[GOR+23] C. Granrath, P. Orth, B. Rumpe, L. Wachtmeister:
Optimierungspotentiale zur effizienten Systemmodellierung im Kontext der Automobilen Systementwicklung.
In: Tag des Systems Engineering 2023, W. Koch, D. Wilke, S. Dreiseitel, R. Kaffenberger (Eds.),
pp. 307-313, Gesellschaft für Systems Engineering (GfSE) e.V., Nov. 2023.

Optimierungspotentiale zur effizienten Systemmodellierung im Kontext der Automobilen Systementwicklung

Christian Granrath[1], Philipp Orth[1], Bernhard Rumpe [2], Louis Wachtmeister[2]

[1]FEV.io GmbH, Technology Systems, Neuenhofstraße 181, 52078 Aachen, {nachname}@fev.io
[2]RWTH Aachen University, Chair of Software Engineering, Ahornstraße 55, 52074 Aachen, {nachname}@se-rwth.de

Keywords: *Automotive, Systems Engineering, Best Practices, CUBE*

Zusammenfassung: Die Automobilindustrie steht vor der Herausforderung, immer komplizierte Systeme mit einer wachsenden Anzahl neuer Funktionalitäten unter hohem Zeitdruck und Kostendruck in einem wettbewerbsintensiven Umfeld zu entwickeln. Die Bewältigung dieser Herausforderung erfordert eine ganzheitliche und systematische Herangehensweise in der Systementwicklung und somit einen Wandel hin zu einem ausgeprägten ganzheitlichen Systemdenken. Um diese Ziele zu erreichen, benötigt die Industrie effiziente Prozesse und Werkzeuge. Obwohl es bereits eine Vielzahl von Methoden für die Entwicklung von Systemen gibt, existieren noch Optimierungspotenziale bei der Entwicklung von automobilen Systemen. Dies betrifft insbesondere die effiziente Modellierung von Systemen. Um diese Potenziale optimal auszuschöpfen, zeigt dieser Beitrag auf, wie das Systemdenken genutzt werden kann, um innerhalb kurzer Zeit und unter begrenzten Ressourcen Systemmodelle zu erstellen. Dazu werden die zu modellierenden Entwicklungsartefakte betrachtet und Ansätze zur effizienten Modellierung dieser Artefakte analysiert, aus denen Empfehlungen auf Basis der in der industriellen Anwendung gewonnenen Erkenntnisse abgeleitet werden.

1 Einleitung

Systems Engineering (SE) verkörpert einen Paradigmenwechsel weg von der komponentengetriebenen Entwicklung hin zu einer nachhaltigen, kundenorientierten Entwicklung durch Systemdenken und Feature-Orientierung. Das Ziel besteht darin, in kürzester Zeit und mit knappen Ressourcen ein maßgeschneidertes Produkt für den Endkunden zu entwickeln. Einerseits ermöglicht das Systemdenken, komplexe Systeme als Ganzes zu verstehen, anstatt sich ausschließlich auf einzelne Teile oder Aspekte zu konzentrieren und dabei die Wechselwirkungen zu übersehen. Andererseits erleichtert die Feature-Orientierung das Aufteilen des Gesamtsystems in einzelne funktionale Bestandteile, die als sogenannte Features bezeichnet werden [2]. Besonders in der Automobilindustrie spielt das Systemdenken eine herausragende Rolle, um die komplexen Herausforderungen im Zusammenhang mit der zunehmenden Vernetzung von Fahrzeugen und ihren Komponenten zu erfassen und in sich kontinuierlich verkürzten Entwicklungszyklen zu bewältigen. Daher geht es nicht mehr nur um die Entwicklung einzelner Komponenten eines Fahrzeugs wie Motor, Fahrwerk und

Steuergeräte, sondern auch darum, die Wechselwirkungen zwischen diesen Systemen in verschiedenen Fahrzeugvarianten zu berücksichtigen. Um die Potenziale des Systemdenkens in der Automobilindustrie besser auszuschöpfen, beschäftigt sich dieser Beitrag damit, wie Systemdenken in optimierten Entwicklungsmethoden für die Entwicklung von Automobil-Systemen eingesetzt werden kann, um in kurzer Zeit und mit begrenzten Ressourcen wiederverwendbare Systemmodelle zu erstellen. In diesem Beitrag werden zu diesem Zweck folgende Fragen näher beleuchtet:

1. **F1:** Welche Entwicklungsartefakte müssen bei der Entwicklung automobiler Systeme erstellt werden?
2. **F2:** Wie können diese Entwicklungsartefakte effizient modelliert werden?
3. **F3**: Können aus den durch industrielle Anwendung der Konzepte gewonnenen Erkenntnissen Empfehlungen abgeleitet werden?

Dazu präsentieren wir ein Prozess-Struktur-Modell für die Entwicklung von Automobilsystemen, das wir im Rahmen des CUBE (Compositional Unified system-Based Engineering) Systems Design Prozesses [2] entwickelt haben. Anhand dieser Analyse zeigen wir die Zusammenhänge zwischen den Artefakten, die für die Automatisierung und zeitliche Optimierung des Systemmodellierungsprozesses relevant sind. Anschließend teilen wir unsere Erfahrungen, die wir aus den ersten Anwendungen dieser Ansätze gewonnen haben.

2 Prozess-Struktur-Modell des Automobilen Systems Engineerings

Im Software- und Systems Engineering ist die Analyse von Prozessen und Strukturen, insbesondere die erstellten Artefakte, ein wichtiges Instrument, um eine modellgetriebene und ganzheitliche Entwicklung zu ermöglichen [2]. Beispielsweise kann ein Modell der im Prozess erstellten Artefakte verwendet werden, um Abhängigkeiten und Zusammenhänge der einzelnen Systemkomponenten zu ihren Anforderungen zu visualisieren [3]. Im Rahmen dieses Beitrags wird in einem Prozess-Struktur-Modell die System Design Artefakte der CUBE-Methodik [1] präsentiert, um eine modellgetriebene Systementwicklung [2] zu ermöglichen. Die CUBE-Methodik kombiniert konventionelle Software-Engineering-Grundlagen mit dem agilen Entwicklungsansatz der feature-getriebenen Entwicklung. Im System Design werden zunächst die Bedarfe der Stakeholder des Systems in der „Stakeholder Value Specification (SVL)“ spezifiziert. Anschließend werden diese auf Grundlage ihrer Use Cases zu Features in einer "Feature List (FL)" zusammengefasst und für jedes Feature wird ein Wirkprinzip in Form einer "Operating Principle Specification (OPS)" weiter verfeinert. Durch die Zuordnung der Aktionen zu ausführenden logischen Komponenten ergibt sich die "Logical Architecture Specification (LAS)". Diese wird schließlich in der Produktarchitektur technisch umgesetzt, die in der CUBE-Methodik als "Product Architecture Specification (PAS)" spezifiziert ist. Basierend auf diesen Artefakten für das System Design wurden im nächsten Schritt die Zusammenhänge der Artefakte im Kontext der Architekturbeschreibung definiert. Obwohl es mit der Norm ISO/IEC/IEEE 42010 [5] einen industrieunabhängigen Standard für die Architekturbeschreibung gibt, konnte in der Automobilindustrie bisher kein einheitlicher Ansatz zur Modellierung von Systemarchitekturen etabliert werden. Daher sind Architekturbeschreibungen in der Automobilindustrie oft projekt- und unternehmensspezifisch, was einen Vergleich erschwert.

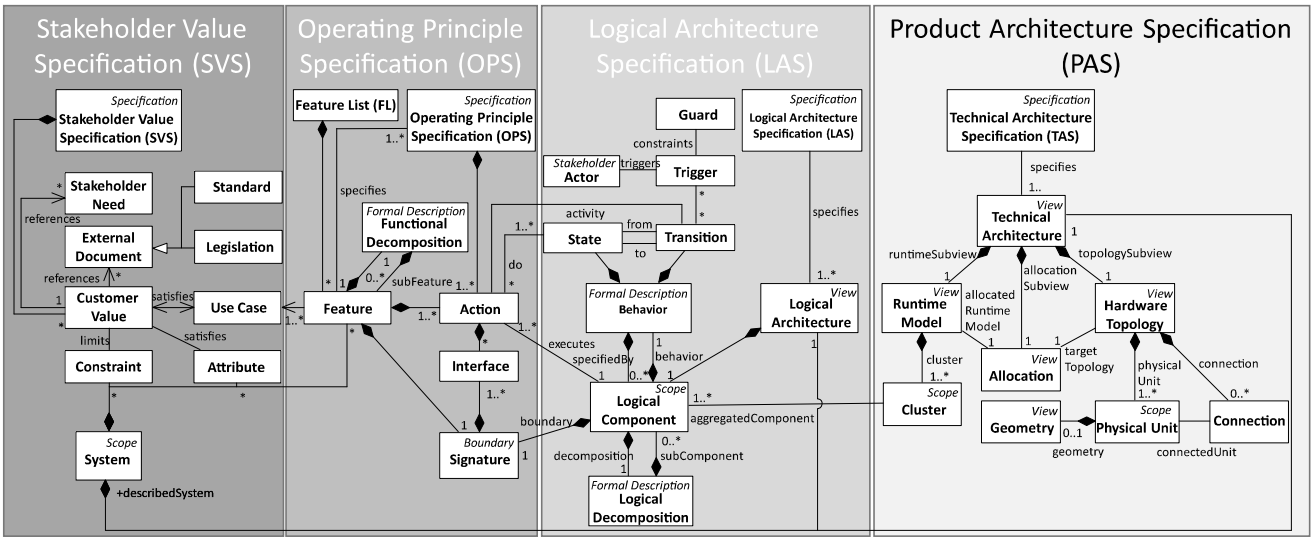

Abbildung 2 Prozess-Struktur-Modell des Systems Engineerings im Kontext der CUBE-Methodik

Das Automotive Architecture Framework (AAF) [4] reicht am nächsten an einen akzeptierten Standard heran. Es beschreibt ein generelles Architekturframework für die Automobilindustrie gemäß [5]. Das AAF muss jedoch noch an die jeweiligen unternehmens- und projektspezifischen Anforderungen angepasst werden. Daher wird in Abbildung 2 eine spezifische Anpassung als Prozess-Struktur-Modell für die zuvor beschriebene CUBE-Methodik präsentiert.

Beispielweise wird die Bezeichnung „User Function“ durch „Feature“ ersetzt, da nicht alle Funktionen eines Fahrzeugs unmittelbar dem Nutzen des Systemnutzers dienen. Im Automobilbereich sind viele elementare und gesetzgebungsrelevante Funktionen auch für Nicht-Benutzer relevant, wie beispielsweise in Bezug auf Lärm, Umweltauswirkungen oder Unfallbeteiligungen. Zusätzlich wird eine zusätzliche Ebene für die Bedarfe der Stakeholder eingeführt, um deren Anforderungen besser zu berücksichtigen. Aus diesem Grund haben wir bewusst das Konzept von CUBE aufgegriffen und eine Ebene für Stakeholderbedarfe eingeführt, auf die sich die Systemfeatures beziehen.

In diesem Kontext werden die Features sowohl in der funktionalen als auch in der logischen Architektur ergänzt. Dies ermöglicht eine Ableitung von einer Perspektive zur anderen als Optimierungspotenzial. Schließlich wird die technische Architektur um „Physical Units“ erweitert, um die physische Umsetzung der Komponenten zu berücksichtigen. Diese Anpassungen und Ergänzungen des Frameworks tragen dazu bei, eine umfassendere und spezifischere Architektur gemäß der CUBE-Methodik zu entwickeln.

3 Zusammenhänge und Optimierungspotentiale

Wie in **Abbildung 2 Prozess-Struktur-Modell des Systems Engineerings im Kontext der** CUBE-Methodik dargestellt, werden bei der Entwicklung automobiler Systeme verschiedene Artefakte erstellt, die durch eine geeignete Modellierung der Elemente Optimierungsmöglichkeiten bieten. Als Grundvoraussetzung sollte zunächst eine modellbasierte

Entwicklungsmethode eingesetzt werden, bei der Systemmodelle in einer maschinell interpretierbaren Syntax und mit klaren definierten Semantiken erstellt werden. Dies ermöglicht zusätzliche Automatisierung in der Entwicklung, da beispielsweise Modelltransformationen zwischen verbundenen Artefakten ermöglicht werden, Konsistenzchecks automatisiert werden können und Spezifikationen bereits in der Modellform frühzeitig testbar sind **(OP1)**. Dadurch kann das Systemmodell als zentrales Artefakt im Rahmen einer modellgetriebenen Entwicklung verwendet werden [6].Wenn eine funktionsorientierte Modellierung [7] mit dem Ansatz der agilen, feature-getriebenen Entwicklung kombiniert wird, lässt sich der Entwicklungsprozess ebenfalls effizienter gestalten [1]. Durch lösungsneutrale Funktionsspezifikationen, die als Grundlage für die Systemmodellierung dienen, können diese später im Entwicklungsprozess wiederverwendet werden. Durch die Verwendung von Modellen auf höheren Abstraktionsebenen können automatisch Modelle oder Modellteile der nächsthöheren Abstraktionsebene generiert und miteinander verknüpft werden **(OP2)**. Dadurch entstehen Zusammenhänge zwischen Use-Case-Szenarien, Wirkketten, (logischen) Architekturen und technischen Architekturen, die für weitere Automatisierung genutzt werden können. Zum Beispiel können Use-Case-Spezifikationstemplates verwendet werden, um Wirkketten abzuleiten oder die logische Architektur zu gestalten [7].Um die Effizienz der Modellierer weiter zu erhöhen, ist es ebenso wichtig, die Modelle und ihre Elemente wiederzuverwenden **(OP3)**. Da sich beispielsweise Fahrzeuge in vielen Grundfunktionen und Use Cases kaum unterscheiden, können solche Beschreibungen auch in anderen Fahrzeug- oder Systementwicklungsprojekten wiederverwendet werden. Dies reduziert den Aufwand bei der Systemmodellierung, wenn bereits erstellte Modelle und Modellartefakte wiederverwendet werden können. Zudem kann durch geeignetes Variantenmanagement eine erhebliche Wiederverwendbarkeit erreicht werden. Varianten der Modelle können als Modellartefakte betrachtet und entsprechend verwaltet werden.

Schließlich besteht ein weiterer Optimierungsansatz darin, die Modelle kontinuierlich zu verfeinern. Durch schrittweise Verfeinerung und Erweiterung der Modelle können detailliertere Informationen und spezifischere Darstellungen integriert werden. Dadurch können zu detaillierte oder zu abstrakte Beschreibungen vermieden werden [7]. Zusammenfassend ist es wichtig, dass die Erstellung des Systemmodells nicht nur einen Selbstzweck erfüllt, sondern auch einen Mehrwert in der Entwicklung bietet. Da die Modellierung selbst jedoch auch Arbeitsaufwand erfordert, ist es wichtig, verschiedene Ansätze zur Steigerung der Effizienz zu kombinieren. Dadurch kann verhindert werden, dass innerhalb eines Systementwicklungsprojekts oder über mehrere Projekte hinweg redundante Tätigkeiten ausgeführt werden. Durch die Anwendung dieser Optimierungspotenziale kann die Effizienz der Systemmodellierung erheblich gesteigert werden, was zu einer verbesserten Entwicklung im Kontext der automobilen Systementwicklung führt und zu höherer Effizienz und niedrigeren Kosten beiträgt.

4 Erfahrungen aus der Anwendung im industriellen Umfeld

Durch die Anwendung der in Kapitel 2 vorgestellten Prozesse und Artefakte unter Verwendung der Optimierungspotentiale aus Kapitel 3 konnten wir durch die industrielle Anwendung die folgenden Lektionen ableiten:

Lektion 1 (Systemdenken als Mindset): Systemdenken ist ein Mindset, das implizites Wissen in explizites Wissen überführen soll. Obwohl das Systemdenken die Möglichkeit bietet, implizites Wissen in expliziter Form darzulegen, haben wir gelernt, dass dieses Denken nicht zwangsläufig in jeder Organisation und jedem Projekt praktiziert wird, sondern aktiv gefordert und gefördert werden muss. Es stellt eine grundlegende Voraussetzung für den Erfolg eines Entwicklungsprojekts dar, weshalb das Etablieren dieses notwendigen Mindsets im gesamten Entwicklungsteam ein Hauptziel zu Projektbeginn sein muss.

Lektion 2 (Analytisches Denken beim Modellieren): Modellieren erfordert analytisches Denken sowie die Fähigkeit, komplexe Informationen zu strukturieren und abstrakte Konzepte in visueller Form darzustellen. Es geht nicht allein darum, vorhandene Informationen zu visualisieren, sondern auch darum, ein Systemmodell zu erstellen, das sowohl die aktuelle Projektsituation abbildet als auch generische Spezifikationsanteile für künftige Unternehmensentwicklungen einschließt. Wir haben erkannt, dass dafür eine einheitliche Sprache und ein Verständnis der Prozesse im Systems Engineering notwendig sind. Um diese Prozesse aktiv anwenden und steuern zu können, ist es wesentlich, dass die zu modellierenden Artefakte frühzeitig explizit definiert werden. Dies verbessert das Verständnis der Zusammenhänge zwischen den Artefakten und ermöglicht dem Projektmanagement eine strukturierte Planung und Verfolgung der Systems Engineering-Aktivitäten basierend auf den Artefakten. Als Orientierung für die Systementwicklung in der Automobilindustrie im CUBE [1] dient z.B. das Prozess-Struktur-Modell aus Abbildung 1.

Lektion 3 (Unterschied Syntax und Semantik): Das Systemmodell ist eine abstrakte Repräsentation des Systems, die einen definierten Zweck in der Systementwicklung verfolgt. Jedoch wird eine Systemmodellierungssprache nicht nur durch ihre Syntax („was wir sehen“) definiert, sondern vor allem durch ihre Semantik („was es bedeutet“) [8]. Wenn die Beteiligten eines Systementwicklungsprojekts diese Konzepte nicht klar unterscheiden, sind nicht nur präzise Kommunikation unmöglich, sondern auch eine Vielzahl von Missverständnissen vorprogrammiert. Es ist daher wesentlich, ein Verständnis für den Unterschied zwischen Syntax und Semantik zu verdeutlichen und dies unternehmensweit zu kommunizieren, um die in OP1 identifizierten Potenziale überhaupt nutzen zu können. Um jedoch von den Potenzialen der Modellierungssprache zu profitieren, müssen die Ergebnisse der Modellierung für das Systementwicklungsprojekt nutzbar sein, denn „wer malt, modelliert nicht“! Obwohl allein durch die Darstellung des Systems auf Papier ein Erkenntnisgewinn erzielt werden kann, erzeugen gemalte Diagramme, angewendet auf ein großes Automobilsystem, Zusatzaufwände in der Diagrammerstellung und Konsistenzhaltung. Dies lässt die Potenziale des Systemmodells sowie die Möglichkeiten zur Aufwandsreduktion aus OP1 & OP2 ungenutzt.

Lektion 4 (Grad der Formalisierung): Obwohl ein hoher Formalisierungsgrad der Modellierungssprache vor allem in der Verhaltensbeschreibung von Vorteil sein kann, indem er Ausführbarkeit und Möglichkeiten zur mathematischen Verifikation ermöglicht, kann er jedoch in heterogenen Entwicklungsteams, in denen Teammitglieder aus verschiedenen Domänen und mit unterschiedlichen Kenntnissen zusammenarbeiten auch Nachteile mit sich bringen. Der Grad der Formalisierung einer Systemmodellierungssprache und eines Systemmodells sollte daher an den Anwendungsbereich und die Anwender angepasst werden. Dabei sollte bei der Entscheidung über den Formalisierungsgrad das Verhältnis zwischen Nutzen, Kosten und Auswirkungen auf die Akzeptanz im Entwicklungsprojekt abgewogen werden.

Zusammenfassend sollte berücksichtigt werden, dass wenn etwas effizient automatisiert werden kann, dieses Potenzial genutzt werden sollte, um Fehler zu vermeiden, Zeit zu sparen und die Qualität zu erhöhen.

Lektion 5 (Wiederverwendung von Modellen): Die Wiederverwendung von Modellen über verschiedene Systementwicklungsprojekte hinweg, wie in OP3 als Optimierungspotenzial identifiziert wurde, erfordert eine klare Trennung zwischen einer funktionsorientierten/lösungsneutralen Sicht und einer produktorientierten/lösungsspe-zifischen Sicht. Da sich in der Automobilindustrie die Funktionen bei unterschiedlichen Produkten in vielen Fällen kaum unterscheiden, können diese auch bei verschiedenen Produktarchitekturen wiederverwendet werden. Funktionen wie Transportfunktionen, Beleuchtung oder Medienwiedergabe sind weitestgehend unabhängig von technischen Unterschieden wie Fahrzeugtyp oder Antriebstechnologie. Um sicherzustellen, dass dieselben (logischen) Systeme in beiden Projekten vorhanden sind, ermöglicht eine projektunabhängige logische Referenzarchitektur die projektübergreifende Wiederverwendung. Zusätzlich ist es wichtig, Modellelemente, wie in OP2 beschrieben, zu generieren und zu prüfen. Bei der Entwicklung einer lösungsneutralen Architektur sollte die Grad der Lösungsneutralität, ähnlich wie beim Grad der Formalisierung, abgewogen werden, um zu bestimmen, ob der Nutzen durch eine mögliche Wiederverwendung in einer anderen Lösung den Aufwand in Erstellung und Modellverständnis rechtfertigt.

Lektion 6 (Modulare Systemmodelle): Die Modularität der Modellelemente erleichtert nicht nur die Wiederverwendung, sondern trägt auch zur Verbesserung der Performance der verwendeten Modellierungstools bei. Durch die Nutzung von Versionsverwaltung, Variantenmanagement und Bibliothekskonzepten bereits beim Aufsetzen der Modelle kann das Prinzip der Modularität und des Baukastensystems effektiv genutzt werden, welches darüber hinaus ein durchgängiges Prinzip in der Entwicklung automobiler Systeme darstellt [10]. Dazu ist es wichtig, zwischen einer lösungsneutralen und produktspezifischen Schnittstelle zu unterscheiden und entsprechende Pakete auf den Dekompositionsebenen oder Systemschnitten anzulegen, um die Förderung eines systemorientierten Denkens zu unterstützen. Des Weiteren sollten Modelle in kleine, wiederverwendbare Artefakte aufgeteilt werden, um die Flexibilität zu erhöhen und die Wiederverwendbarkeit zu verbessern. Modelle sollten so strukturiert sein, dass sie ein gemeinschaftliches Arbeiten erleichtern und gleichzeitig eine optimale Leistung gewährleisten. Durch diese Maßnahmen zur Förderung der Modularität können wir die Effizienz der Systemmodellierung steigern und die Wiederverwendbarkeit von Modellen verbessern.

Lektion 7 (Wechselwirkung zwischen Organisationsstruktur und Systemmodell). Die Organisationsstruktur und das Systemmodell beeinflussen sich gegenseitig [10]. Diese Wechselwirkung muss sowohl im Modell als auch in der Projekt- und Organisationsplanung berücksichtigt werden.

5 Fazit

Im Rahmen dieses Beitrags haben wir folgende Aspekte untersucht: Erstens, die Identifikation der benötigten Entwicklungsartefakte in der automobilen Systementwicklung. Zweitens, die effiziente Modellierung dieser Entwicklungsartefakte mithilfe von Modellen auf verschiedenen Abstraktionsniveaus. Durch das Generieren von Modellen auf konkreteren Ebenen aus Modellen auf höheren Abstraktionsniveaus können redundante Artefakte vermieden werden. Sichten sollten nicht als separate Aufgabe des Modellierers dienen, um redundante Artefakte zu erzeugen. Stattdessen sollten sie aus anderen Perspektiven gewonnen werden, beispielsweise durch Modeltransformationen. Drittens haben uns Erfahrungen aus der industriellen Anwendung dieser Methoden vorgestellt. Zusammenfassend kann festgehalten werden, dass die Ableitung von Designartefakten unter Berücksichtigung dieser Lektionen nicht nur eine verbesserte Nachvollziehbarkeit ermöglicht, sondern auch die effiziente Modellierung unterstützt. Die Anwendung einer einheitlichen Modellierungsmethodik und gezielte Modelltransformationen ermöglichen uns, die Effektivität und Qualität unserer Entwicklungsarbeit zu steigern.

Literaturverzeichnis

[1] C. Granrath, C. Kugler, S. Silberg, M.-A. Meyer, P. Orth, J. Richenhagen und J. Andert, „Feature-driven systems engineering procedure for standardized product-line development," Systems Engineering, Bd. 24, Nr. 6, pp. 456-479, 2021.

[2] M. Dalibor, N. Jansen, B. Rumpe, L. Wachtmeister und A. Wortmann, „Model-Driven Systems Engineering for Virtual Product Design," in Proceedings of MODELS 2019, Munich, 2019.

[3] ISO/IEC/IEEE 42010 Systems and Software Engineering – Architecture Description, 2022.

[4] M. Broy, M. Gleirscher, P. Kluge, W. Krenzer, S. Merenda und D. Wild, „Automotive architecture framework: Towards a holistic and standardised system architecture description," 2009.

[5] R. Hutcheson, D. Mcadams, R. Stone und I. Tumer, „Function-based systems engineering (FUSE)," DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, pp. 715-716, 28-31 August 2007.

[6] J. G. Lamm und T. Weilkiens, „Method for Deriving Functional Architectures from Use Cases," Systems Engineering, Bd. 17, Nr. 2, pp. 225-236, 2014.

[7] J. Philipps und B. Rumpe, „Refinement of information flow architectures," First IEEE International Conference on Formal Engineering Methods, pp. 203-212, 1997.

[8] M. Broy und B. Rumpe, „Development Use Cases for Semantics-Driven Modeling Languages," Communications of the ACM, Bd. 66, Nr. 5, pp. 62-71, 2023.

[9] J. P. MacDuffie, „Modularity-as-property, modularization-as-process, and 'modularity'-as-frame: Lessons from product architecture initiatives in the global automotive industry," Global strategy journal, Bd. 3, Nr. 1, pp. 8-40, 20.

[10] M. E. Conway, „How do committees invent," Datamation, Bd. 14, Nr. 4, pp. 28-31, 1968.

[11] S. Hillemacher, N. Jäckel, C. Kugler, P. Orth, D. Schmalzing und L. Wachtmeister, „Artifact-Based Analysis for the Development of Collaborative Embedded Systems," in Model-Based Engineering of Collaborative Embedded Systems, Springer, 2021, pp. 315-331.

[12] A. Butting, T. Greifenberg, B. Rumpe und A. Wortmann, „On the Need for Artifact Models in Model-Driven Systems Engineering Projects," Software Technologies: Applications and Foundations, 2018.