M. W.; MALAKUTI, S.; GRÜNER, S.; FINSTER, S.; GEBHARDT, J.; TAN, R.; SCHINDLER, T.; GAMER, T.: Developing Industrial CPS: A Multi-Disciplinary Challenge: Sensors. 21 (6), 1991
- [HPC+09] HSU, J. C.; PRICE, M.; CLYMER, J.; GARCIA-JR, J.; GONZALEZ, E.: Agent-Based Modeling the Emergent Behavior of A System-of-Systems: INCOSE International Symposium. 19 (1), 2009, S. 1581–1590

- [HS21] HENDERSON, K.; SALADO, A.: Value and benefits of model-based systems engineering (MBSE): Evidence from the literature: *Systems Engineering*. 24 (1), 2021, S. 51–66
- [HSM+23] HAUGEN, R. A.; SKEIE, N.-O.; MULLER, G.; SYVERUD, E.: Detecting emergence in engineered systems: A literature review and synthesis approach: *Systems Engineering*. 26 (4), 2023, S. 463–481
- [HV06] HONOUR, E.; VALERDI, R.: Advancing an Ontology for Systems Engineering to Allow Consistent Measurement: *Proceedings of the Conference on Systems Engineering Research*. 06. - 09. April 2006, Los Angeles, CA, USA, 2006, S. 1–12
- [HWF+19] HABERFELLNER, R.; WECK, O. DE; FRICKE, E.; VÖSSNER, S.: *Systems Engineering*. Springer International Publishing, Cham, 2019
- [IFF+23] IMAMURA, M.; FERREIRA, F.; FERNANDES, J.; GRACIANO N., V. V.; SANTOS, R. P.: Towards a Catalog of Heuristics for the Design of Systems-of-Systems: *Proceedings of the XIX Brazilian Symposium on Information Systems*. 29. Mai - 01 Juni 2023, Maceio, Brazil, 2023, S. 128–135
- [INC15] INCOSE: *Systems Engineering Handbook 4e – A guide for system life cycle processes and activities*. 4th Edition, Wiley, Hoboken, NJ, 2015
- [INC21-ol] INCOSE: *Systems Engineering Vision 2035*. Unter: <https://www.incose.org/about-systems-engineering/se-vision-2035>, 20. November 2024
- [INC23-ol] INCOSE: *About INCOSE*. Unter: https://www.incose.org/about-incose_2, 1. November 2023
- [IPP+14] INGRAM, C.; PAYNE, R.; PERRY, S.; HOLT, J.; HANSEN, F. O.; COUTO, L. D.: *Modelling patterns for systems of systems architectures: IEEE International Systems Conference Proceedings*. 31. März - 03. April 2014, Ottawa, ON, Canada, IEEE, 2014, S. 146–153
- [Ise08] ISERMANN, R.: *Mechatronische Systeme – Grundlagen*. 2. Auflage, Springer, Berlin, Heidelberg, 2008
- [ISO15288] ISO/IEC/IEEE 15288:2023 *Systems and software engineering - System life cycle processes*, 2023
- [ISO21839] ISO/IEC/IEEE 21839:2019, *Systems and software engineering —System of systems (SoS) considerations in life cycle stages of a system*, 2019
- [ISO21840] ISO/IEC/IEEE 21840:2019 *Systems and software engineering—Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems (SoS)*, 2019
- [ISO21841] ISO/IEC/IEEE 21841:2019: *Systems and software engineering - Taxonomy of systems of systems*, 2019
- [ISO42010] ISO/IEC/IEEE 42010:2011 *Systems and software engineering -- Architecture description*, 2011
- [Jam08a] JAMSHIDI, M.: *System of systems engineering - New challenges for the 21st century: IEEE Aerospace and Electronic Systems Magazine*. 23 (5), 2008, S. 4–19
- [Jam08b] JAMSHIDI, M.: *Systems of Systems Engineering: Principles and Applications*. CRC Press, 2008
- [KAG+16] KAGERMANN, H.; ANDERL, R.; GAUSEMEIER, J.; SCHUH, G.; WAHLSTER, W.: *Industrie 4.0 im globalen Kontext – Strategien der Zusammenarbeit mit internationalen Partnern*. aca-tech STUDIE, Herbert Utz Verlag GmbH, München, 2016
- [Kag14] KAGERMANN, H.: *Industrie 4.0 und Smart Services*. In: Brenner, W.; Hess, T. (Hrsg.): *Wirtschaftsinformatik in Wissenschaft und Praxis – Festschrift für Hubert Österle*. Business Engineering, Springer Gabler, Berlin, 2014

- [Kai13] KAISER, L.: Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme. Dissertation, Universität Paderborn, 2013
- [KGM03] KOLP, M.; GIORGINI, P.; MYLOPOULOS, J.: Organizational Patterns for Early Requirements Analysis. In: Eder, J.; Missikoff, M. (Hrsg.): Advanced Information Systems Engineering. CAiSE 2003. Lecture Notes in Computer Science. Band 2681, Springer, Berlin, Heidelberg, 2003, S. 617–632
- [KJT+13] KALAWSKY, R. S.; JOANNOU, D.; TIAN, Y.; FAYOUMI, A.: Using Architecture Patterns to Architect and Analyze Systems of Systems: *Procedia Computer Science*. Band 16, 2013, S. 283–292
- [KLM+04] KAHLERT, T.; LANGENBERG, L.; MARWINSKI, T.; SELL, K.: Engineering Knowledge Management. *ZWF*, (99), 2004, S. 424–427
- [KSO+20] KOSSMANN, M.; SAMHAN, A.; ODEH, M.; QADDOUMI, E.; TBAKHI, A.; WATTS, S.: Extending the scope of configuration management for the development and life cycle support of systems of systems—An ontology-driven framework applied to the Enceladus Submarine Exploration Lander: *Systems Engineering*. 23 (3), 2020, S. 366–391
- [KWH13] KAGERMANN, H.; WAHLSTER, W.; HELBIG, J.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 – Abschlussbericht des Arbeitskreises Industrie 4.0, 2013
- [Lan08] LANE, D. C.: The Emergence and Use of Diagramming in System Dynamics: A Critical Account: *Systems Research and Behavioral Science*. 25 (1), 2008, S. 3–23
- [LLP11] LOPES, A. J.; LEZAMA, R.; PINEDA, R.: Model Based Systems Engineering for Smart Grids as Systems of Systems: *Procedia Computer Science*. Band 6, 2011, S. 441–450
- [Mai15] MAIER, M. W.: The Role of Modeling and Simulation in System of Systems Development. In: Rainey, L. B.; Tolk, A. (Eds.): Modeling and Simulation Support for System of Systems Engineering Applications. Wiley, Hoboken, 2015, pp. 11–41
- [Mai98] MAIER, M. W.: Architecting Principles for Systems-of-Systems: *The Journal of the International Council on Systems Engineering*. 1 (4), 1998, S. 267–284
- [MAS23] MADNI, A. M.; AUGUSTINE, N.; SIEVERS, M.: Handbook of Model-Based Systems Engineering. Springer, Schweiz, 2023
- [mat24-ol] MATTER-SMARTHOME: Raumthermostate mit Matter von Eberle Controls. Unter: <https://matter-smarthome.de/produkte/raumthermostate-mit-matter-von-eberle-controls/>, 20. November 2024
- [MGW+09] MANSOURI, M.; GOROD, A.; WAKEMAN, T. H.; SAUSER, B.: Maritime Transportation System of Systems management framework: a System of Systems Engineering approach: *International Journal of Ocean Systems Management*. 1 (2), 2009, S. 200–226
- [MJI+19] MOHSIN, A.; JANJUA, N. K.; ISLAM, S. M. S.; GRACIANO NETO, V. V.: Modeling Approaches for System-of-Systems Dynamic Architecture: Overview, Taxonomy and Future Prospects: 14th Annual Conference System of Systems Engineering (SoSE). IEEE, 2019, pp. 49–56
- [Moc21] MOCKENHAUPT, A.: Digitalisierung und Künstliche Intelligenz in der Produktion – Grundlagen und Anwendung. Lehrbuch, Springer Vieweg, Wiesbaden, 2021
- [MOD16] MORDECAI, Y.; ORHOF, O.; DORI, D.: Model-Based Interoperability Engineering in Systems-of-Systems and Civil Aviation: *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 48 (4), 2016, S. 637–648
- [Mog06] MOGUL, J. C.: Emergent (mis)behavior vs. complex software systems: *ACM SIGOPS Operating Systems Review*. 40 (4), 2006, S. 293–304

- [MR15] MITTAL, S.; RAINEY, L. B.: Harnessing Emergence: The Control and Design of Emergent Behavior in System of Systems Engineering: Proceedings of the conference on summer computer simulation. Chicago, IL, USA, 2015, S. 1–10
- [MSB09] MEENTEMEYER, S. M.; SAUSER, B.; BOARDMAN, J.: Analysing a system of systems characterisation to define system of systems engineering practices: International Journal of System of Systems Engineering. 1 (3), 2009, S. 329–346
- [NL18] NCUBE, C.; LIM, S. L.: On Systems of Systems Engineering: A Requirements Engineering Perspective and Research Agenda: 2018 IEEE 26th International Requirements Engineering Conference (RE). 20. - 24. August 2018, Banff, AB, IEEE, 2018, S. 112–123
- [NLF+15] NIELSEN, C. B.; LARSEN, P. G.; FITZGERALD, J.; WOODCOCK, J.; PELESKA, J.: Systems of Systems Engineering: ACM Computing Surveys. 48 (2), 2015, S. 1–41
- [Nor11] NORTH, K.: Wissensorientierte Unternehmensführung – Wertschöpfung durch Wissen. 5. Auflage, Gabler Lehrbuch, Gabler Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2011
- [NT12] NONAKA, I.; TAKEUCHI, H.: Die Organisation des Wissens – Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen. Campus Verlag GmbH, Frankfurt am Main, 2012
- [NVM13] NICKERSON, R. C.; VARSHNEY, U.; MUNTERMANN, J.: A method for taxonomy development and its application in information systems: European Journal of Information Systems. 22 (3), 2013, S. 336–359
- [OPC23-ol] OPC FOUNDATION: Members. Unter: <https://opcfoundation.org/members>, 21. November 2023
- [PB21] PAHL, G.; BEITZ, W.: Gestaltungsprinzipien. In: Bender, B.; Gericke, K. (Hrsg.): Pahl/Beitz Konstruktionslehre. Springer, Berlin, Heidelberg, 2021, S. 525–565
- [PBB16] PETITDEMANGE, F.; BORNE, I.; BUISSON, J.: Assisting the evolutionary development of SoS with reconfiguration patterns. In: Bahsoon, R.; Weinreich, R. (Hrsg.): Proceedings of the 10th European Conference on Software Architecture Workshops. ECSAW '16: European Conference on Software Architecture Workshops, Copenhagen Denmark, ACM, New York, NY, USA, 2016, S. 1–7
- [Pet03] PETERS, J. F.: Design patterns in intelligent systems. In: Zhong, N.; Ras, Z. W.; Tsumoto, S.; Suzuki, E. (Eds.): Foundations of intelligent systems – 14th international symposium. 28. - 31. October 2003, Maebashi City, Japan, Springer Science & Business Media, 2003
- [PH14] PORTER, M. E.; HEPPELMANN, J. E.: How Smart, Connected Products Are Transforming Competition: Harvard Business Review. 92 (11), 2014, S. 64–88
- [PJ20] PEREIRA PESSÔA, M. V.; JAUREGUI BECKER, J. M.: Smart design engineering: a literature review of the impact of the 4th industrial revolution on product design and development: Research in Engineering Design. 31 (2), 2020, S. 175–195
- [PKA+20] PELLICCIONE, P.; KNAUSS, E.; AGREN, S. M.; HELDAL, R.; BERGENHEM, C.; VINEL, A.; BRUNNEGÅRD, O.: Beyond connected cars: A systems of systems perspective: Science of Computer Programming. Band 191, 2020, S. 1–21
- [PKN+21] PRAKASHA, P. S.; KILKIS, S.; NAEEM, N.; RATEI, P.; NAGEL, B.: System of Systems Simulation Driven Wildfire Fighting Aircraft Design and Fleet Assessment: AIAA Aviation 2021 Forum. 2. -6. August, 2021, S. 1–18
- [Plö09] PLÖBL, K.: Mehrseitig sichere Ad-hoc-Vernetzung von Fahrzeugen. Dissertation. DuD-Fachbeiträge, Gabler, Wiesbaden, 2009
- [Pol66] POLANYI, M.: The Tacit Dimension, 1966

- [PRR10] PROBST, G. J. B.; RAUB, S.; ROMHARDT, K.: Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen. 6. Auflage, Gabler, Wiesbaden, 2010
- [PTG+06] PEFFERS, K.; TUUNANEN, T.; GENGLER, C. E.; ROSSI, M.; HUI, W.; VIRTANEN, V.; BRAGGE, J.: The Design Science Research Process: A Model for Producing and Presenting Information Systems Research: Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006). Claremont, CA, USA, 2006
- [RHO12] RENDON, R. G.; HUYNH, T. V.; OSMUNDSON, J. S.: Contracting processes and structures for systems-of-systems acquisition: Systems Engineering. 15 (4), 2012, S. 471–482
- [Ris07] RISING, L.: Understanding the Power of Abstraction in Patterns: IEEE Software. 24 (4), 2007, S. 46–51
- [Ris99] RISING, L.: Patterns: A Way to Reuse Expertise: IEEE Communications Magazine. 37 (4), 1999, S. 34–36
- [RM21] ROGERS, E. B.; MITCHELL, S. W.: MBSE delivers significant return on investment in evolutionary development of complex SoS: Systems Engineering. 24 (6), 2021, S. 385–408
- [Rop09] ROPOHL, G.: Allgemeine Technologie: Eine Systemtheorie der Technik. 3. Auflage, Universitätsverlag Karlsruhe, Karlsruhe, 2009
- [Rou18] ROUSSEAU, D.: On the Architecture of Systemology and the Typology of Its Principles: Systems. 6 (1), 2018, S. 1–17
- [RRR14] RICCI, N.; RHODES, D. H.; ROSS, A. M.: Evolvability-Related Options in Military Systems of Systems: Procedia Computer Science. Conference on Systems Engineering Research (CSER 2014), Band 28, 2014, S. 314–321
- [SA14] SELBERG, S. A.; AUSTIN, M. A.: Toward an Evolutionary System of Systems Architecture: INCOSE International Symposium. 18 (1), 2014, S. 1065–1078
- [SB07] SAGE, A. P.; BIEMER, S. M.: Processes for System Family Architecting, Design, and Integration: IEEE Systems Journal. 1 (1), 2007, S. 5–16
- [Sch05] SCHUH, G.: Produktkomplexität managen – Strategien - Methoden - Tools. 2. Auflage, Hanser, München, 2005
- [SHP+11] SEIN, M. K.; HENFRIDSSON, O.; PURAO, S.; ROSSI, M.; LINDGREN, R.: Action Design Research: MIS Quarterly. 35 (1), 2011, S. 37–56
- [SM09] SHEARD, S. A.; MOSTASHARI, A.: Principles of Complex Systems for Systems Engineering: Systems Engineering. 12 (4), 2009, S. 295–311
- [SR19] SHAKED, A.; REICH, Y.: Designing development processes related to system of systems using a modeling framework: Systems Engineering. 22 (6), 2019, S. 561–575
- [SRB+07] SHAH, N. B.; RICHARDS, M. G.; BRONIATOWSKI, D. A.; LARACY, J. R.; SPRINGMANN, P. N.: System of Systems Architecture: The Case of Space Situational Awareness: AIAA SPACE 2007 Conference & Exposition. 18. - 20. September 2007, Long Beach, CA, USA, 2007
- [Sta73] STACHOWIAK, H.: Allgemeine Modelltheorie. Springer, Wien, New York, 1973
- [Str98] STRUBE, G.: Modelling Motivation and Action Control in Cognitive Systems. In: Schmid, U.; Krems, J. F.; Wysocki, F. (Hrsg.): Mind Modelling – A Cognitive Science Approach to Reasoning, Learning and Discovery. Berlin, 1998, S. 89–108
- [SZ03] SANZ, R.; ZALEWSKI, J.: Pattern-based control systems engineering: IEEE Control Systems Magazine. 23 (3), 2003, S. 43–60
- [Tek19] TEKINERDOGAN, B.: Multi-Dimensional Classification of System-of-Systems: 14th Annual Conference System of Systems Engineering (SoSE). IEEE, 2019, pp. 278–283

- [Tek22] TEKINERDOGAN, B.: Obstacles of System-of-Systems: 2022 IEEE International Symposium on Systems Engineering (ISSE). 24. - 26. Oktober 2022, Vienna, Austria, IEEE, 2022, S. 1–7
- [Tsc16] TSCHIRNER, C.: Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme. Dissertation, Heinz Nixdorf Institut, Universität Paderborn, 2016
- [Tur07] TURNER, R.: Toward Agile Systems Engineering Processes: The Journal of Defense Software Engineering, 2007, S. 11–15
- [UM15] UDAY, P.; MARAIS, K.: Designing Resilient Systems-of-Systems: A Survey of Metrics, Methods, and Challenges: Systems Engineering. 18 (5), 2015, S. 491–510
- [Uni89] UNITED STATES SENATE AND THE COMMITTEE ON ARMED SERVICES: Restructuring of the Strategic Defense Initiative (SDI) Program. United States Government Printing Office, 1989
- [VDI2206a] VDI 2206:2004 Entwicklungsmethodik für mechatronische Systeme, 2004
- [VDI2206b] VDI/VDE 2206:2021 Entwicklung mechatronischer und cyber-physischer Systeme, 2021
- [VDI5610] VDI5610:2009 Wissensmanagement im Ingenieurwesen – Grundlagen, Konzepte, Vorgehen, 2009
- [VRR07] VALERDI, R.; ROSS, A. M.; RHODES, D. H.: A Framework for Evolving System of Systems Engineering: The Journal of Defense Software Engineering, 2007, S. 28–30
- [WC18] WYNN, D. C.; CLARKSON, P. J.: Process models in design and development: Research in Engineering Design. Band 29, 2018, S. 161–202
- [WD01] WOOD, M. F.; DELOACH, S. A.: An Overview of the Multiagent Systems Engineering Methodology. In: Ciancarini, P.; Wooldridge, M. (Hrsg.): Agent-Oriented Software Engineering. AOSE 2000. Lecture Notes in Computer Science. Band 1957, Springer, Berlin, Heidelberg, 2001, S. 207–221
- [Wei06] WEILKIENS, T.: Systems engineering mit SysML/UML – Modellierung, Analyse, Design. dpunkt. verlag, Heidelberg, 2006
- [Wes17] WESTERMANN, T.: Systematik zur Reifegradmodell-basierten Planung von Cyber-Physical Systems des Maschinen- und Anlagenbaus. Dissertation, Universität Paderborn, 2017
- [Wie14] WIERINGA, R.: Design science methodology for information systems and software engineering – for Information Systems and Software Engineering. Springer, Berlin, Heidelberg, 2014
- [WJ11] WHITE, B. E.; JEAN, P. N.: Case study in system of systems engineering: NASA's Advanced Communications Technology Satellite: 2011 6th International Conference on System of Systems Engineering (SoSe). 27. - 30. Juni 2011, Albuquerque, NM, USA, IEEE, 2011, S. 237–244
- [WLW+16] WEILKIENS, T.; LAMM, J. G.; WALKER, M.; ROTH, S.: Model-based system architecture. John Wiley & Sons Inc, Hoboken, New Jersey, 2016
- [WMG+24] WIECHER, C.; MANDEL, C.; GÜNTHER, M.; FISCHBACH, J.; GREENYER, J.; GREINERT, M.; WOLFF, C.; DUMITRESCU, R.; MENDEZ, D.; ALBERS, A.: Model-based analysis and specification of functional requirements and tests for complex automotive systems: Systems Engineering. 27 (4), 2024, S. 728–744
- [WSR+23] WALDEN, D. D.; SHORTELL, T. M.; ROEDLER, G. J.; DELICADO, B. A.; MORNAS, O.; YEW-SENG, Y.; ENDLER, D.: INCOSE Systems Engineering Handbook – A Guide for system life cycle processes and activities. 5. Auflage, John Wiley & Sons, 2023
- [XW19] XIAO, Y.; WATSON, M.: Guidance on Conducting a Systematic Literature Review: Journal of Planning Education and Research. 39 (1), 2019, S. 93–112

- [YNF19] YASSINE, S.; NAJIB, E. K.; FATIMA, L.: A Survey: Centralized, Decentralized, and Distributed Control Scheme in Smart Grid Systems: 2019 7th Mediterranean Congress of Telecommunications (CMT). IEEE, 2019, pp. 1–11
- [YZT+19] YANG, T.; ZHANG, Y.; TAN, J.; QIU, T. Z.: Research on Forward Collision Warning System Based on Connected Vehicle V2V Communication: 5th International Conference on Transportation Information and Safety (ICTIS). 14. - 17. Juli 2019, Liverpool, UK, IEEE, 2019, pp. 1174–1181
- [Zin13] 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. Dissertation, Karlsruher Institut für Technologie, 2013
- [ZJK23] ZEGEYE, W.; JEMAL, A.; KORNEGAY, K.: Connected Smart Home over Matter Protocol: 2023 IEEE International Conference on Consumer Electronics (ICCE). 06. - 08. Januar 2023, Las Vegas, NV, USA, IEEE, 2023, S. 1–7

Anhang

A1	Ergänzungen zur Klassifizierung von SoS	191
A1.1	Übersicht der Anwendungsbeispiele	191
A1.2	Clusteranalyse anhand der Merkmale von SoS	195
A2	Ergänzungen zum Gestaltungsrahmen	199
A2.1	Methodensteckbriefe	199
A2.2	Designprinzipien.....	204
A2.3	Lösungsmuster.....	207

A1 Ergänzungen zur Klassifizierung von SoS

Im Folgenden wird die durchgeführte Clusteranalyse der SoS-Beispiele im Rahmen der Arbeit vorgestellt (vgl. Kapitel 4.2.2). Hierzu wird zunächst die Datenbasis vorgestellt (Anhang A1.1). Im Anschluss folgt die methodische Vorgehensweise sowie die Ergebnisse der Clusteranalyse (Anhang A1.2).

A1.1 Übersicht der Anwendungsbeispiele

Im Rahmen der Arbeit wurden unterschiedliche SoS-Beispiele analysiert, um Gemeinsamkeiten zu identifizieren. Insgesamt wurden zehn Anwendungsbeispiele von Industriepartnern im Forschungsprojekt MoSyS und zwölf Beispiele aus der Literatur in die Analyse einbezogen. Sämtliche Beispiele wurden in einem einheitlichen Schema anhand des in Bild A-1 dargestellten Steckbriefs in Workshops mit Forschenden und Anwendungspartnern aus dem Projekt MoSyS aufbereitet.

Die Bewertung der einzelnen Beispiele wurde anhand der in Kapitel 4.2.1 vorgestellten charakteristischen Merkmale von SoS durchgeführt. Die Nummerierung der Merkmale folgt den Ausführungen in Kapitel 4.2.1. Um die Leserlichkeit der nachfolgenden Steckbriefe zu unterstützen, werden sie hier noch einmal aufgeführt:

1. Managementunabhängigkeit
2. Operative Unabhängigkeit
3. Verteilung
4. Evolution
5. Dynamische Rekonfiguration
6. Emergentes Verhalten
7. Interdependenz
8. Interoperabilität

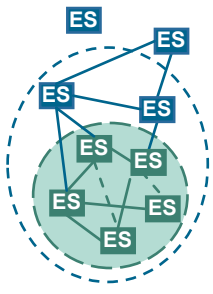
SoS-Steckbrief	
Name: ...	
Kurzbeschreibung <i>(Welche Ziele hat das SoS? Welche Anwendungsfälle werden realisiert? ...)</i>	Einordnung in SoS-Netzwerke <i>(Welcher Kontext ist zu betrachten? Welche Systeme und SoS sind von Interesse?)</i>
	 Zusammenhängende SoS ▪ ... ▪ ...  SoS von Interesse ▪ ...  Eigenständige Systeme ▪ ... ▪ ... ▪ ... 
Managementunabhängigkeit abhängig ————— unabhängig 1 ————— 2 ————— 3	Funktionales Konzept <i>(Welche Funktionalitäten werden benötigt und wie hängen diese zusammen?)</i>
Operative Unabhängigkeit abhängig ————— unabhängig 1 ————— 2 ————— 3	
Räumliche Verteilung lokal ————— global 1 ————— 2 ————— 3	
Evolution gering ————— stark 1 ————— 2 ————— 3	
Dynamische Rekonfiguration gering ————— hoch 1 ————— 2 ————— 3	Logisches Konzept <i>(Welche eigenständigen Systeme sind beteiligt und wie interagieren diese? Welche Schnittstellen sind zu betrachten?)</i>
Emergentes Verhalten gering ————— stark 1 ————— 2 ————— 3	
Interdependenz hoch ————— gering 1 ————— 2 ————— 3	
Interoperabilität hoch ————— gering 1 ————— 2 ————— 3	

Bild A-1: Beispielsteckbrief zur Beschreibung eines SoS

Tabelle A-1: Anwendungsbeispiele aus der Industrie

ID	Titel	Kurzbeschreibung	SoS-Merkmale							
			1	2	3	4	5	6	7	8
1	Verbrauchsmodell für Smart Mobility	Das SoS umfasst eine herstellerunabhängige „Bereitstellung von Verbrauchsvorhersagen“ in dem Daten aus einem Fahrzeug in einen eigenständigen Cloudservice mit weiteren relevanten externen Daten ergänzt und ausgewertet wird, um ein Verbrauchsmodell für eine Strecke eines Fahrzeugs bereitzustellen.	3	2	3	2	2	1	2	3
2	Komponentenbewegung in der Fertigung	Das SoS umfasst die Verarbeitung eines Rohlings über mehrere Bearbeitungssysteme mithilfe von Automatisierungslösungen. Dabei wird eine flexible Anordnung der Bearbeitungsschritte mithilfe von Standard-Automatisierungslösungen ermöglicht.	2	2	1	2	2	2	2	2
3	Autonome Zu- und Ablieferung	Die Produktionsanlage soll mittels fahrerloser Transportsysteme automatisch mit Komponenten beliefert werden und die Endprodukte entnommen werden. Das Zusammenspiel der Systeme wird durch ein automatisches Auslösen von „Lieferungsaufgaben“ initiiert. Die Interaktion wird standardisiert über OPC UA.	2	2	2	2	2	2	2	3
4	Erntekette	Das SoS umfasst Erntesysteme (Traktoren, Anbaugeräte und Terminals), die herstellerübergreifend über ISOBUS kommunizieren und für verschiedene Aufgaben interagieren.	2	2	2	1	2	1	2	2
5	Teilehandling	Das SoS umfasst den Transport von Komponenten und Baugruppen zwischen Fertigungsanlagen, die per Plug-and-Play flexibel kombiniert werden können.	1	1	1	1	2	1	2	1
6	Verpackungslinie	Das SoS umfasst die Bearbeitung und den Transport von Verpackungsgütern zwischen Automatisierungssystemen, die autark Aufgaben übernehmen.	2	2	2	2	2	1	1	2
7	OPL-Kommunikationslinie in der Produktion	Das SoS umfasst eine Liniensteuerung, in der Maschinen in einer Produktionslinie integriert sind und mit Vorgänger und Nachgänger per OPL (open programming language) kommunizieren.	2	1	1	1	1	2	1	2
8	Flexible Produktion (gemanagt)	Das SoS umfasst die zentrale Analyse von Maschinendaten für ein optimales Zusammenspiel zwischen flexiblen Produktionsanlagen.	1	1	1	1	2	2	3	2
9	Intelligentes Laden von Fzg.	Das SoS umfasst das intelligente Laden von Fahrzeugen im Zusammenspiel mit der Energieinfrastruktur. Dabei soll das Fahrzeug im Austausch mit der Energieinfrastruktur entscheiden, ob und wann es optimal geladen werden kann.	2	2	2	1	1	2	2	2
10	Patientenstatuskommunikation	Das SoS umfasst den herstellerübergreifenden Austausch von Informationen über Patienten. Die Patientenhistorie („digitale Krankenakte“) wird nach jeder Behandlung durch die jeweiligen Informationen erweitert.	3	2	2	2	2	1	1	2

Tabelle A-2: Anwendungsbeispiele aus der Literatur

ID	Titel	Quelle	Kurzbeschreibung	SoS-Merkmale							
				1	2	3	4	5	6	7	8
11	Maritime Transportation SoS	[MGW+09]	Maritime Transportsysteme ermöglichen den Transfer großer Warenmengen über internationale Grenzen hinweg.	3	2	3	2	3	3	2	3
12	Flottenfahrt	[Axe19a]	Die Flottenfahrt beschreibt einen Anwendungsfall, in dem mindestens zwei Systeme (LKW) über eine bestimmte Distanz hintereinander herfahren.	2	3	2	2	2	2	3	2
13	Smart Grid	[LLP11]	Im Smart Grid laufen Informationen über aktuelle Energieerzeugung und -verbrauch zusammen, wodurch eine gezielte Steuerung möglich ist.	2	2	3	2	2	2	2	3
14	Raketenabwehr	[GAB+11]	„Plug and Fight“: automatische Erkennung und Zusammenstellung verschiedener Systeme wie Sensoren und Waffensystemen zu einem SoS.	3	2	2	3	2	2	1	2
15	Weltraumbeobachtung	[SRB+07]	Bei katalogisierten Objekten werden abwechselnd kurze Beobachtungen durchgeführt, um Bahndaten zu aktualisieren. Dies wird über den Datenaustausch der verschiedenen Sensorsysteme ermöglicht, um den Abdeckungsgrad zu erhöhen.	2	1	3	1	2	2	1	2
16	Krisenbewältigung	[PKN+21]	Im Katastrophenfall müssen in kürzester Zeit verschiedenste Akteure von einem Krisenstab koordiniert und eingesetzt werden. Das SoS umfasst interoperable Organisationseinheiten, wie Polizei, Feuerwehr oder THW.	3	3	2	3	3	3	2	3
17	Fahrzeug-Ad-hoc-Netz	[Plö09]	Ein Fahrzeug-Ad-hoc-Netzwerk (VANet) ist ein mobiles Ad-hoc-Netz. Es besteht aus mobilen Knoten (Fahrzeugen) und Basisstationen entlang des Straßen- bzw. Verkehrsnetzes.	3	3	2	2	2	3	2	2
18	Air-Traffic-Control	[CC05]	Die Flugsicherung stellt die Sicherheit im Luftverkehr sicher und muss dabei eine Vielzahl unterschiedlicher Interessen konkurrierender Teilnehmer berücksichtigen.	2	1	3	2	2	2	2	2
19	Integrated Deepwater Systems	[GSB08]	Das Integrated Deepwater System Program (IDS-Programm oder Deepwater) war ein auf 25 Jahre angelegtes Programm zum Ersatz der gesamten Ausrüstung der US-Küstenwache (Flugzeuge, Schiffe, Logistik- und Kommando- und Kontrollsysteme).	1	2	2	1	2	2	1	1
20	Unfallbewältigung mit vernetzten Fahrzeug	[PKA+20]	Ein Unfall wird von einem Fahrzeug (Sensoren) erkannt und entsprechend an relevante Verkehrsteilnehmer (weitere Fahrzeuge) und der Verkehrsinfrastruktur (Ampeln) kommuniziert.	2	2	2	2	2	2	2	1
21	Home-Thermostatenmanagement („matter“-Standard)	i.A.a. [mat24-ol]	Die Wärmeregulierung im Smart Home wird mithilfe unterschiedlicher Systeme optimal reguliert. Über den „matter“-Standard wird die Kommunikation zwischen den unterschiedlichen Systemen im Smart Home reguliert.	2	2	1	2	3	2	2	2
22	Forward collision warning („V2X“-Standard)	[YZT+19]	In diesem Anwendungsfall wird eine Kollisionswarnung, die von einem vorausfahrenden Fahrzeug erkannt wird, an ein nachfolgendes Fahrzeug weitergegeben, um eine Kollision zu vermeiden.	2	2	2	2	3	2	2	2

A1.2 Clusteranalyse anhand der Merkmale von SoS

Das Ziel einer Clusteranalyse ist eine Unterteilung einer Menge von Objekten in Gruppen (Clustern). Diese Objekte sollen innerhalb der Gruppen möglichst homogen und im Vergleich zwischen den Gruppen möglichst heterogen sein. Das Vorgehen einer Clusteranalyse beschreiben GAUSEMEIER und PLASS [GP14, S. 65]. Im ersten Schritt wurden die Merkmale und Ausprägungen festgelegt und in einer Datenbasis zusammengeführt (Tabelle A-1 und A-2). Im Anschluss wurden die Distanzen zwischen den Objekten berechnet. Hierzu wurde die quadrierte euklidische Distanz mithilfe des Statistiktools SPSS gebildet. Das Ergebnis ist eine Distanzmatrix, die in Tabelle A-3 aufgeführt ist und die Ähnlichkeit zwischen Objekten repräsentiert.

Tabelle A-3: Näherungsmatrix durch Berechnung der „quadrierten euklidischen Distanz“

Fall	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0,0	7,0	3,0	4,0	14,0	4,0	11,0	13,0	6,0	3,0	5,0	6,0	2,0	5,0	6,0	8,0	7,0	4,0	12,0	7,0	8,0	5,0
2	7,0	0,0	2,0	3,0	5,0	3,0	4,0	4,0	3,0	4,0	8,0	3,0	5,0	4,0	7,0	7,0	4,0	5,0	5,0	2,0	1,0	2,0
3	3,0	2,0	0,0	3,0	9,0	3,0	6,0	6,0	3,0	4,0	4,0	3,0	1,0	4,0	5,0	5,0	4,0	3,0	7,0	4,0	3,0	2,0
4	4,0	3,0	3,0	0,0	4,0	2,0	5,0	5,0	2,0	3,0	9,0	4,0	4,0	7,0	4,0	12,0	7,0	4,0	4,0	3,0	4,0	3,0
5	14,0	5,0	9,0	4,0	0,0	6,0	5,0	3,0	6,0	9,0	19,0	10,0	12,0	13,0	8,0	22,0	15,0	8,0	4,0	5,0	6,0	7,0
6	4,0	3,0	3,0	2,0	6,0	0,0	5,0	9,0	4,0	1,0	9,0	6,0	4,0	3,0	4,0	10,0	7,0	4,0	4,0	3,0	4,0	3,0
7	11,0	4,0	6,0	5,0	5,0	5,0	0,0	6,0	3,0	6,0	14,0	11,0	9,0	8,0	5,0	17,0	10,0	7,0	5,0	6,0	7,0	8,0
8	13,0	4,0	6,0	5,0	3,0	9,0	6,0	0,0	5,0	12,0	14,0	7,0	9,0	14,0	9,0	17,0	12,0	7,0	7,0	6,0	5,0	6,0
9	6,0	3,0	3,0	2,0	6,0	4,0	3,0	5,0	0,0	5,0	9,0	4,0	4,0	7,0	4,0	12,0	5,0	4,0	4,0	3,0	6,0	5,0
10	3,0	4,0	4,0	3,0	9,0	1,0	6,0	12,0	5,0	0,0	8,0	7,0	5,0	2,0	5,0	9,0	6,0	5,0	7,0	4,0	5,0	4,0
11	5,0	8,0	4,0	9,0	19,0	9,0	14,0	14,0	9,0	8,0	0,0	7,0	3,0	6,0	7,0	3,0	4,0	5,0	13,0	8,0	7,0	4,0
12	6,0	3,0	3,0	4,0	10,0	6,0	11,0	7,0	4,0	7,0	7,0	0,0	4,0	7,0	10,0	6,0	3,0	6,0	8,0	3,0	4,0	3,0
13	2,0	5,0	1,0	4,0	12,0	4,0	9,0	9,0	4,0	5,0	3,0	4,0	0,0	5,0	4,0	6,0	5,0	2,0	8,0	5,0	6,0	3,0
14	5,0	4,0	4,0	7,0	13,0	3,0	8,0	14,0	7,0	2,0	6,0	7,0	5,0	0,0	7,0	5,0	4,0	5,0	9,0	4,0	5,0	4,0
15	6,0	7,0	5,0	4,0	8,0	4,0	5,0	9,0	4,0	5,0	7,0	10,0	4,0	7,0	0,0	14,0	9,0	2,0	4,0	5,0	8,0	5,0
16	8,0	7,0	5,0	12,0	22,0	10,0	17,0	17,0	12,0	9,0	3,0	6,0	6,0	5,0	14,0	0,0	3,0	10,0	16,0	9,0	6,0	5,0
17	7,0	4,0	4,0	7,0	15,0	7,0	10,0	12,0	5,0	6,0	4,0	3,0	5,0	4,0	9,0	3,0	0,0	7,0	9,0	4,0	5,0	4,0
18	4,0	5,0	3,0	4,0	8,0	4,0	7,0	7,0	4,0	5,0	5,0	6,0	2,0	5,0	2,0	10,0	7,0	0,0	6,0	3,0	6,0	3,0
19	12,0	5,0	7,0	4,0	4,0	4,0	5,0	7,0	4,0	7,0	13,0	8,0	8,0	9,0	4,0	16,0	9,0	6,0	0,0	3,0	6,0	5,0
20	7,0	2,0	4,0	3,0	5,0	3,0	6,0	6,0	3,0	4,0	8,0	3,0	5,0	4,0	5,0	9,0	4,0	3,0	3,0	0,0	3,0	2,0
21	8,0	1,0	3,0	4,0	6,0	4,0	7,0	5,0	6,0	5,0	7,0	4,0	6,0	5,0	8,0	6,0	5,0	6,0	6,0	3,0	0,0	1,0
22	5,0	2,0	2,0	3,0	7,0	3,0	8,0	6,0	5,0	4,0	4,0	3,0	3,0	4,0	5,0	5,0	4,0	3,0	5,0	2,0	1,0	0,0

Auf Basis der Distanzen wurde im nächsten Schritt eine Clustermethode angewendet, um ähnliche Objekte zu verbinden. Bild A-2 zeigt das Ergebnis mithilfe des Average Linkage

Verfahrens. Die rote Linie visualisiert die Bestimmung einer möglichen Anzahl an Clustern mit möglichst homogenen Objekten innerhalb der Cluster und heterogenen Eigenschaften zwischen den Clustern. Dies wird durch den Abstand zwischen der Zusammenführung von Objekten deutlich. Das Dendrogramm zeigt, dass die Daten keine eindeutige Identifikation von Cluster zulassen. Daher wurden die Ergebnisse nicht als finales Ergebnis im Rahmen der Arbeit verwendet, sondern sind als ergänzende Information in die Klassifikation von SoS mit eingeflossen.

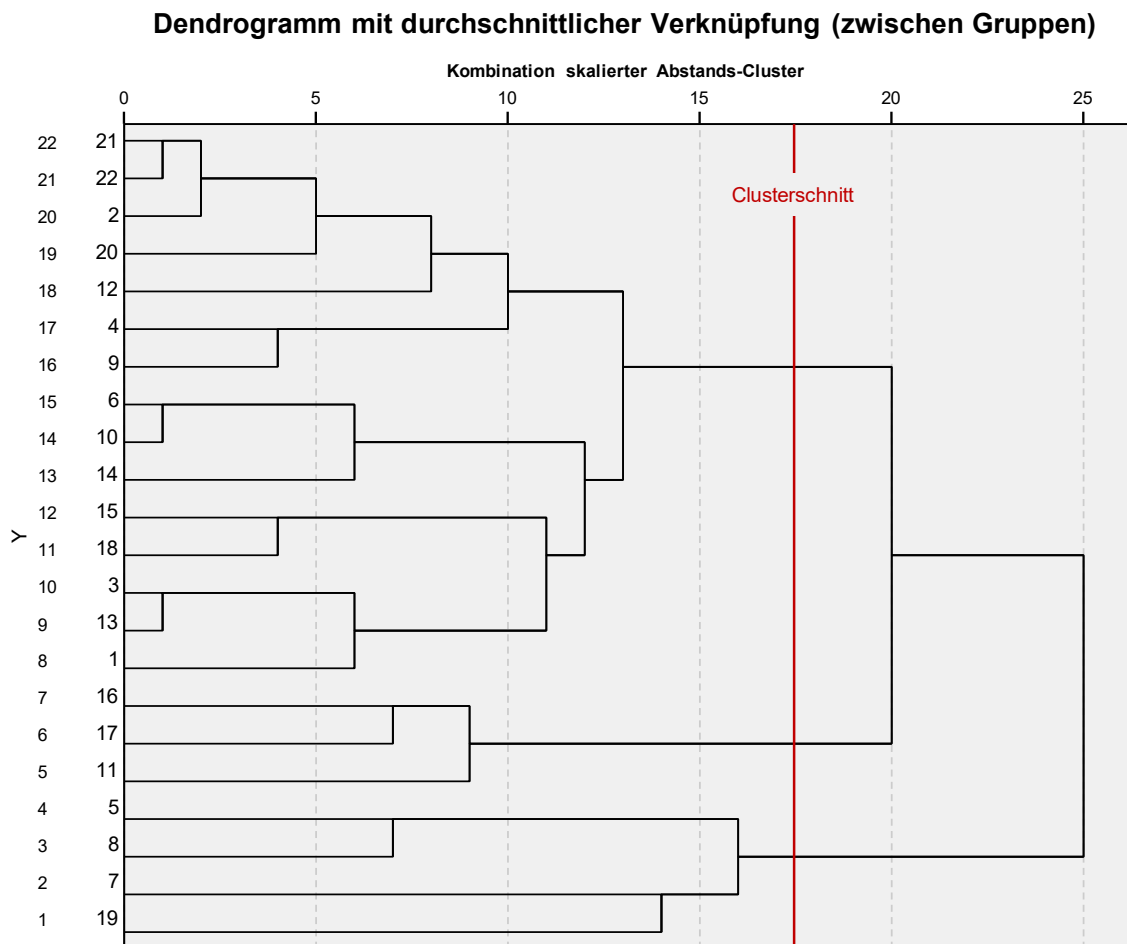


Bild A-2: Dendrogramm mithilfe des Average Linkage Verfahrens zur Bestimmung von Clustern

Zur Visualisierung der Cluster, wurde das Ergebnis mithilfe einer multidimensionalen Skalierung aufbereitet. Die Ausgangsgraphik zur Visualisierung der Cluster wird in Bild A-3 dargestellt.

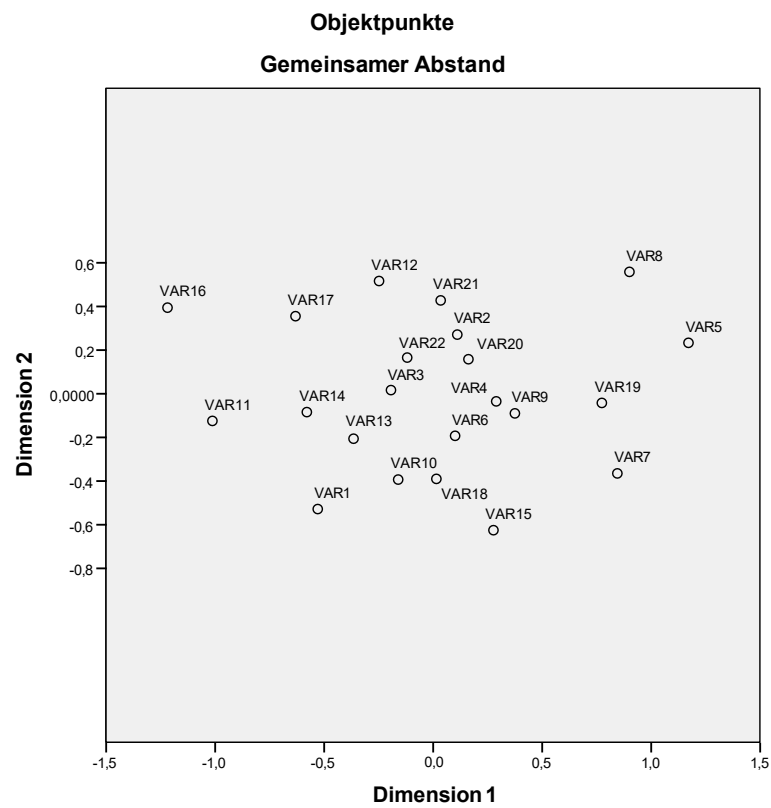


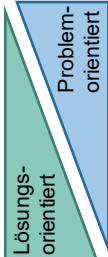
Bild A-3: Multidimensionale Skalierung der SoS-Beispiele auf Basis der Distanzen

A2 Ergänzungen zum Gestaltungsrahmen

A2.1 Methodensteckbriefe

Wie in Kapitel 4.3.2 beschrieben, existieren zahlreiche Methoden und Ansätze zur Entwicklung von Systemen. Im Folgenden werden die im Kontext der Arbeit und im Kontext des Forschungsprojekts MoSyS analysierten und prototypisch betrachteten Methoden mit Relevanz zum Betrachtungsgegenstand der Dissertation aufgeführt. Diese ergänzen die in Kapitel 4.3.2 beschriebenen Methoden und sollen Anwendern Anhaltspunkte für unterschiedliche Problemstellungen geben. Die hier aufgeführten Methoden bilden einen Querschnitt aus unterschiedlichen Bereichen und erheben keinen Anspruch auf Vollständigkeit. Weitere Methoden können auf Basis des Methodensteckbriefs erweitert werden.

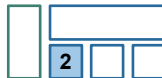
M1-2	Problemraumanalyse	Zuordnung zu den SoS-Gestaltungskonzepten: 1
Zielsetzung Ziel ist die Analyse und Beschreibung der Problemstellung im Kontext von SoS. Hierzu gehören die Anwendungsfälle sowie die an der Zielstellung beteiligten Systeme und Akteure.		Beschreibung ▪ Zunächst wird die Ausgangssituation betrachtet und Stakeholder und deren Bedarfe und Ziele identifiziert. ▪ Auf Basis der Anwendungsfallanalyse werden die Systeme identifiziert, die im Rahmen der Anwendungsfälle benötigt werden. ▪ Unternehmen, die die entsprechenden Systeme managen, werden identifiziert und zugeordnet.
Aktivitäten Problem-orientiert Lösungs-orientiert ▪ Stakeholder identifizieren ▪ Stakeholderbedarfe identifizieren ▪ Ziele beschreiben ▪ Anwendungsfälle identifizieren ▪ Interagierende Systeme ableiten ▪ Unternehmensbeteiligungen festlegen		
Hilfsmittel - ...		
Literatur ▪ Die Methode wurde im Rahmen des Forschungsprojekts MoSyS entwickelt und erprobt. Sie basiert u. a. auf den grundsätzlichen Herangehensweisen etablierter MBSE-Methoden (u.a. OOSEM) und wurde ergänzt um SoS-spezifische Aspekte.		

M1-3	Innovationswerkstatt (MBSE Konzept 1)	Zuordnung zu den SoS-Gestaltungskonzepten: 1
Zielsetzung Ziel ist die Analyse und Beschreibung der Problemstellung. Der Problemraum beschreibt die zu lösenden Herausforderungen und setzt gewisse Rahmenbedingungen, in denen eine Lösung zu finden ist.		Beschreibung ▪ In einem iterativen Vorgehen werden parallel die Anwendungsszenarien identifiziert und das Umfeld analysiert. Das Umfeld eines SoS umfasst gleichzeitig die Systeme, die an einer möglichen Lösung beteiligt sein können. ▪ Hieraus werden die Anforderungen und Funktionen abgeleitet, die den Problemraum definieren
Aktivitäten  ▪ Anwendungsszenarien identifizieren ▪ Umfeld analysieren ▪ Anforderungen ableiten ▪ Funktionen ableiten		
Hilfsmittel - CONSENS Partialmodelle		
Literatur ▪ Tschirner, C.: Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme. Dissertation, Heinz Nixdorf Institut, Universität Paderborn, 2016 ▪ Gausemeier, J.; Lanza, G.; Lindemann, U.: Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung. Carl Hanser Fachbuchverlag, s.l., 2012		

M2-2

Konzipierung von SoS-Konstellationen

Zuordnung zu den
SoS-Gestaltungskonzepten:

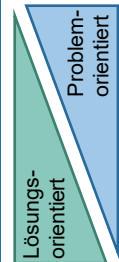


2

Zielsetzung

Ziel ist die Konzeptentwicklung für interagierende Systeme innerhalb einer SoS-Konstellation auf der Meso-Ebene. Dabei wird sowohl die übergeordnete Makro-Ebene als auch die untergeordnete Mikro-Ebene in die Konzeptentwicklung einbezogen.

Aktivitäten



- SoS-Kontextanalyse
- Analyse der emergenten Eigenschaften des SoS
- Analyse der Interaktionsmöglichkeiten und -wahrscheinlichkeiten im SoS
- Dynamikanalyse von SoS-Konstellationen
- Analyse und Auslegung von stabilen SoS-Konstellationen

Beschreibung

- Die Makro-Ebene betrachtet auf abstrakter Ebene u.a. die gesellschaftlichen Rahmenbedingungen von SoS.
- Die Meso-Ebene betrachtet mögliche SoS-Konstellationen und aus welchen Gründen diese zusammenfinden und wieder aufgelöst werden.
- Die Mikro-Ebene fokussiert einzelne SoS-Konstellationen und beschreibt deren Interaktion.

Hilfsmittel

- Phasenmodell



```

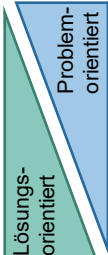
graph TD
    A[Societal context] <--> B[Macro analysis]
    B <--> C[Meso analysis]
    C <--> D[Micro analysis]
    D <--> E[Operational context]
  
```

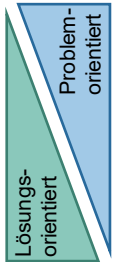
Literatur

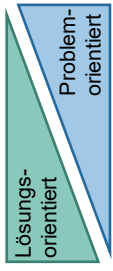
- Axelsson, J.: A Refined Terminology on System-of-Systems Substructure and Constituent System States: 14th Annual Conference System of Systems Engineering (SoSE), 2019, pp. 31–36

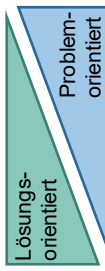


M2-3	Modellbasierte Analyse und Design von SoS	Zuordnung zu den SoS-Gestaltungskonzepten: </
-------------	--	--

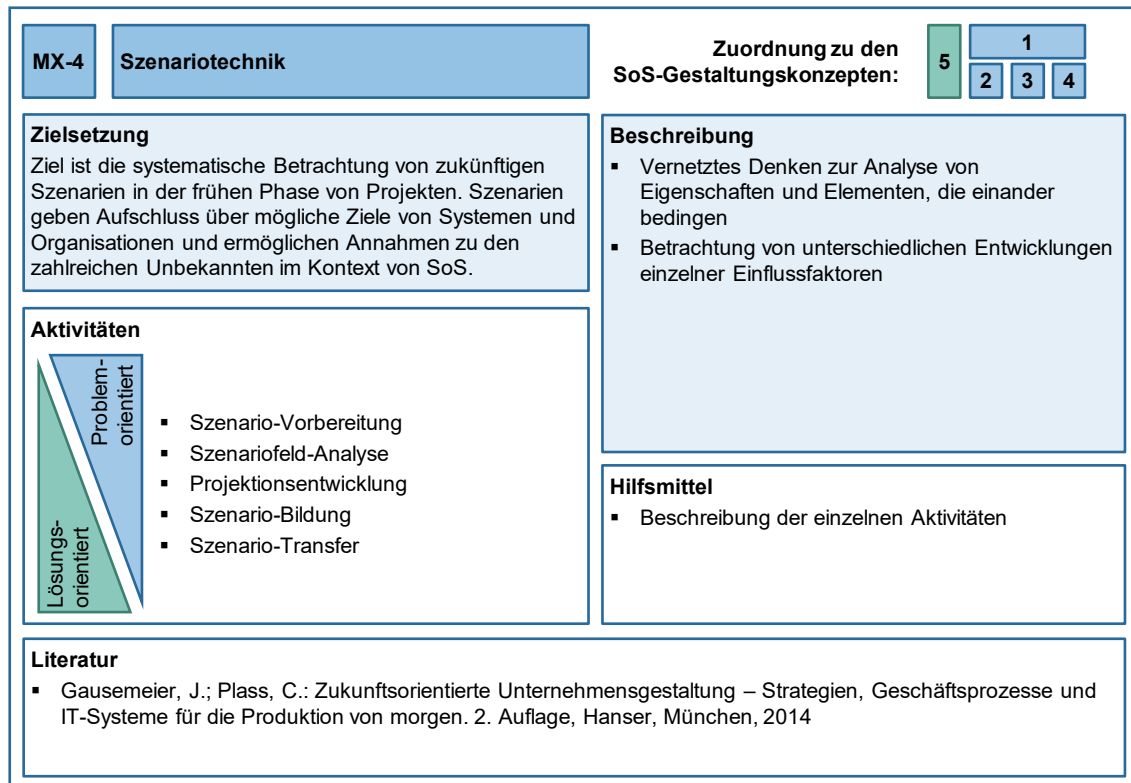
M3-2	Identifikation relevanter Use Cases	Zuordnung zu den SoS-Gestaltungskonzepten: 
Zielsetzung Ziel ist die Reduzierung von einer Vielzahl an möglichen Anwendungsfällen, die bei der Kombination von mehreren Systemen in einem SoS realisiert werden können.		Beschreibung ▪ Die Vielzahl an Fähigkeiten von eigenständigen Systemen führt zu einer unübersichtlichen Anzahl an Möglichkeiten ▪ Die Use Cases werden in einer Matrix gegenübergestellt und hinsichtlich der Kausalität bewertet, um die Edge-Cases zu identifizieren. ▪ Im letzten Schritt werden die Edge-Cases hinsichtlich der Zielstellung priorisiert, um eine fokussierte Anzahl an relevanten Anwendungsfällen zu erhalten
Aktivitäten  ▪ Problemraum analysieren ▪ Mögliche Use Cases auflisten ▪ Use Cases gegenüberstellen und Kausalität bewerten (gering, mittel, stark) ▪ Stark korrelierende Use Cases zusammenführen ▪ Use Cases priorisieren (bspw. auf Basis der Zielstellung)		Hilfsmittel ▪ -
Literatur ▪ Die Methode wurde im Rahmen des Forschungsprojekts MoSyS entwickelt und erprobt. Sie orientiert sich an den grundsätzlichen Herangehensweisen der Extremwertsuche (bspw. nach BOX und BEHNKEN).		

M5-2	OOSEM – Objekt-orientierte Systems Engineering Methode	Zuordnung zu den SoS-Gestaltungskonzepten:	5
Zielsetzung Ziel ist die Entwicklung von Systemen, die sich an den an sie gestellten Anforderungen orientieren. Die Methode beschreibt grundlegende Aktivitäten und ist iterativ für unterschiedliche Hierarchieebenen anwendbar		Beschreibung - Analyse der Stakeholder, die ein Interesse am System haben. - Ableitung von Systemanforderungen, die die Ziele und Bedarfe der Stakeholder adressieren. - Definition der logischen Systeme und Subsysteme, die zur Realisierung der Anforderungen notwendig sind. - Allokation der Anforderungen auf physische Umsetzungen in Form von Hardware, Software, Daten, Personen oder Prozessen.	
Aktivitäten  - Analyse der Stakeholderbedarfe - Analyse von Systemanforderungen - Definition der logischen Architektur - Synthetisieren der physikalischen Architektur		Hilfsmittel Prozessmodell	
Literatur Walden, D. D.; Roedler, G. J.; Forsberg, K. J.; Hamelin, R. D.; Shortell, T. M.: INCOSE Systems Engineering Handbook – A Guide for System life cycle Processes and activities. 4. Auflage, 2015			

MX-1	Zielanalyse	Zuordnung zu den SoS-Gestaltungskonzepten:	5	1		
				2	3	4
Zielsetzung Ziel ist die Analyse und Dokumentation von Zielen eines zu entwickelnden Systems in der frühen Phase der Entwicklung. Dabei wird insbesondere auf die Zusammenhänge und Abhängigkeiten der einzelnen Ziele eingegangen.		Beschreibung - Analyse von Zielen und Zielkonflikten durch Betrachtung von Systemen und deren Stakeholdern - Unterschiedliche Intentionen werden beschrieben und anschließend deren Zusammenhänge und Konfliktbeziehungen identifiziert				
Aktivitäten  - Identifikation von Stakeholdern - Analyse von Intentionen einzelner Akteure - Beschreibung der Ziele, Aufgaben, Ressourcen und Überzeugungen einzelner Systeme und Stakeholder - Identifikation von Zusammenhängen und Konflikten zwischen den Zielen		Hilfsmittel -				
Literatur Böhm, W.; Broy, M.; Klein, C.; Pohl, K.; Rumpe, B.; Schröck, S.: Model-Based Engineering of Collaborative Embedded Systems. Springer International Publishing, Cham, 2021						

MX-2	Vertragsgestaltung	Zuordnung zu den SoS-Gestaltungskonzepten: 5 1 2 3 4
Zielsetzung Ziel ist die Formalisierung von Interaktionen zwischen Partnern. Durch die Definition von Bedingungen und Erwartungen in Form von Verträgen, wird eine erfolgreiche Interaktion zwischen unabhängigen Partnern sichergestellt.		Beschreibung - Es werden grundsätzliche Mechanismen zur Interaktion zwischen Systemen festgelegt. - Verträge unterstützen die dynamische Vernetzung und Entkopplung von Systemen. Sie formalisieren die notwendigen Voraussetzungen und zu übertragenden Informationen für Interaktionen.
Aktivitäten  - Analyse der Anwendungsfälle - Analyse von eigenständigen Systemen (Rollen) und deren Beziehungen zueinander - Ableitung von Vor- und Nachbedingung für die Interaktionen - Definition von Parameter für den Informationsaustausch		Hilfsmittel Sprachelemente: System, Vertrag (inkl. Eigenschaften)
Literatur - Böhm, W.; Broy, M.; Klein, C.; Pohl, K.; Rumpe, B.; Schröck, S.: Model-Based Engineering of Collaborative Embedded Systems. Springer International Publishing, Cham, 2021 - Dridi, C. E.; Benzadri, Z.; Belala, F.: System of Systems Engineering: Meta-Modelling Perspective: IEEE 15th International Conference of System of Systems Engineering (SoSE). 2020, Budapest, 2020, S. 135–144		

MX-3	Systemdynamikanalyse	Zuordnung zu den SoS-Gestaltungskonzepten:	5 1 2 3 4
Zielsetzung Ziel ist die Identifikation von Verhaltensmustern in komplexen Systemen. Die Verhaltensweisen werden auf Basis von Kausalitäten analysiert und festgehalten.		Beschreibung ▪ Kausale Zusammenhänge zwischen unterschiedlichen Einflussgrößen werden in einen Gesamtzusammenhang gebracht ▪ Das analysierte Verhalten ist die Basis, um Einflussfaktoren zu identifizieren und diese ggf. für das Gesamtergebnis der Interaktion beeinflussen zu können.	
Aktivitäten Problem-orientiert Lösungs-orientiert ▪ Variablen aufstellen (Stocks, Flows und aux. Variable) ▪ Kausale Verbindungen zwischen den Variablen aufstellen ▪ Ggf. Feedback Loops aufstellen		Hilfsmittel ▪ Sprachelemente: Stock, Flow und Variable <<Stock>> <<Flow>> <<Variable>>	
Literatur ▪ Ahmadi-Achachlouei, M.; Hilty, L. M.: System Dynamics vs. agent-based modeling - comparing models and approaches: A literature review and a transformation procedure: Journal of Environmental Modelling & Software, 2015			



A2.2 Designprinzipien

Die folgenden Tabellen stellen eine Zusammenfassung dar, in denen generische und spezifischen Designprinzipien aus der SLR sowie den Ansätzen aus dem Stand der Forschung zusammengefasst werden.

Tabelle A-4: Generische Designprinzipien

Nr.	Quelle	Prinzip
1	[SM09]	Verwenden Sie einen Entscheidungsbaum, um festzustellen, welche Designansätze (Bottom-up, Top-Down, Analogie oder interaktive Evolution) am besten funktionieren
2		"Verzicht auf das letzte bisschen Optimierung"
3		Erstellung umfassender mentaler Modelle zum Verständnis des Problembereichs
4		Annehmen, dass ein System komplex ist, bis das Gegenteil bewiesen ist
5		Koordinieren Sie die Menschen
6		Schaffen Sie die richtige Art der Koordination
7		Finden Sie die richtigen Experten
8		Fokussierung der technischen Bemühungen und Ressourcen auf zentrale Aufgaben der Produktentwicklung
9		Verstehen, dass sich Pareto-Diagramme von Potenzgesetzen ableiten
10		Entscheidungen auf Daten stützen

11	[Ada11]	Pareto
12		Erforderliche Sparsamkeit
13		Erforderliche Prägnanz
14		Minimale kritische Spezifikation
15	[Hes12]	Prinzip der Multifinalität
16		Zirkuläre Kausalität
17	[WJ11]	1 (Bescheidenheit)
18		2 (Umfassende Systemperspektive)
19		4 (Transdisziplinäre Integration)
20		5 (POET-Faktor-Betrachtung)
21		6 (Semantischer Konvergenz-Ansatz)
22		7 (Chancen-Risiko-Synergie)
23		8 (Heuristische Entscheidungsunterstützung)
24	[MSB09]	Nutzung von Systemigrammen
25		Paradoxes Denken
26	[Cho16]	Einsatz der richtigen Informationstechnologie
27	[UM15]	Eigenschaften auf Systemebene verbessern
28		Einsatz von Menschen
29		Präventive Korrektur
30		Kombination von Resilienzprinzipien

Tabelle A-5: Spezifische Designprinzipien

Nr.	Quelle	Prinzip	SoS Merkmale							
			1	2	3	4	5	6	7	8
1	[Mai98]	Stabile Zwischenformen				x				
2		Policy Triage				x				
3		Hebelwirkung an den Schnittstellen								x
4		Sicherstellung der Zusammenarbeit	x							
5	[SM09]	Denken Sie eher an die "Entwicklung" des Systems als an die "Gestaltung" des Systems				x				
6		Suchen Sie nach lokalen Aktionen, die globale Auswirkungen haben können.						x		
7		Mehrere Möglichkeiten lebensfähig halten	x	x			x	x	x	x
8		Dem System explizit Vielfalt auferlegen		x				x		x
9		Gestalten Sie in Schichten, die Elemente mit unterschiedlichen Veränderungsgeschwindigkeiten voneinander isolieren				x				

10		Erwägen Sie die Bildung von Schwärmen zur Ausführung bestimmter Aufgaben					x		
11		Erlauben Sie die Verwendung von Modulen außerhalb der normalen Nutzung			x	x	x		x
12		Modellieren Sie bei Bedarf		x			x	x	
13		Analysieren Sie Systemnetzwerke					x	x	x
14		Bereiten Sie sich während der Systementwicklung auf erwartete Designänderungen vor und berücksichtigen Sie diese			x				
15		Verbinden Sie Menschen und Gruppen so weit wie möglich	x						
16		Explizites Management der Entwicklungsumgebung	x		x				
17		Bauen Sie fähige Organisationen auf	x						
18		Beurteilen Sie die tatsächlichen Ergebnisse	x		x		x		
19		Geschäftsregeln für die Interaktion explizit machen	x						
20		Maßnahmen ergreifen, um Katastrophen zu vermeiden	x	x	x		x		x
21	[Jam08b],	Standardisierte Schnittstellen							x
22	[Aza08]	Selbstverwaltung	x		x	x			
23		Konservierung			x			x	
24		Neugestaltung				x			x
25		Symbiose	x						
26		Modularität	x	x	x	x			x
27	[HPC+09]	Bedingung der Emergenz					x		
28		Emergentes Verhalten ist umgekehrt proportional zum Grad der Bindung zwischen Systemen					x		
29		Emergentes Verhalten ist nichtlinear					x		
30		Emergentes Verhalten ist selbstorganisiert					x		
31	[WJ11]	3 (Systemisches Gleichgewichtsprinzip)	x	x					
32		9 (Vertrauenszentrierte Zusammenarbeit)	x						
33		12 (Die Philosophie des einfachen Elementdesigns)		x	x	x		x	x
34		13 (Geschichtete, lose gekoppelte Architektur)		x	x	x		x	x
35	[MSB09]	Integrierte Produktteams	x						
36		Globales Risikomanagement und Entscheidungsanalyse	x		x	x			
37		Schnittstellenmanagement							x

38		Agile Entwicklungskonzepte: Plug-n-Play-Architekturskalierbarkeit					X	X		
39	[Cho16]	Unsicherheiten einbeziehen					X	X		
40	[UM15]	Physische Redundanz				X	X			
41		Funktionale Redundanz				X	X			
42		Reparierbarkeit				X				
43		Übergreifende Systeminteraktion		X					X	
44		Lokalisierte Kapazität					X			
45		Verbesserte Kommunikation auf Organisationsebene		X						
46		Nutzung von Vorfahren				X				
47	[RRR14]	Disruptive architektonische Neugestaltung				X				
48		Nachahmung				X				
49		Sinnvolle Adaptierung				X				
50		Dezentralisierung				X				
51		Gezielte Modularität				X				
52		Integrabilität				X				
53		Rekonfigurierbarkeit				X				
54		Redundanz				X				
55		Skalierbarkeit				X				
56		Freie Kapazität				X				
57		Freiraum				X				

A2.3 Lösungsmuster

Im Folgenden werden die in Kapitel 4.4.2 aufgeführten Lösungsmuster detailliert beschrieben.

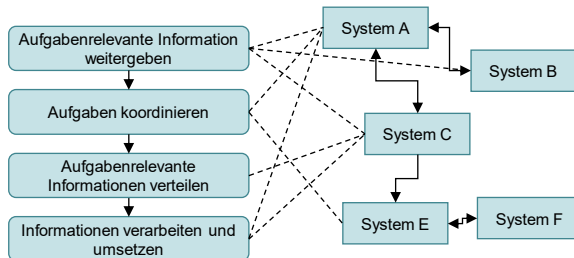
Muster | Dezentrale Architektur

Problembeschreibung

Das Problem besteht in der Koordination der Interaktion von unabhängigen Systemen. Das Ziel ist die gemeinsame Erfüllung einer übergeordneten Aufgabe.

Lösungskonzept

Die Koordination der Interaktion wird dezentral von den beteiligten Systemen realisiert. Dabei werden die für die lokale Lösung von Aufgaben relevanten Informationen herangezogen und im Austausch zwischen den Systemen gelöst. Für die Interaktion müssen die beteiligten Systeme miteinander kommunizieren können.



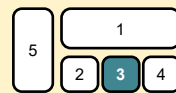
Kontext

Gestaltungskonzeptmerkmale

Managementunabh.
abhängig unabhängig
1 — 2 — 3

Emergentes Verhalten
gering stark
1 — 2 — 3

Interoperabilität
hoch gering
1 — 2 — 3



Weitere Merkmale

Operative Unabhängigkeit
abhängig unabhängig
1 — 2 — 3

Räumliche Verteilung
lokal global
1 — 2 — 3

Evolution
gering stark
1 — 2 — 3

Dynamische Rekonf.
gering stark
1 — 2 — 3

Interdependenz
hoch gering
1 — 2 — 3

Beziehung zu weiteren Mustern

- **Zusammenhängende Muster:** -
- **Alternative Muster:** Zentrale Architektur, Verteilte Architektur

Referenzen

Ingram, C.; Payne, R.; Perry, S.; Holt, J.; et al.: Modelling patterns for systems of systems architectures. 2014

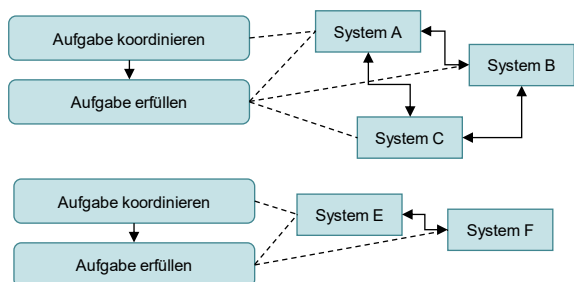
Muster | Verteilte Architektur

Problembeschreibung

Das Problem besteht in der Koordination der Interaktion von unabhängigen Systemen. Das Ziel ist die gemeinsame Erfüllung einer übergeordneten Aufgabe.

Lösungskonzept

Die Koordination der Interaktion wird verteilt in lokalen Netzwerken durchgeführt. Es werden lediglich lokale Optima verfolgt, ohne eine ganzheitlich optimale Lösung abzustimmen. Die Interaktionen erfolgen bedarfsgerecht und dynamisch.



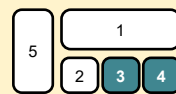
Kontext

Gestaltungskonzeptmerkmale

Managementunabh.
abhängig unabhängig
1 — 2 — 3

Emergentes Verhalten
gering stark
1 — 2 — 3

Interoperabilität
hoch gering
1 — 2 — 3



Weitere Merkmale

Operative Unabhängigkeit
abhängig unabhängig
1 — 2 — 3

Räumliche Verteilung
lokal global
1 — 2 — 3

Evolution
gering stark
1 — 2 — 3

Dynamische Rekonf.
gering stark
1 — 2 — 3

Interdependenz
hoch gering
1 — 2 — 3

Beziehung zu weiteren Mustern

- **Zusammenhängende Muster:** -
- **Alternative Muster:** Zentrale Architektur, Dezentrale Architektur

Referenzen (angelehnt an)

Bröring, A.; Schmid, S.; Schindhelm, C.-K.; Khelil, A.; Kabisch, S.; Kramer, D.; Le Phuoc, D.; Mitic, J.; Anicic, D.; Teniente, E.: Enabling IoT Ecosystems through Platform Interoperability. IEEE Software, (34)1, 2017, S. 54–61

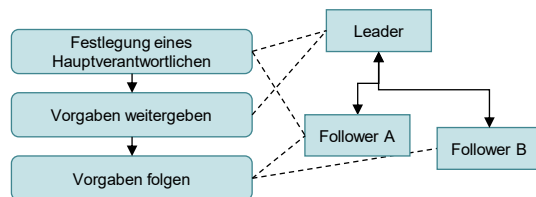
Muster | Leader-Follower Interaktion

Problembeschreibung

Das Problem besteht in der Koordination der Interaktion von unabhängigen Systemen. Das Ziel ist die gemeinsame Erfüllung einer übergeordneten Aufgabe.

Lösungskonzept

Die Koordination der Interaktion wird durch einen Verantwortlichen (Leader) initiiert. Unabhängige Systeme können sich entscheiden, den Vorgaben des Verantwortlichen zu folgen oder einen eigenen Weg zu gehen. Die Verteilung der Verantwortung erfolgt dynamisch und kann durch unterschiedliche Systeme erfolgen.



Kontext

Gestaltungskonzeptmerkmale

Managementunabh.

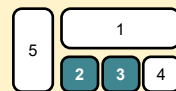
abhängig unabhängig
1 2 3

Emergentes Verhalten

gering stark
1 2 3

Interoperabilität

hoch gering
1 2 3



Weitere Merkmale

Operative Unabhängigkeit

abhängig unabhängig
1 2 3

Räumliche Verteilung

lokal global
1 2 3

Evolution

gering stark
1 2 3

Dynamische Rekonf.

gering stark
1 2 3

Interdependenz

hoch gering
1 2 3

Beziehung zu weiteren Mustern

- **Zusammenhängende Muster:** Zentrale Architektur
- **Alternative Muster:** Dezentrale Architektur, Verteilte Architektur

Referenzen (angelehnt an)

Böhm, W.; Broy, M.; Klein, C.; Pohl, K.; Rumpe, B.; Schröck, S.: Model-Based Engineering of Collaborative Embedded Systems. Springer International Publishing, Cham, 2021

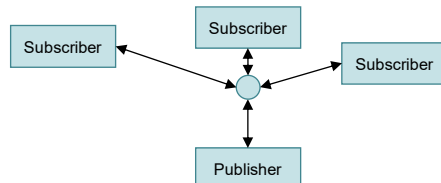
Muster | Publish-Subscribe Interaktion

Problembeschreibung

Das Problem besteht in der Verteilung von Informationen für verteilte Systeme. Um übergeordnete Aufgaben zu lösen, müssen bestimmte Informationen verteilt werden können.

Lösungskonzept

Ein System bietet Informationen an, die von weiteren Systemen abgefragt werden. Der Austausch von Informationen erfolgt durch eine lose Kopplung der Systeme über eine unverbindliche Anfragebeziehung und kann jederzeit gelöst werden.



Kontext

Gestaltungskonzeptmerkmale

Managementunabh.

abhängig unabhängig
1 2 3

Emergentes Verhalten

gering stark
1 2 3

Interoperabilität

hoch gering
1 2 3



Weitere Merkmale

Operative Unabhängigkeit

abhängig unabhängig
1 2 3

Räumliche Verteilung

lokal global
1 2 3

Evolution

gering stark
1 2 3

Dynamische Rekonf.

gering stark
1 2 3

Interdependenz

hoch gering
1 2 3

Beziehung zu weiteren Mustern

- **Zusammenhängende Muster:** -
- **Alternative Muster:** -

Referenzen

Ingram, C.; Payne, R.; Perry, S.; Holt, J.; et al.: Modelling patterns for systems of systems architectures. 2014

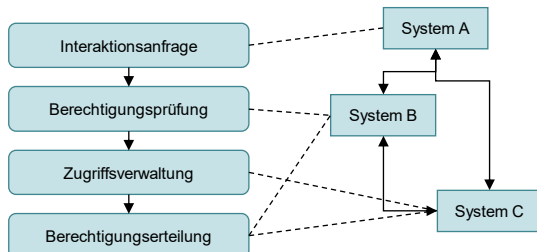
Muster | Autorisierung

Problembeschreibung

Sicherheitsvorgaben, wie bspw. Datenschutz, erfordern Schutzmechanismen, die den Austausch von Informationen oder Objekten autorisieren.

Lösungskonzept

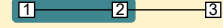
Die Lösung basiert auf einem unabhängigen Drittsystem, welches Zugriffsberechtigungen verwaltet. Im Fall einer Interaktionsanfrage können sich Systeme durch eine Berechtigungsprüfung vorab autorisieren, bevor eine Interaktion stattfindet. Dies verhindert ungewollte oder schädliche Interaktionen.



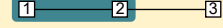
Kontext

Gestaltungskonzeptmerkmale

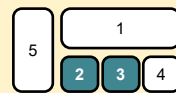
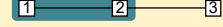
Managementunabh.
abhängig unabhängig



Emergentes Verhalten
gering stark

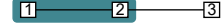


Interoperabilität
hoch gering

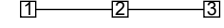


Weitere Merkmale

Operative Unabhängigkeit
abhängig unabhängig



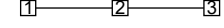
Räumliche Verteilung
lokal global



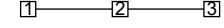
Evolution
gering stark



Dynamische Rekonf.
gering stark



Interdependenz
hoch gering



Beziehung zu weiteren Mustern

- **Zusammenhängende Muster:** Zentrale Architektur, Dezentrale Architektur, Verteilte Architektur
- **Alternative Muster:** -

Referenzen (angelehnt an)

Priebe, T.; Fernandez, E. B.; Mehlaui, J. I.; Pernul, G.: A Pattern System for Access Control. In: Farkas, C.; Samarati, P. (Hrsg.): Research Directions in Data and Applications Security XVIII, 2004, S. 235–249

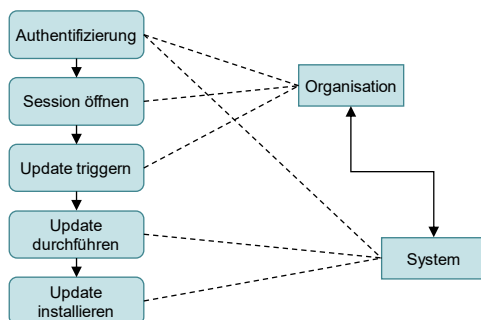
Muster | Over-the-Air-Update

Problembeschreibung

Das Problem besteht in der Anpassung von Systemen an Veränderungen im Umfeld. Dies findet insbesondere in SoS häufig statt.

Lösungskonzept

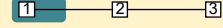
Ein Update-over-the-Air ist eine Maßnahme zur Anpassung und/oder Erweiterungen von Fähigkeiten eines Systems. Hierzu muss eine Fernverbindung zwischen einer Organisation und einem System hergestellt werden können.



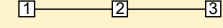
Kontext

Gestaltungskonzeptmerkmale

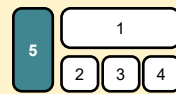
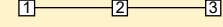
Managementunabh.
abhängig unabhängig



Emergentes Verhalten
gering stark

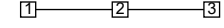


Interoperabilität
hoch gering

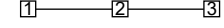


Weitere Merkmale

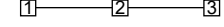
Operative Unabhängigkeit
abhängig unabhängig



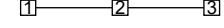
Räumliche Verteilung
lokal global



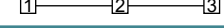
Evolution
gering stark



Dynamische Rekonf.
gering stark



Interdependenz
hoch gering



Beziehung zu weiteren Mustern

- **Zusammenhängende Muster:** -
- **Alternative Muster:** -

Referenzen (angelehnt an)

Bauwens, J.; Ruckebusch, P.; Giannoulis, S.; Moerman, I.; Poorter, E. de: Over-the-Air Software Updates in the Internet of Things: An Overview of Key Principles. IEEE Communications Magazine, (58)2, 2020, S. 35–41

