Implementierung einer komplexitätsreduzierenden

 MBSE-Methode

Anna-Lena Menn1, Anna Nanzig2

1Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 St. Augustin,
anna-lena.menn@h-brs.de
2TH Köln, Steinmüllerallee 1, 51643 Gummersbach, anna.nanzig@th-koeln.de

Track (Industriebeitrag/Wissenschaftlicher Beitrag): *hier den Track angeben*

Keywords:Komplexitätsreduktion, Methode, Systemmodell, Elektromotor

**Zusammenfassung**: In diesem Paper wird eine komplexitätsreduzierende Methode für die systemorientierte Auslegung technischer Systeme vorgestellt und angewendet. Systemisches Denken stärkt und beschleunigt den Entwicklungsprozess. Model Based Systems Engineering (MBSE) stellt ein Systemmodell in den Mittelpunkt der Betrachtungen, welches für eine durchgängige Kommunikation aller Projektbeteiligten sorgt und wichtige Informationen über das technische System bündelt. Für eine erfolgreiche Bearbeitung von Entwicklungsprojekten ist eine anwenderfreundliche Erstellung von Systemmodellen wichtig. Viele bekannte Methoden des MBSE weisen allerdings eine hohe Anwendungskomplexität auf, was einer intuitiven Erstellung eines Systemmodells entgegensteht. Die hier vorgestellte Methode soll diese Komplexität reduzieren, einem weiten Anwenderkreis zugänglich sein, bei gleichzeitigem Erfüllen der technischen Anforderungen. Als Anwendungsbeispiel dient ein elektrischer Antrieb in Form einer permanenterregten Synchronmaschine. An diesem Beispiel wird der zentrale Aspekt der Methode, die Dekomposition eines technischen Systems und der Betrachtung der Schnittstellen, erläutert. Die parasitären Effekte, Wärme und Geräusche, dienen dabei als Bewertungsgegenstand.

# 1 Einleitung

Bekannt ist, dass Systemdenken als Kernkompetenz den Entwicklungsprozess stärkt und beschleunigt. Das Model Based Systems Engineering (MBSE) basiert auf digitalen Systemmodellen. Die Schaffung einer einheitlichen und durchgängigen Kommunikationsebene durch Systemmodelle hilft dabei, den Überblick über komplexe Systeme zu behalten. Die verbesserte Kommunikation zwischen allen Beteiligten des Projektes sorgt für ein gemeinsames Verständnis von Anforderungen und Zusammenhängen. Systemmodelle halten Informationen konsistent und ermöglichen verschiedene Sichten und Abstraktionsebenen auf ein technisches System, durch das Aufzeigen von Schnittstellen gelingt es die Herausforderungen einer komplexen Produktentwicklung zu visualisieren und zu beherrschen [1].

Die Erstellung des Systemmodells erfordert eine Modellierungssprache, eine Modellierungsmethode und ein Softwarewerkzeug, welche aufeinander abgestimmt sein müssen [2,3]. Eine Herausforderung bei der Generierung von Systemmodellen ist die Anwendung dieser Kernelemente, da der Einsatz dieser Elemente eine komplexe Herangehensweise beansprucht, die eine hohe Hemmschwelle in der Generierung von Systemmodellen nach sich ziehen. Dem entgegen steht der Wunsch nach einem breiten Anwenderkreis. Es braucht also komplexitätsreduzierte Methoden und Sprachen, die einem heterogenen Anwenderkreis zugänglich und auch in der Hochschullehre einsetzbar sind. Die funktionale Anwendung des MBSE kann nur durch einen breiten Anwenderkreis durchgesetzt werden. Dafür müssen nicht nur Mitarbeiter:innen, sondern auch die folgende Generationen Entwickler:innen geschult werden.

Ziel ist die Entwicklung einer anwenderfreundlichen Methode, welche trotz simpler Anwendung die Herausforderungen technischer Produktentwicklung widerspiegelt. Ein besonderer Fokus wird bei der Erstellung des Systemmodells auf die Definition der technischen Schnittstellen zwischen Komponenten und auch innerhalb von Bauteilen gelegt, da diese eine Produktentwicklung beeinflussen. Dadurch entsteht ein eindeutiges Vokabular, welches die Kommunikation in der Entwicklung sowie mit anderen Fachbereichen verbessert. Durch die hierarchische Zergliederung der einzelnen Komponenten entsteht eine Systemarchitektur im Sinne des MBSE und ermöglicht unterschiedlich detaillierte Sichten auf das Modell. Als Softwarewerkzeug wird, exemplarisch für blockorientierte Werkzeuge und Sprachen, der Systems Composer der Firma Mathworks® benutzt, da er den Einsatz von niederschwelligen Werkzeugen, Methoden und Sprachen unterstützt, die Softwareprodukte einem breiten Anwenderkreis bereits bekannt sind und die Produktfamilie ermöglicht auch eine 1D Simulation zur Produktauslegung.

# 2 Herausforderungen bei den aktuellen Werkzeugen und Methoden

Viele Methoden sind eine Erweiterung der konventionellen Konstruktionsmethoden. Da die Auffassungen, wie MBSE zu verstehen und umzusetzen ist, in der Realität weit auseinander gehen, gibt es mannigfaltige Weiterentwicklungen dieser Methoden [4]. Im Rahmen der bisherigen Forschungstätigkeiten wurden einige Methoden, inklusive der verwendeten Sprache und Werkzeuge, hinsichtlich ihrer Anwendungsfreundlichkeit untersucht. Für dieses Paper wurden drei Methoden ausgewählt, die eine mittlere, hohe und sehr hohe Komplexität aufweisen.

## 2.1 Drei Methoden

Wie sich Tabelle 1 entnehmen lässt, implementieren alle betrachteten Methoden komplexe Sprachen, die eine hohen Schulungsbedarf fordern. Außerdem wird ein hohes Abstraktionslevel verlangt, welches den direkten Bezug zur Entwicklungsaufgabe erschwert. Generell lässt sich sagen, dass es Zielkonflikte hinsichtlich: effektive Produktentwicklung und Benutzung der Sprachen, Methoden und Werkzeuge gibt. Bei der Komplexitätsuntersuchung wurde das dazugehörige Softwarewerkzeug außen vorgelassen, da das Werkzeug im Prinzip nur die gewählte Sprache implementiert. Basierend auf diesen Ergebnissen wurde eine komplexitätsreduzierte Methode entwickelt, welche sich für den Einsatz in industrieller Produktentwicklung und der Lehre eignet.

Tabelle 1. Übersicht über Anwenderfreundlichkeit bestehender Modellierungsmethoden

|  |  |  |  |
| --- | --- | --- | --- |
| **Methode:** | **MSE Architecture [5, 6]** | **CONSENS****Methode****[3]** | **SYSMOD****[7]** |
| Ursprung | RWTH | IEM Paderborn | Tim Weilkiens |
| Modellierungssprache: |
| Sprache | SysML | CONSENS | SysML |
| Anpassbarkeit des Sprachprofil | wenig flexibel | flexibel | Sehr flexibel |
| Sprach Umfang | hoch | niedrig | hoch |
| Komplexität der Sprache  | hoch | mittel | sehr hoch |
| Modellierungsmethode: |
| Systemmodell | funktional | funktional | objektorientiert |
| Anpassungsfähigkeit | mittel | gut | gut |
| Abstraktionslevel | hoch | mittel | sehr hoch |
| Konkreter Bezug zur Umsetzung | hoch | mittel | niedrig |
| Komplexität | hoch | mittel | mittel |
| Ziel | Neuschöpfung von Produkten | Neuschöpfung von Produkten | Neuschöpfung von Produkten |

# 3 Vorstellung der komplexitätsreduzierenden Methode

Bild 1 zeigt eine Übersicht über das methodische Vorgehen bei der Erstellung des Systemmodells, diese gliedert sich in drei Schritte: 1) die abstrakte Systemmodellierung, 2) die Festlegung der Rahmenbedingungen für die Modellierung und 3) die tatsächliche Erstellung des Systemmodells.



Bild 1. Schematischer Aufbau der Methode

Für die erste Erstellung eines Systemmodells sind alle drei Schritte notwendig. Für eine Erweiterung oder eine tiefergehende Spezifizierung des Systems muss, je nach Anwendung, nur der letzte Schritt iteriert werden.

Bei der abstrakten Systemmodellierung wird auf Basis der Gesamtfunktion des Systems eine lösungsneutrale, funktionale Analyse des Systems durchgeführt. So können bestehende Lösungen besser regelmäßig hinterfragt werden. Den Teilfunktionen, die bei der Zergliederung der Hauptfunktionalität entstehen, werden technische Lösungsprinzipien zugeordnet. Die Methode bedient sich der Bondgraphtheorie. Diese basiert auf der Annahme, dass ein technisches System aus Subsystemen, Komponenten und Bauelementen besteht und Energie ausgetauscht wird [8]. Diese Annahme führt zu einem erzwungenen Aufzeigen von Verbindungen zwischen den verschiedenen Komponenten.

Die Festlegung der Rahmenbedingungen beinhaltet zum einen die Festlegung einer sinnvollen und erweiterbaren Nomenklatur für das Systemmodell, zum anderen die Entwicklung von Sprachprofilen, um die allgemeine Sprache des System Composer möglichst auf die spezifische Entwicklung des Systems auslegen zu können.

Der letzte Schritt spezifiziert das methodische Vorgehen zur konkreten Modellierung eines Systemmodells. Zunächst wird das Hauptsystem, in diesem Fall der elektrische Antriebsstrang, modelliert. Dazu werden die Hauptkomponenten hinsichtlich ihrer Struktur und Hauptumsätze analysiert. Unter Berücksichtigung der Regelungsstrukturen des Systems wird eine Architektur der Hauptkomponenten erstellt. Durch die Dekomposition der Komponenten werden unerwünschte Funktionen und parasitäre Effekte untersucht. Schlussendlich wird die Architektur der Hauptkomponenten in den Zusammenhang mit seiner Umgebung gesetzt und aussagekräftiges Systemmodell entsteht. In diesem Schritt geht es vor allem um die Schnittellendefinition, welchen den Kern der Methode darstellt und die Kommunikation verbessern soll. Die Methode legt einen besonderen Fokus auf das Aufzeigen von Schnittstellen. Die Betrachtung der Schnittstellen wird durch die Dekomposition eines technischen Systems erzwungen.

Das Aufzeigen von Verbindungen zwischen Komponenten, die Schnittstellendefinition und die Dekomposition wirken darauf ein, dass es mit der Methode möglich ist, ein Systemmodell zu erstellen, welches das Potential bietet mit bewährten Werkzeugen 1D Simulationen anzustoßen.

# 4 Anwendungsbeispiel elektrischer Antriebsstrang

Als Anwendungsbeispiel dient die Entwicklung eines batterieelektrisch angetriebenen Fahrzeuges, da dies einen hohen industriellen Anwendungsbezug und eine hohe Marktdurchdringung hat. Hierbei soll der prinzipielle Aufbau und die Umsetzung eines elektrischen Antriebsstranges als bekannt vorausgesetzt werden, die abstrakte Systemmodellierung mit der Zuordnung der Lösungsprinzipien und den Leistungsverbindungen (erster Teil der Methode) ist bereits erfolgt (siehe Bild 2).



Bild 2. Schematischer Aufbau elektrischer Antriebsstrang

Exemplarisch soll die Erstellung der detaillierten Systemarchitektur mittels Dekomposition (dritter Teil der Methode) einer PMSM (permanenterregten Synchronmaschine) durchgeführt werden. Zentraler Bewertungsgegenstand sind in diesem Beispiel die unerwünschten Effekte, die nicht in direktem Zusammenhang mit der Funktion des technischen Systems stehen, aber für eine sachgemäße Auslegung und späteren Betrieb zwingend berücksichtigt werden müssen [9]. Durch die Dekomposition sollen die Ursachen und Wirkungen parasitärer Effekte sichtbar werden. In der abstrakten Systemmodellierung wird der elektrische Motor auf oberster funktionaler Ebene, ohne Dekomposition, als elektromechanischer Energiewandler betrachtet (siehe Bild 2).

Im Folgenden wird die Dekomposition der PMSM (siehe Bild 3) durchgeführt, hierbei wird die Regelung vernachlässigt. Neben der Darstellung der Leistungsverbindungen aus Elektrik und Mechanik werden hier nun auch die Komponenten und deren Verbindung untereinander sichtbar.



Bild 3. Dekomposition des elektrischen Motors

Die Synchronmaschine besteht aus der elektrischen Komponente Stator und der mechanischen Komponente Rotor. Dazwischen befindet sich der Luftspalt. Im Luftspalt entsteht aufgrund des Rotordrehfeldes und dem Statorstrombelag eine tangentiale Kraft am Rotor, die diesen in Bewegung versetzt. Außerdem dient der Luftspalt zur mechanischen Trennung von Stator und Rotor. Da der Luftspalt mehrere Funktionen hat, wird ihm keine Domäne zugewiesen. Damit ist die Darstellung der gewünschten Funktion abgeschlossen.
Durch die Erweiterung der Dekomposition um die parasitären Effekte Geräusche und Wärme werden die Schnittstellen zwischen den Bauteilen und den Auswirkungen sichtbar, bspw. die elastische Verformung des Stators oder die Eisenverluste. Entwicklerinnen sind somit gezwungen sich über die Entstehung der Effekte Gedanken zu machen und diese in der Produktentwicklung zu berücksichtigen, indem präventive Maßnahmen zur Reduzierung und Beherrschung dieser Effekte ergriffen werden können, an dieser Stelle sei die Auslegung des Kühlsystems erwähnt. Für die eingehende Untersuchung solcher parasitären Effekte sind fundierte Fachkenntnisse nötig, daher wird die Dekomposition in Zusammenarbeit mit Experten der jeweiligen Fachbereiche durchgeführt. Doch auch für den Laien werden die Zusammenhänge, aus denen die parasitären Effekte entstehen sichtbar [10].

Nun ist es möglich die parasitären Effekte auch auf höherer funktionaler Ebene (siehe Bild 4) sichtbar zu machen, ohne die Ursachen und Wirkung hinterfragen zu müssen. Es ist jedoch sichtbar, dass es offenliegende Schnittstellen gibt, die in den Gesamtzusammenhang eingebettet werden müssen. Diese Ebene unterstützt beispielsweise die Versuchsplanung bei der Erstellung von Testszenarien.



Bild 4. höhere funktionale Ebene, die parasitäre Effekte sichtbar macht

# 5 Forschungsbedarf und Ausblick

In weiteren Forschungsvorhaben ist die Anwendung und Optimierung der komplexitätsreduzierten Methode Gegenstand der Untersuchungen. Neben den technischen Anforderungen, die an Systeme gestellt werden, ist die Erfüllung der Anforderungen in Bezug auf Energieeffizienz und Ressourcenschonung immer wichtiger. Vor diesem Hintergrund wird die Methode entsprechend mit einer Nachhaltigkeitsbewertung erweitert. Wie die physikalischen Verbindungen können auch Ressourcen sichtbar gemacht werden, die Verknüpfung von physikalischem Verhalten und dem Verbrauch von sensiblem Material soll hier durch Einbettung in die abstrakte Systemmodellierung geschehen.

Bei der Auslegung elektrischer Antriebe für die Luftfahrt kommen den Anforderungsaspekten „Funktionale Sicherheit“ und „Lebensdauer“ eine besondere Bedeutung zu. Diese sollen ebenfalls in der Methode verankert werden.

# Literaturverzeichnis

[1] J. Lu, D.-J. Chen, D. Gürdür und M. Törngren, „An Investigation of Functionalities of Future Tool-chain for Aerospace Industry“, INCOSE International Symposium, Jg. 27, Nr. 1, S. 1408–1422, 2017, doi: 10.1002/j.2334-5837.2017.00437.x.

[2] D. D. Walden, Hg., Systems engineering handbook: A guide for system life cycle processes and activities. Hoboken, NJ: Wiley, 2023. [Online]. Verfügbar unter: https://learning.oreilly.com/library/view/-/9781119814290/?ar

[3] Lydia Kaiser, „Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme“, Universität Paderborn, Paderborn, 2013. [Online]. Verfügbar unter: http://digital.ub.uni-paderborn.de/urn/urn:nbn:de:hbz:466:2-12873

[4] Krause, Hg., Design Methodology for Future Products: Data driven, agile and flexible, 1. Aufl. [S.l.]: Springer International Publishing, 2022.

[5] D. Krause und E. Heyden, Hg., Design methodology for future products: Data driven, agile and flexible. Cham: Springer, 2022.

[6] T. Zerwas et al., „Mechanical concept development using principle solution models“, IOP Conf. Ser.: Mater. Sci. Eng., Jg. 1097, Nr. 1, S. 12001, 2021, doi: 10.1088/1757-899X/1097/1/012001.

[7] T. Weilkiens, Systems Engineering mit SysML: Anforderungen, Analyse, Architektur, 3. Aufl. Heidelberg, München: dpunkt.verlag, 2014. [Online]. Verfügbar unter: http://ebooks.ciando.com/book/index.cfm/bok\_id/2250671

[8] W. Roddeck, Grundprinzipien der Mechatronik: Modellbildung und Simulation mit Bondgraphen, 2. Aufl. Wiesbaden: Springer Vieweg; divibib GmbH, 2017.

[9] C. Fräger und W. Amrhein, Kleinmotoren, Leistungselektronik, 5. Aufl. Berlin: De Gruyter Oldenbourg, 2020.

[10] H. Tschöke, P. Gutzmer und T. Pfund, Hg., Elektrifizierung des Antriebsstrangs: Grundlagen - vom Mikro-Hybrid zum vollelektrischen Antrieb, 1. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg; Springer Vieweg, 2019. [Online]. Verfügbar unter: https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5977065