

Variantenkomplexität in der automobilen Steuergerätekonfiguration: Status quo und Potenziale durch SPLE

Variant Complexity in Automotive Control Unit Configuration: Status quo and Potential through SPLE

Richard von Esebeck^{1,♦,*}, Yannick Lindebauer^{2,♦}, Thomas Vietor²

¹ Volkswagen AG

² Technische Universität Braunschweig, Institute for Engineering Design

♦ These authors contributed equally to this work

* Korrespondierender Autor:

Richard von Esebeck

Volkswagen AG, P.O. Box 1679/1583

38436 Wolfsburg

Deutschland

☎ 0176 8041 9060

✉ richard.von.esbeck@volkswagen.de

Abstract

Automotive manufacturers are increasingly confronted not only with a growing number of hardware components, but also with rising software variability, particularly due to the parameterization of ECU software. However, the parameter management methods currently in use are proving to be inadequate and no longer meet the demands of software-driven vehicles. This study investigates what a future methodology could look like, one that not only improves parameter management but also structurally integrates it with variant management. To this end, current processes at an OEM are analyzed, and key challenges in dealing with parameters are identified. Based on these findings, we propose a methodology that extends parameter management through a systematic variant management approach grounded in SPLE.

Keywords

Variant Management, ECU Software Configuration, Software Product Line Engineering

1. Einleitung & Motivation

Die Automobilindustrie ist geprägt von einer hohen Variantenvielfalt, die sich aus individuellen Kundenwünschen sowie markt- und länderspezifischen Regularien ergibt [1]. Die Kombination aus immer kürzeren Modellpflegezyklen und steigenden Kundenansprüchen, bezogen auf den Funktionsumfang und das Individualisierungspotential der Produkte ist eine Herausforderung vieler Industriebranchen [2]. Die Automobilindustrie begegnet dieser Herausforderung mit einer Ausweitung der ohnehin schon großen Variantenvielfalt [3].

Ein erheblicher Teil der Varianten wird heute über die Konfiguration von Steuergeräten realisiert: Die zugrunde liegende Software wird mithilfe von Kalibrierparametern an den jeweiligen Bauauftrag bzw. die konkrete fahrzeugspezifische Ausprägung hinsichtlich Ausstattung und gesetzlichen Anforderungen angepasst [4]. Die zentrale Herausforderung liegt dabei in der enormen Anzahl möglicher Varianten und der systematischen und fehlerfreien Zuordnung einer großen Menge an Kalibrierparametern zu diesen Variantenmerkmalen [5]. Parallel dazu befindet sich die Branche in einem tiefgreifenden technologischen Wandel: Der Übergang vom hardwarezentrierten Fahrzeug zum Software-Defined Vehicle (SDV) führt zu einer Verschiebung der Komplexität in Richtung Software [6]. Damit einher geht ein signifikanter Anstieg der Vielfalt in der Softwarevarianz sowie in der Parametrierung.

Die derzeit in der Industrie etablierten Verfahren und Prozesse zur Steuergeräteparametrierung sind historisch gewachsen und proprietär geprägt. Sie basieren vielfach auf manuellen Tätigkeiten, sind in hohem Maße fehleranfällig und nicht an die Anforderungen zunehmend digitalisierter Fahrzeugarchitekturen sowie die Erfordernisse des SDVs ausgerichtet. Vor diesem Hintergrund ist es zwingend notwendig, bestehende Konzepte zur Steuergerätekonfiguration methodisch neu auszurichten.

Im Rahmen dieser Arbeit wird eine empirische Analyse des bestehenden Prozesses der Steuergerätekonfiguration bei einem führenden Automobilhersteller durchgeführt. Der Schwerpunkt liegt auf der Untersuchung der prozessualen Abläufe und strukturellen Rahmenbedingungen, insbesondere im Hinblick auf das Parametermanagement, zentrale Variantentreiber sowie weitere Komplexitätstreiber im Kontext des Entwicklungsprozesses.

Auf Basis dieser Analyse werden die zentralen Herausforderungen des derzeit praktizierten Konfigurationsprozesses in Bezug auf die Steuerung und Verwaltung von Parametern systematisch identifiziert. Darauf aufbauend wird ein neues, konzeptionell weiterentwickeltes Modell für die Steuergerätekonfiguration und das Parametermanagement vorgestellt, das auf einem erweiterten SPLE-Framework basiert und die Herausforderungen adressiert.

2. Hintergrund

Im Rahmen des untersuchten Anwendungsfalls zur Konfiguration und Bedatung von Steuergerätesoftware in der Automobilindustrie zeigt sich, dass Konfigurationselemente sowohl der Software als auch dem jeweiligen Steuergerät eindeutig zugeordnet werden müssen. Dies erfordert eine strukturierte Anbindung an das Variantenmanagement sowohl auf der Hardware- als auch auf der Softwareseite. Um den Kontext unseres spezifischen Use-Cases angemessen einzuordnen, werden im Folgenden zunächst die grundlegenden Konzepte des Hardware- sowie des Software-Variantenmanagements skizziert. Auf dieser Basis erfolgt anschließend die allgemeine Erklärung und Relevanz unseres Anwendungsfalls.

2.1. Variantenmanagement

Variantenmanagement beschreibt die systematische Steuerung und Organisation von Unterschieden zwischen verschiedenen Produktvarianten innerhalb einer Produktlinie. Für die Modellierung von Produktlinien bzw. Produktvarianten existieren eine Vielzahl methodischer Ansätze, die je nach Domäne und Anwendungsfall variieren können [7].

Eine Produktlinie lässt sich als eine Gruppe von Produkten definieren, die auf einem gemeinsamen Baukasten von Artefakten basieren. Diese Artefakte bilden die funktionale Grundlage zur Realisierung der jeweiligen Produktausprägungen [8]. Obwohl der Begriff „Produktlinie“ von der Softwareentwicklung geprägt ist, weist er inhaltliche Parallelen zum in der Konstruktionsmethodik etablierten Baukastenprinzip auf [9]. Unabhängig von der verwendeten Terminologie liegt beiden Konzepten dasselbe Grundprinzip zugrunde: die Konfiguration individueller Produktvarianten durch gezielte Kombination von Artefakten - also Bausteinen aus einem vordefinierten Baukasten [3]. Zu den typischen Artefakten zählen physische Komponenten wie Bauteile und Baugruppen, aber auch Softwarekomponenten, sofern softwarebezogene Produktaspekte berücksichtigt werden.

Ein Produkt gilt als Variante, wenn es sich von einem anderen Produkt mit vergleichbarem Verwendungszweck in mindestens einem Bestandteil unterscheidet [10]. Aufgrund der einheitlichen übergeordneten Zielsetzung innerhalb einer Produktlinie verfügen sowohl Produktlinien als auch Baukastensysteme in der Regel über eine gemeinsame Basisstruktur. Diese wird durch definierte Variationspunkte ergänzt, an denen unterschiedliche funktionale Ausprägungen wählbar sind. Durch die Kombination dieser Artefakte lassen sich spezifische Produktkonfigurationen erzeugen, die auf individuelle Anforderungen abgestimmt sind [3].

Im Kontext des Hardware-Variantenmanagements manifestiert sich die Variabilität primär durch physisch unterschiedliche Komponenten, etwa durch den Einsatz verschiedener Sensortypen oder Motorvarianten. Die Verwaltung und Steuerung solcher Varianten erfolgt üblicherweise über klassische Stücklisten in Verbindung mit Konfigurationsmanagementsystemen. Solche Konfigurationsmanagementsysteme - in der Praxis oft als Produktkonfiguratoren bezeichnet - unterstützen die Modellierung und Auswahl von Produktvarianten auf Basis definierter Entscheidungslogiken. Dabei kommen unter anderem Entscheidungs-/Konfigurationstabellen, Entscheidungs-/Konfigurationsmatrizen oder Entscheidungs-/Konfigurationsbäume zum Einsatz.

2.2. Software-Variantenmanagement

Vor dem Hintergrund zunehmender Kundenanforderungen an digitalisierte Fahrzeuge gewinnt Software als zentrales Element moderner Fahrzeugsysteme stetig an Bedeutung. In heutigen Fahrzeugen sind bereits Softwareumfänge von bis zu 150 Millionen Zeilen Code realisiert [11, 12] - mit entsprechend wachsender Komplexität in Entwicklung, Wartung und Konfiguration. Da die Softwarefunktionalitäten zunehmend an individuelle Kundenwünsche angepasst werden müssen [13], entsteht eine Vielzahl möglicher Softwarevarianten, deren Handhabung und Steuerung erhebliche Herausforderungen mit sich bringt. Zur Bewältigung dieser Herausforderungen ist ein systematisches Software-Variantenmanagement erforderlich. Ziel ist es, Gemeinsamkeiten zwischen Varianten gezielt zu nutzen, Wiederverwendung zu fördern und gleichzeitig die Anzahl sowie Vielfalt der Varianten beherrschbar zu halten [14]. Für diesen Zweck wurde das Konzept des Software Product Line Engineering (SPLE) entwickelt. Zentrale Grundlage des SPLE-Ansatzes ist eine gemeinsame Softwareplattform, aus der spezifische Produktvarianten abgeleitet werden. Das SPLE-Framework gliedert sich hierzu in zwei komplementäre Teilbereiche: das Domain Engineering und das Application Engineering [15]. Im Domain Engineering werden die wiederverwendbaren Kernbestandteile sowie die identifizierte Variabilität der Plattform definiert und strukturiert aufbereitet. Das Application Engineering hingegen fokussiert sich auf die Ableitung konkreter Produktvarianten, die auf Basis der Plattformmerkmale zusammengesetzt und angepasst werden [15]. Eine Darstellung dieses zweigeteilten Frameworks findet sich in Abbildung 1.

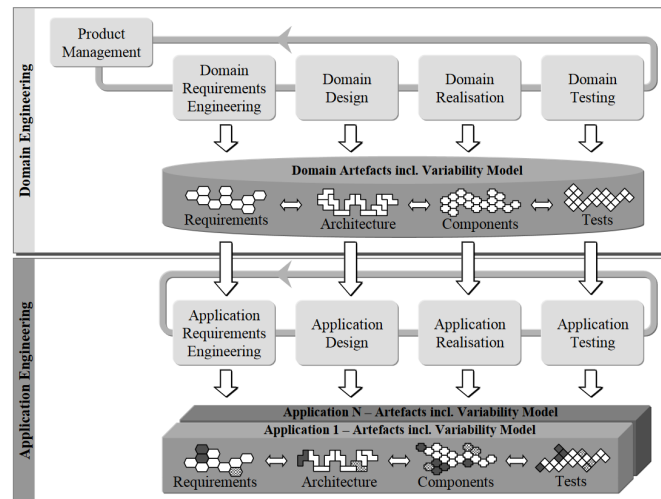


Abbildung 1: SPLE Framework [15]

Ein zentraler Baustein des SPLE ist die Modellierung der Variabilität. Hierfür existieren unterschiedliche Ansätze, wobei die sogenannte Feature-Modellierung in der Praxis am weitesten verbreitet ist. Feature-Modelle beschreiben Eigenschaften eines Systems in strukturierter Form. In der Regel erfolgt dies über eine hierarchische Baumstruktur, in der einzelne Features - also Systemeigenschaften oder -funktionen - abgebildet sind [16]. Diese können dabei in unterschiedlichen Beziehungen zueinander stehen, etwa in Form von Alternativen, OR, XOR, optionalen oder verpflichtenden Elementen. Zudem lassen sich semantische Abhängigkeiten zwischen einzelnen Features definieren, etwa in Form von requires-/excludes-Beziehungen [3]. Ein Beispiel für ein einfaches Feature-Modell ist in Abbildung 2 dargestellt.

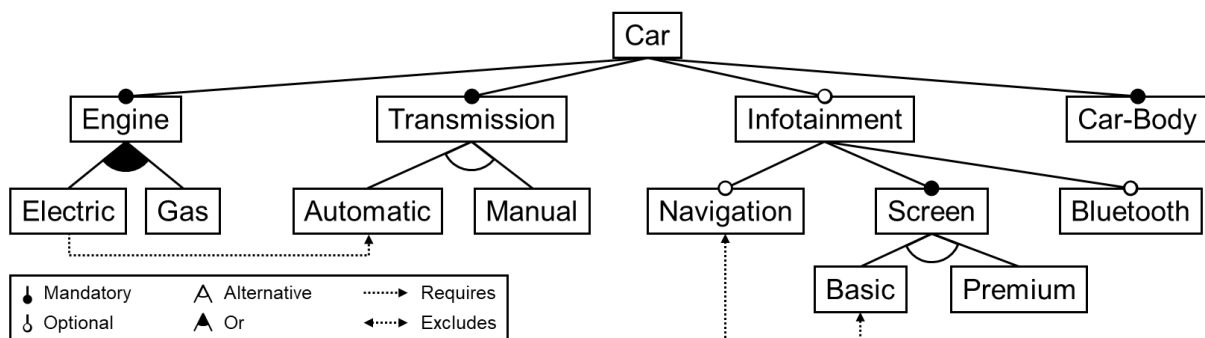


Abbildung 2: Beispielhaftes Feature Modell

Die Feature-Modelle werden im Rahmen des Domain Engineering erstellt und dienen der systematischen Abbildung der Variabilität innerhalb der Softwareplattform. Dabei werden die einzelnen Features mit den entsprechenden Softwareartefakten verknüpft. Auf diese Weise entsteht eine durchgängige Verbindung zwischen der Variabilitätsbeschreibung und den jeweiligen Artefakten. Im Application Engineering ermöglicht diese Verknüpfung, dass bei der Auswahl einer spezifischen Produktvariante automatisch die zugehörige Konfiguration sowie die relevanten Softwareartefakte identifiziert und generiert werden können.

3. Forschungsfragen und Vorgehensweise

Angeichts des Paradigmenwechsels von hardware- zu softwarezentrierten Fahrzeugsystemen und der damit einhergehenden zunehmenden Variantenvielfalt in der Steuergerätekonfiguration ist es essenziell, auch die zugrunde liegenden Entwicklungs- und

Konfigurationsprozesse weiterzuentwickeln und an die Anforderungen dieser neuen Realität anzupassen. Im Vorfeld dieser Arbeit wurde dazu eine Befragung von Industrien mit hoher Variantenvielfalt durchgeführt, welche die Relevanz dieser Problemstellung bestätigt hat. Die Automobilindustrie ist aufgrund der hohen Stückzahl und hohen Individualisierbarkeit ein Extrembeispiel der Steuergerätekonfiguration. Ziel dieser Arbeit ist es daher, die bestehenden Konzepte am Beispiel der Volkswagen AG zu analysieren, Schwachstellen zu identifizieren und methodisch fundierte Lösungsansätze zu erarbeiten. Konkret verfolgen wir einen zweistufigen Forschungsansatz: Im ersten Schritt soll exemplarisch die bestehende Methodik der Steuergerätekonfiguration einschließlich des variantenbezogenen Parametermanagements detailliert untersucht und dokumentiert werden. Ziel ist es, ein fundiertes Verständnis der aktuellen Praxis zu gewinnen und zentrale Herausforderungen systematisch zu identifizieren. Im zweiten Schritt wird analysiert, inwieweit diese Herausforderungen mit den existierenden Methodiken des Variantenmanagements bewältigt werden können. Dabei werden auch gezielt Erweiterungen vorgeschlagen, um den spezifischen Anforderungen der Steuergerätekonfiguration in der Automobilindustrie gerecht zu werden. Aus diesem Vorgehen ergeben sich zwei zentrale Forschungsfragen:

1. Wie gestaltet sich der aktuelle Prozess des Parametermanagements bei Volkswagen und welche Herausforderungen bestehen dabei?
2. Wie können die Herausforderungen adressiert werden, wobei gleichzeitig die aktuellen und zukünftigen Anforderungen der Automobilindustrie berücksichtigt werden?

Zur Beantwortung der ersten Forschungsfrage wurde eine umfassende Analyse des Prozesses bei Volkswagen durchgeführt. Im Rahmen dieser Analyse wurden zahlreiche Gespräche mit Experten und relevanten Stakeholdern entlang des Entwicklungs- und Freigabeprozesses geführt. Auf dieser Grundlage konnte ein differenziertes Bild des gegenwärtigen Parametermanagements und der Steuergerätekonfiguration gezeichnet werden, aus dem die zentralen Herausforderungen abgeleitet wurden. Im weiteren Verlauf der Arbeit wird darauf aufbauend eine methodische Lösung vorgestellt, die diese Herausforderungen adressiert.

4. Untersuchung der Steuergerätekonfiguration bei der Volkswagen AG

Im Rahmen der durchgeführten Untersuchung wird zunächst das bestehende Konzept mit seinen zentralen Eigenschaften und Strukturelementen vorgestellt. Darauf aufbauend werden die identifizierten Herausforderungen sowie deren konkrete Auswirkungen auf die Praxis analysiert und systematisch dargestellt. Im Anschluss daran erfolgt eine kontextuelle Einordnung, die sowohl die ursprünglichen Anforderungen zum Zeitpunkt der Konzeptentstehung als auch die veränderten Rahmenbedingungen und Anforderungen der heutigen Zeit berücksichtigt. Die Konfiguration bzw. Bedatung von Steuergerätesoftware erfordert eine präzise und nachvollziehbare Zuordnung von Parametern zu spezifischen Merkmalen des Fahrzeugs. Um diesen Prozess ganzheitlich zu verstehen und systematisch einordnen zu können, wurden im Rahmen der Untersuchung die folgenden zentralen Aspekte analysiert:

- Welche Konfigurationselemente existieren und welche Funktionen erfüllen sie?
- Welche Merkmale des Fahrzeugs werden durch die Konfiguration abgebildet und was sind die maßgeblichen Variantentreiber?
- Wie erfolgt die Zuordnung von Konfigurationselementen zu den Fahrzeugmerkmalen?
- Wie wird die Zuordnung gesteuert?

Die Konfigurationselemente umfassen Parameter und Datensätze (z.B. Kennlinien), die Funktionen aktivieren oder deaktivieren können (z.B. mittels `ifdef`-Anweisungen) oder bestimmte Werte festlegen. Im aktuellen Entwicklungsprozess werden Softwarecode und Kalibrierparameter als klar voneinander getrennte Entitäten behandelt. Die Entwickler des Softwarecodes sind in der Regel nicht für die Definition der Parameter und deren Werte verantwortlich. Dadurch verlaufen diese beiden Entwicklungsstränge weitgehend unabhängig voneinander und werden erst in einem späteren Prozessschritt wieder zusammengeführt - in der Hoffnung, dass Parameter und Software funktional und semantisch zueinander passen.

Die Merkmale und die Ausstattung des Fahrzeugs werden durch sogenannte PR-Nummern repräsentiert. Diese dreistelligen alphanumerischen Codes abstrahieren spezifische Merkmale. Dabei können diese PR-Nummern verschiedenes darstellen, von z.B. Ausstattungsvarianten wie Recht- oder Linkslenker bis zu Länderkodierungen.

Die Verbindung zwischen den Parametern und den PR-Nummern bildet das zentrale Element der Steuergerätekonfiguration. Diese Verbindung wird in einem Datencontainer hergestellt, in dem die PR-Nummern aufgelistet und die verschiedenen Parameterausprägungen dazu in Relation gesetzt werden. Im Allgemeinen wird ein Datencontainer für ein Steuergerät erstellt, sodass für ein Fahrzeug eine Vielzahl von Datencontainern existiert, die alle notwendigen Kombinationen von Parametern und Ausstattungen enthalten.

Beim Bau eines Fahrzeugs wird ein Datencontainer anhand des fahrzeugspezifischen Bauauftrags, der aus PR-Nummern besteht, „interpretiert“ und daraus ein fahrzeugspezifischer String erstellt, mit dessen Hilfe die Steuergeräte im Fahrzeug konfiguriert werden. Um den korrekten Datencontainer auszuwählen, sind zusätzliche Steuerungselemente erforderlich: zum einen für die zeitliche Steuerung der Datencontainer, da verschiedene Versionen der Parameter existieren können, zum anderen für die Verbindung zum Steuergerät und damit zur Stückliste. Da die Software und deren Parametrierung schnelllebig sind und häufig geändert werden müssen, sind die Datencontainer nicht direkt in der Stückliste eingebettet. Die Prozesse dafür sind zu aufwändig und zeitintensiv. Stattdessen werden die Datencontainer mithilfe einer Tabelle gesteuert, in der definiert wird, wann (mithilfe von Einsatz- und Entfallterminen) ein Datencontainer für welches Steuergerät gültig ist. Dies stellt die Verbindung zur Stückliste dar - die in der Tabelle für die Identifikation des Steuergerätes genannte Nummer ist eine gültige Teilenummer in der Stückliste und beschreibt das Steuergerät und damit die Hardware und Software zusammen.

Das Parametermanagement, einschließlich der Zuordnung zu Varianten und Steuerungselementen, birgt mehrere Herausforderungen, von welchen die wichtigsten im Folgenden beschrieben werden. Eine wesentliche Herausforderung besteht in der mangelhaften Nachvollziehbarkeit und Durchgängigkeit der Daten. Der Prozess wird in proprietären Systemen durchgeführt, die zahlreiche Toolbrüche aufweisen. Zudem existieren keine systeminternen Verbindungen zwischen den Parametern und deren Werten zu Anforderungen, Funktionen oder Varianten. Dies führt zu einem hohen manuellen Arbeitsaufwand und erhöht die Wahrscheinlichkeit menschlicher Fehler. Darüber hinaus gibt es viel implizites Wissen, das hauptsächlich in den Köpfen der Ingenieure vorhanden ist und nicht dokumentiert oder geteilt wird. Ein weiterer Aspekt, der berücksichtigt werden muss, ist die Nachvollziehbarkeit von Änderungen. Nur durch einen manuellen Vergleich zweier verschiedener Versionen von Datencontainern kann festgestellt werden, was geändert wurde, was wiederum einen hohen Arbeitsaufwand zur Folge hat. Es gibt weitere Anwendungsfälle zur Analyse, bei denen es von Nachteil ist, dass die Parameter und deren Zuordnungen nicht einfach nachvollziehbar sind und es kein zusammenhängendes Bild der Parametrierung gibt. Eine weitere Schwierigkeit ist die Inkonsistenz und Limitierung der Variantenbeschreibung. Die Limitierung der Variantenbeschreibung ergibt sich daraus, dass die PR-Nummern dreistellige Codes sind, was zu einer begrenzten und unzureichenden Anzahl von Codes führt. Die Inkonsistenz entsteht durch unterschiedliche Anforderungen je nach Fahrzeugprojekt und die

darauf folgende Anpassung der PR-Nummern. Diese Flexibilität führt zu Variantenmodellen, die über Fahrzeugprojekte hinweg inkonsistent sind, was die Variantenvielfalt und notwendige Dokumentation erhöht und es deutlich schwieriger macht, Änderungen an Parametrierungen über einen längeren Zeitraum und verschiedene Fahrzeugprojekte hinweg zu verfolgen.

Die Steuerung der Parameter birgt weitere Herausforderungen, insbesondere die zeitliche Steuerung, die eine potenzielle Fehlerquelle darstellt. Dies lässt sich am folgenden Beispiel verdeutlichen: In der EU ist durch die UNECE R156 festgelegt, dass ein Software-Update-Management-System (SUMS) implementiert werden soll. Das SUMS wird mittels eines Baseline-Konzepts umgesetzt, bei dem zu einem bestimmten Zeitpunkt bzw. einer Baseline E/E-Umfänge, darunter auch Software, Steuergeräte und Datencontainer, zusammengefasst, homologiert und freigegeben werden. Schwierigkeiten entstehen beispielsweise dadurch, dass die Vielzahl an Datencontainern individuell zeitlich über Einsatz- und Entfalltermine gesteuert werden, aber letztlich in einer Baseline zusammengeführt werden sollen. Menschliche Fehler bei der Zuordnung der Datencontainer können zu einem erheblichen Mehraufwand führen, um die korrekten Datencontainer der Baseline zuzuordnen. Ein weiterer Aspekt ist die Steuerung mittels der Tabelle zu den Steuergeräten. Diese Vorgehensweise lehnt sich an die Arbeit mit der Stückliste an und entspricht nicht den Anforderungen des Software Engineerings, bei dem Änderungen schnell und nachvollziehbar umgesetzt werden müssen. Zudem entsteht hier ein weiteres Objekt, das eigenständig gepflegt werden muss und somit Arbeitsressourcen bindet. In Tabelle 1 stellen wir die festgestellten Herausforderungen und deren Auswirkungen übersichtlich und zusammengefasst dar:

Tabelle 1: Herausforderungen und deren Auswirkungen im Prozess

Herausforderung	Auswirkung
Fehlende Nach- und Rückverfolgbarkeit, Wechselwirkungen von Konfigurationen einzelner ECUs	Schwierige Analyse bei Änderungen und Fehlern
Implizites Wissen	Wissen ist nicht breit verfügbar und liegt nur bei spezifischen Ingenieuren
Harte Trennung Parameter & Software	Inkonsistenz und Fehler bei Zuordnung von Parameter zur Software
Keine E2E Tool-Durchgängigkeit	Inkonsistente Daten zwischen verschiedenen Tools, Kommunikation zwischen Verantwortlichen schwierig
Inkonsistente Variantenbeschreibung	Pflege der Daten im Lifecycle ist aufwändig oder schwer möglich
Steuerung zur Stückliste	Weiteres Objekt, welches gepflegt werden muss und dadurch Mehraufwand bringt
Zeitliche Steuerung der Datencontainer	Mehraufwand bei der Erstellung der Baseline durch inkonsistente Datencontainer

Das aktuell eingesetzte Konzept zur Steuergerätekonfiguration entstammt einer Zeit, in der primär hardwarebezogene Anforderungen das Fahrzeug dominierten. Dementsprechend finden sich darin vor allem Strukturen und Elemente wieder, die ihren Ursprung im Hardware-Variantenmanagement haben. Ein zentrales Beispiel hierfür ist die sogenannte Steuerungstabelle, die als schnelllebige Variante einer Stückliste fungiert. Sie enthält unter anderem Einsatz- und Entfalltermine von Komponenten und übernimmt damit grundlegende Aufgaben der zeitlichen Steuerung - ähnlich wie es auch in der Stückliste erfolgt. Allerdings fehlt es diesem Ansatz an einer durchgängigen Verknüpfung zu anderen im Entwicklungsprozess essenziellen Artefakten, was eine direkte Folge veralteter Systeme und Prozesse ist, die in der damaligen technischen Realität vollkommen ausreichend waren. Dieses Konzept erweist sich jedoch als unzureichend für die heutigen Anforderungen an moderne Fahrzeuge. Es basiert auf einem Entwicklungsmodell, das davon ausgeht, dass nach dem SOP des Fahrzeuges keine wesentlichen Änderungen oder Dokumentationspflichten

mehr bestehen und Software lediglich eine untergeordnete Rolle spielt. Inzwischen hat sich die Realität grundlegend gewandelt: Mit dem Aufkommen von SDVs, OTA-Updates sowie regulatorischen Vorgaben wie UNECE R156 sind neue Anforderungen, vor allem für die Software entstanden, für die das bestehende Konzept nicht ausgelegt ist.

Um diesen veränderten Rahmenbedingungen gerecht zu werden, ist eine grundlegende Anpassung erforderlich. Insbesondere müssen Methoden und Konzepte aus dem Software-Variantenmanagement sowie dem modernen Software Engineering in ein neues, integratives Gesamtkonzept überführt werden. So lassen sich die in Kapitel 4 identifizierten Herausforderungen adäquat adressieren und eine zukunftsfähige, flexible und nachvollziehbare Steuerung von Softwarevarianten und -parametern sowie deren korrekte Zuordnung zu Varianten gewährleisten.

5. Integration der Steuergerätekonfiguration in ein erweitertes SPLE-Framework

Zur Bewältigung der identifizierten Herausforderungen im Variantenmanagement bietet das SPLE eine geeignete methodische Grundlage. Das SPLE-Framework, welches primär zwischen Domain Engineering und Application Engineering unterscheidet, erweist sich jedoch als nicht hinreichend differenziert, um die spezifischen Anforderungen der automobilen Steuergerätekonfiguration adäquat abzubilden. Daher greifen wir auf eine erweiterte Interpretation des SPLE-Frameworks zurück, die zusätzlich zwischen einem Problem Space und einem Solution Space unterscheidet. Diese Struktur erlaubt eine klarere Trennung der Variantenmerkmale von den konkreten Assets wie Parametern, Datensätzen, Anforderungen und Test Cases. Die Konzeption der Steuergerätekonfiguration wird in dieses erweiterte SPLE-Framework integriert. Zudem erfordert der spezifische Anwendungsfall zusätzliche methodische Weiterentwicklungen, die im von uns entwickelten Framework (siehe Abbildung 3) berücksichtigt werden.

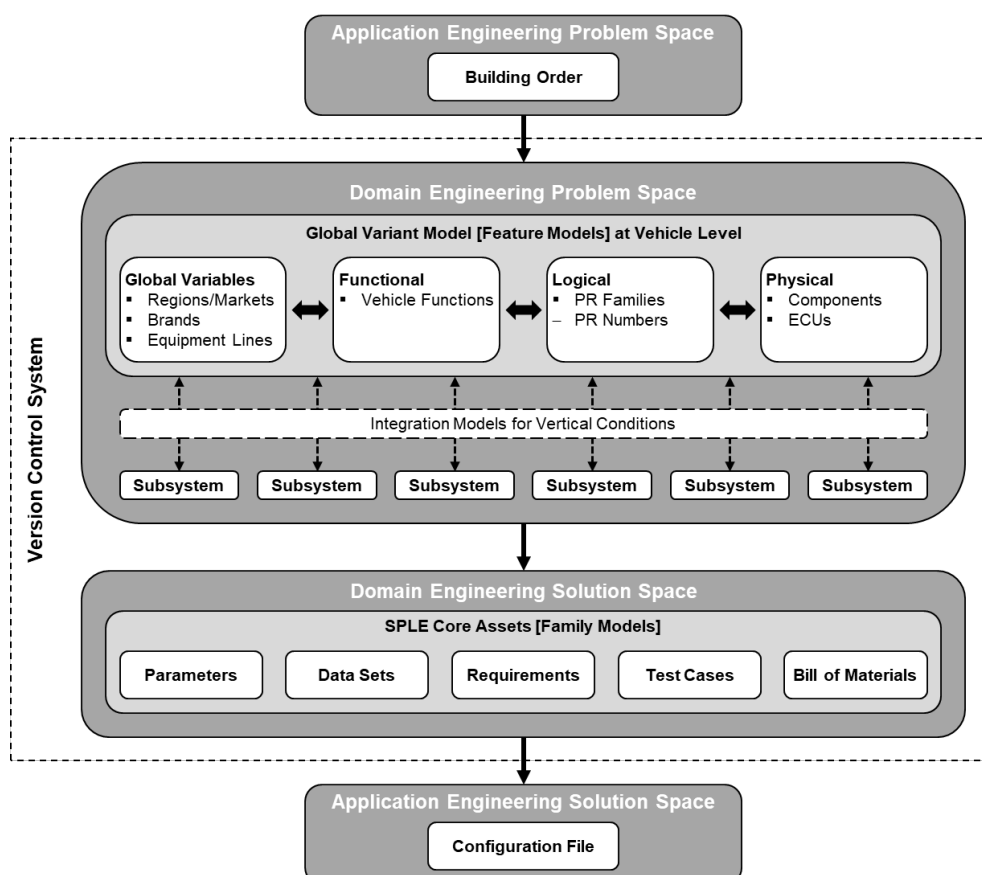


Abbildung 3: Framework zur Steuergerätekonfiguration

Die grundlegenden Ein- und Ausgangsgrößen der Steuergerätekonfiguration bleiben in weiten Teilen unverändert. Als zentrale Eingangsgröße fungiert der Bauauftrag, der im Wesentlichen eine Liste von PR-Nummern enthält. Innerhalb des SPLE-Frameworks ist dieser Bauauftrag dem Application Engineering / Problem Space zuzuordnen.

Im Domain Engineering / Problem Space sieht das Framework auf der obersten Ebene ein globales Variantenmodell auf Fahrzeugebene vor. Dieses setzt sich aus vier miteinander verknüpften Feature-Modellen zusammen: Logical Feature Model bildet die vollständige Menge an PR-Familien und zugehörigen PR-Nummern ab - Functional Feature Model bildet die technischen Fahrzeugfunktionalitäten ab - Global (Context) Variables Feature Model bildet übergeordnete Einflussgrößen wie Zielmärkte / Länder, Marken und Ausstattungslinien ab - Physical Feature Model bildet physische Komponenten des Fahrzeugs wie Steuergeräte ab.

Ergänzend dazu existieren Subsystem-spezifische Feature-Modelle, die auf einer separaten Ebene modelliert sind. Diese sind über vertikale Integrationsbeziehungen in Form individueller Integrationsmodelle mit dem globalen Variantenmodell verbunden.

Im Domain Engineering / Solution Space befinden sich die SPLE Core Assets in Form von Family Models: je ein Family Model für Parameter, Datensätze, Anforderungen, Test Cases und die Stückliste. Im Application Engineering / Solution Space resultiert aus dem Framework das sogenannte Configuration File. Dabei handelt es sich um eine fahrzeugspezifische Konfigurationsdatei, die zu 100 % auf den jeweiligen Bauauftrag zugeschnitten ist.

Das entwickelte Framework adressiert in weiten Teilen die zuvor identifizierten Herausforderungen im Bereich der variantenbezogenen Parametrierung. Darüber hinaus eröffnet es vielfältige Analysepotenziale, die über den ursprünglichen Anwendungsrahmen hinausgehen. So ermöglicht das Framework beispielsweise die transparente Darstellung von Abhängigkeiten zwischen Parametern, Funktionen und weiteren E/E-Artefakten. Aufbauend auf dieser strukturierten Datenbasis lassen sich weiterführende Analysen durchführen, etwa zur Identifikation funktionaler oder technischer Cluster. Solche Analysen ermöglichen es, Parametergruppen zu erkennen, die in engem Zusammenhang mit bestimmten Funktionen stehen. Diese Erkenntnisse können künftig als Grundlage für die Entwicklung neuer Konzepte der Parametrierung dienen - etwa durch gezielte Modularisierung, funktionale Gruppierung oder optimierte Übertragungsstrategien. Damit stellt das Framework nicht nur eine methodische Lösung dar, sondern fungiert auch als Enabler für eine datengetriebene Weiterentwicklung bestehender Parametermanagementansätze.

5.1. Konfigurationsprozess im Rahmen des neuen Frameworks

Der Ablauf beginnt mit dem Einlesen des Bauauftrags, wobei die enthaltenen PR-Nummern im Logical Feature Model des globalen Variantenmodells selektiert werden. Aufgrund der modellierten Abhängigkeiten innerhalb des Variantenmodells werden hierdurch automatisch weitere Merkmale in den anderen Feature-Modellen sowie in den Subsystemen ausgewählt. In einem nachgelagerten Schritt erfolgt auf Basis der gewählten Features eine automatisierte Selektion der entsprechenden SPLE Assets. Im letzten Schritt wird daraus das Konfigurationsdatei generiert. Diese weitgehend automatisierte Verarbeitung ist nur deshalb möglich, weil alle relevanten Abhängigkeiten zwischen Problem- und Solution Space - also zwischen Feature Models und Family Models - im Domain Engineering explizit modelliert sind.

6. Zusammenfassung & Ausblick

In dieser Arbeit wurden die Herausforderungen der Variantenvielfalt in der Softwareentwicklung am Beispiel eines Automobilherstellers untersucht. Aufbauend auf dieser Analyse wurde ein Framework vorgeschlagen, das auf den Prinzipien des SPLE basiert und speziell auf die Konfiguration von Steuergerätesoftware zugeschnitten ist. Das vorgestellte Framework adressiert zentrale Defizite in der heutigen industriellen Praxis. Ein wesentlicher

Beitrag besteht darin, dass domänenspezifisches Wissen nicht länger ausschließlich implizit in den Köpfen einzelner Experten verbleibt, sondern systematisch, zentral und zugänglich dokumentiert wird. Darüber hinaus erlaubt ein differenziertes Rollen- und Berechtigungskonzept die gezielte Steuerung des Zugriffs und der Bearbeitungsrechte für unterschiedliche Nutzergruppen. Die vollständige Integration aller Konfigurationsprozesse innerhalb des Frameworks gewährleistet eine End-to-End-Durchgängigkeit der Toollandschaft und eine konsistente Variantenbeschreibung über den gesamten Produktlebenszyklus hinweg. Gleichzeitig bleiben verschiedene Aspekte offen, die im Rahmen weiterer Forschung bearbeitet werden müssen. Hierzu zählen insbesondere Fragen zur hierarchischen Strukturierung und detaillierten Modellierung von Feature-Modellen, zur Ausgestaltung kollaborativer Arbeitsprozesse sowie zur konkreten methodischen Umsetzung der zeitlichen Steuerung. Diese offenen Punkte bilden die Grundlage für die nächste Phase unserer Forschungsarbeit.

Disclaimer

Ergebnisse, Meinungen und Schlüsse dieser Publikation sind nicht notwendigerweise die der Volkswagen Aktiengesellschaft.

7. Literatur

- [1] Schindewolf, Marc ; Wittler, Jan Willem ; Kühn, Thomas ; Grimm, Daniel ; Sax, Eric: A Model-Based Approach to Automotive Feature Development for Updates and Upgrades. In: 2023 IEEE International Conference on Service-Oriented System Engineering (SOSE) : IEEE, 2023, S. 19–26
- [2] Fricke, Ernst ; Schulz, Armin P.: Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. In: Systems Engineering 8 (2005), Nr. 4
- [3] Huth, Tobias ; Inkermann, David ; Vietor, Thomas: Ein Ansatz für eine integrierte, modellbasierte Anforderungs- und Variantenmodellierung. 2017
- [4] Begizhonov, Shakhrom ; Buyvol, Polina ; Makarova, Irina ; Tsybunov, Eduard: Parameterization of the ABS electronic control unit for increasing the autonomous trucks' active safety. In: MATEC Web of Conferences 341 (2021), S. 26
- [5] Manz, Christian ; Stupperich, Michael ; Reichert, Manfred: Towards Integrated Variant Management in Global Software Engineering: An Experience Report. In: 2013 IEEE 8th International Conference on Global Software Engineering : IEEE, 2013, S. 168–172
- [6] Streloke, Ludwig ; Franke, Jörg: Electrifying the Road: Disruptive Shifts in Automotive Value Creation. In: 2024 1st International Conference on Production Technologies and Systems for E-Mobility (EPTS) : IEEE, 2024, S. 1–8
- [7] Haag, Albert: Managing variants of a personalized product. In: Journal of Intelligent Information Systems 49 (2017), Nr. 1, S. 59–86
- [8] Schmid, Klaus: Produktlinienentwicklung. In: Informatik-Spektrum 33 (2010), Nr. 6, S. 621–625
- [9] Vietor, Thomas ; Stechert, Carsten: Produktarten zur Rationalisierung des Entwicklungs- und Konstruktionsprozesses. In: Feldhusen, Jörg; Grote, Karl-Heinrich (Hrsg.): Pahl/Beitz Konstruktionslehre. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013, S. 817–871
- [10] Firschau, N.L.: Variantenoptimierende Produktgestaltung. Göttingen, 2003
- [11] Mallozzi, Piergiuseppe ; Pelliccione, Patrizio ; Knauss, Alessia ; Berger, Christian ; Mohammadiha, Nassar: Autonomous Vehicles: State of the Art, Future Trends, and Challenges. In: Dajsuren, Yanja; van den Brand, Mark (Hrsg.): Automotive Systems and Software Engineering. Cham : Springer International Publishing, 2019, S. 347–367
- [12] Staron, Miroslaw: Automotive Software Architectures. Cham : Springer International Publishing, 2021
- [13] Antinyan, Vard: Pursuits of Automotive Software Variants Reduction, 2024
- [14] Metzger, Andreas ; Pohl, Klaus: Software product line engineering and variability management: achievements and challenges. In: Herbsleb, James; Dwyer, Matthew B. (Hrsg.): Future of Software Engineering Proceedings. New York, NY, USA : ACM, 2014, S. 70–84
- [15] Böckle, Günter ; Pohl, Klaus ; van der Linden, Frank: A Framework for Software Product Line Engineering. In: Pohl, Klaus; Böckle, Günter; van der Linden, Frank (Hrsg.): Software Product Line Engineering. Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, S. 19–38
- [16] Thum, Thomas ; Kastner, Christian ; Erdweg, Sebastian ; Siegmund, Norbert: Abstract Features in Feature Modeling. In: 2011 15th International Software Product Line Conference : IEEE, 2011, S. 191–200