



**Finanziert von der
Europäischen Union**
NextGenerationEU

Gefördert durch:



Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Verbundprojekt	
MBPLE4Mobility - Durchgehend modellbasierte Produktlinienentwicklung für Steuerungssysteme in der Fahrzeugtechnik;	
Teilvorhaben: Modellbasierte Produktlinienentwicklung für GOA2	
Schlussbericht zum Teilvorhaben	
Berichtszeitraum	01.07.2021 - 31.12.2024
Zuwendungsempfänger	Siemens Mobility GmbH (im Folgenden abgekürzt als SMO)
Förderkennzeichen	19I21025A
Laufzeit des Vorhabens	01.07.2021 – 31.12.2024
Leiter des Teilvorhabens	Andreas Pfeiffer SMO RS CR EN CCIP C2 1
Telefon	+49 162 1938000
Email	pfeiffer.andreas@siemens.com
Erstellungsdatum	11.07.2025



Dokumenteninformation
Autoren
<p>Andreas Pfeiffer – Siemens Mobility GmbH Dr. Daniel Homm - Siemens Mobility GmbH Wolfgang Rössler - Siemens Mobility GmbH George Kähler – Siemens Mobility GmbH</p>
Kontakt
<p>Andreas Pfeiffer Siemens Mobility GmbH SMO RS CR EN CCIP C2 1 Siemenspromenade 4 91058 Erlangen, Deutschland Tel.: +49 (9131) 17-51191 Mobil: +49 (162) 1938000 mailto:pfeiffer.andreas@siemens.com www.siemens.com</p>

Dokumentenhistorie

Version	Datum	Kommentar	Autoren
1.0	11.07.2025	Erstfassung	Pfeiffer, Rössler, Homm, Kähler

Inhalt

I.	Kurzfassung.....	8
1.	Aufgabenstellung	8
2.	Voraussetzungen, unter denen das Vorhaben durchgeführt wurde	9
3.	Planung und Ablauf des Vorhabens	9
4.	Wissenschaftlicher und technischer Stand zu Beginn des Vorhabens.....	11
i.	Verwendete Konstruktionen, Verfahren und Schutzrechte.....	11
ii.	Verwendete Fachliteratur und Informationsdienste	11
5.	Zusammenarbeit mit anderen Stellen.....	11
II.	Eingehende Darstellung.....	12
1.	Verwendung der Zuwendung und erzielte Ergebnisse im Einzelnen	12
1.1.	Methodik (AP 1 und AP 2)	12
1.2.	Evaluierung von MBSE mit PLE anhand eines Beispiels (AP3)	24
1.3.	Evaluierung von MBSE mit PLE anhand eines Beispiels (AP3)	24
1.3.1.	Evaluierung der Spezifikation von Feature-Realisierungen in MagicDraw.....	24
1.3.2.	Evaluierung der Spezifikation von UseCases in DOORS	26
1.3.3.	Evaluierung der Synchronisation von Feature-Bäumen und Feature-Konfigurationen zwischen DOORS und MagicDraw.....	28
1.3.4.	Evaluierung der Kontextdiagramme mit physikalischen Größen	29
1.3.5.	Fazit.....	31
1.4.	Entwicklung von Werkzeugen/Prozessen (AP4)	32
1.4.1.	Entwicklung der Visualisierung des Modeldrift in SonarQube	32
1.4.2.	Entwicklung eines Plugins zur Anzeige von „Variability Smells“	33
1.4.3.	Entwicklung der „Inline Blocks“ für komplexe Softwarekomponenten.....	34
1.4.4.	Entwicklung einer zyklenfreien Modellstruktur.....	37
1.4.5.	Vorbereitung Cloud-Infrastruktur für das Modellierungswerkzeug MagicDraw ...	38
1.5.	Exemplarischer Nachweis der Anwendbarkeit der MBPLE-Methoden am Beispiel der GoA2 Funktionsmodule der Fahrzeugsteuerung (AP5).....	41
1.5.1.	Grades of Automation (GoA)	41
1.5.2.	Aufgaben der Fahrzeugsteuerung eines Triebzuges.....	43
1.5.3.	Fahrzeuganforderungsspezifikation (FAnS)	43
1.5.4.	Fahrzeugarchitekturspezifikation (FARs)	44
1.5.5.	Ableitung einer MBPLE-Produktlinie	44
1.5.6.	Anpassungen der Systemarchitektur	46
1.5.7.	Erweiterung der MBPLE-Produktlinie um GoA2.....	46
1.5.8.	Umsetzung der ProfiNet-Anbindung des ATO-OB	49

1.5.9.	Test der GoA2-Funktion an einem Teststand.....	50
1.5.9.1.	Testspezifikation für manuelle Tests.....	51
1.5.9.2.	Teststandaufbau.....	52
1.5.9.3.	Inbetriebnahme Fahrzeugsteuerung.....	54
1.5.9.4.	Inbetriebnahme ATO-OB und ATO-TS.....	55
1.5.9.5.	Inbetriebnahme ETCS.....	55
1.5.9.6.	Fazit.....	56
1.6.	Eignungsprüfung (AP7).....	58
1.7.	Projektmanagement (AP8).....	59
1.7.1.	Zusammenarbeit im Konsortium.....	59
1.7.2.	Meilensteine.....	59
2.	Wichtigste Positionen des zahlenmäßigen Nachweises.....	60
3.	Notwendigkeit und Angemessenheit der geleisteten Arbeit.....	60
4.	Voraussichtlicher Nutzen.....	60
5.	Update Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen.....	61
6.	Erfolgte und geplante Veröffentlichungen des Ergebnisses.....	61

Abbildungsverzeichnis

Abbildung 1 Arbeitspakete aus der Vorhabensbeschreibung	10
Abbildung 2 Abhängigkeitsgraph in MagicDraw	14
Abbildung 3 Tabellen und Abhängigkeitsgraphen für die modellgestützte Ermittlung von Informationen für eine Auswirkungsanalyse	14
Abbildung 4 Geplante Gesamtarchitektur Streckensimulation	16
Abbildung 5 Serverbasierte Streckensimulation	17
Abbildung 6 Architektur serverbasierte Streckensimulation	18
Abbildung 7 Screenshot der Streckensimulation	19
Abbildung 8 Teststände, Vorderseite (links), Rückseite (rechts)	20
Abbildung 9 Beispiel für einen Gefährdungsbaum und Verknüpfung	21
Abbildung 10 Ausschnitt Feature Tree für GC1 Fahr-/Bremshebel	24
Abbildung 11 Verknüpfung GC1 Feature mit entsprechender Feature-Realization	25
Abbildung 12 Verknüpfung der Architekturelemente mit der Feature Realization	25
Abbildung 13 Ausleitung von Feature-Realisierungen in Dokumente	26
Abbildung 14 Beispielhafter GoA2 UseCase in DOORS	27
Abbildung 15 Feature-Baum in MagicDraw	28
Abbildung 16 Feature-Konfiguration und Feature-Bäume in DOORS	28
Abbildung 17 Bezug einer Anforderung zu einem Feature in DOORS	29
Abbildung 18 Features mit Verknüpfungen zu Anforderungen in DOORS	29
Abbildung 19 Exemplarisches Kontextdiagramm eines Teilsystems der Fahrzeugsteuerung	29
Abbildung 20 Schnittstellentypen	30
Abbildung 21 Exemplarische Hervorhebung von Schnittstellentypen im Kontextdiagramm	30
Abbildung 22 Darstellung des Modelldrifts in SonarQube	33
Abbildung 23 Anzeige von Variability Smells in MagicDraw	34
Abbildung 24 Exemplarische Zerlegung einer komplexen Softwarekomponente in Blöcke	34
Abbildung 25 Generierte Ein- und Ausgänge für Inline Block	35
Abbildung 26 Einbettung des Inline Blocks in den Code-Rumpf der SWC	35
Abbildung 27 Beispielhafte Ausleitung der Dokumentation einer komplexen SWC mit Inline Blocks	36
Abbildung 28 Ausschnitt der analysierten Modellstruktur	37
Abbildung 29 Ableitung von Modellstrukturrichtlinien	37
Abbildung 30 Grades of Automation	41
Abbildung 31 Trackside ATO (ATO-TS) und On-board ATO (ATO-OB)	42
Abbildung 32 Betriebsszenario GoA2	42

Abbildung 33 UseCase Diagram FAnS ATO GoA2	43
Abbildung 34 Feature-Baum zu GoA2	44
Abbildung 35 Testergebnisse aus dem Akzeptanztest der neuen Produktlinie	45
Abbildung 36 Feature-Baum zu GC3	47
Abbildung 37 Ausschnitt der Verhaltensübersicht des Teilsystems GC3	47
Abbildung 38 Ausschnitt der Struktursicht des Teilsystems GC3	48
Abbildung 39 GC3 Unit-Testergebnisse	49
Abbildung 40 Detailansicht Teststand für ATO-OB Integrationstests	49
Abbildung 41 ATO-OB Integrationsteststände	50
Abbildung 42 Logische Sicht auf die Anteile des Teststands	52
Abbildung 43 Aufbau Teststand Phase 1	53
Abbildung 44 Aufbau Teststand Phase 2 - unverdrahtet	54
Abbildung 45 Ausschnitt aus angepasstem Stromlaufplan für GoA2	54
Abbildung 46 ATO-OB	55
Abbildung 47 EVC	56
Abbildung 48 NVC	56
Abbildung 49 Aufgebauter GoA2 Teststand	57

Tabellenverzeichnis

Tabelle 1: Übersicht Meilensteine	10
Tabelle 2: Unbeabsichtigte UUID-Änderung bei Migration in Cloud-Infrastruktur	39
Tabelle 3: Eignungsprüfung SMO	58
Tabelle 4: Übersicht Meilensteine	59

Literaturverzeichnis

Es wurden keine Quellen für diesen Bericht benötigt. Die Arbeit erfolgte weitgehend mit Siemens-internen Know-How und Dokumenten.

I. Kurzfassung

1. Aufgabenstellung

Allgemeine Projektziele:

Ziel des Projektes MBPLE4Mobility war es, in den Bereichen Schienen- und Automobilverkehr Entwicklungsmethoden und Werkzeuge für modellbasierte Entwicklung und Produktlinienentwicklung zu optimieren, um so eine Reduktion von Projektkosten, Risiken und Projektlaufzeiten zu erreichen. Auf diese Weise lässt sich ein breites Angebot von zuverlässigen Lösungen im Bereich der Mobilität bereitstellen, wodurch zur Dekarbonisierung in Deutschland beigetragen werden kann.

Projektziele von Siemens Mobility:

- Das Projekt MBPLE4Mobility adressiert zentrale Herausforderungen und Chancen im Bereich des zukünftigen automatisierten Fahrens, insbesondere im schienengebundenen Regional- und Fernverkehr. Mit dem Einsatz von GoA2 (Grade of Automation 2) wird ein wesentlicher Beitrag zur Effizienzsteigerung und zur Reduktion des Energieverbrauchs geleistet.
- Ein zentrales Ziel des Projekts ist die Dekarbonisierung des Verkehrssektors. Dabei steht der Schienenpersonenverkehr im Fokus, da nur durch eine verstärkte Nutzung öffentlicher Verkehrsmittel – insbesondere des Zugverkehrs – eine nachhaltige Reduktion der CO₂-Emissionen realisierbar ist. Die Digitalisierung und Automatisierung bestehender Netze spielt hierbei eine Schlüsselrolle. Sie ermöglicht eine Erhöhung der Zugfolge, was zu kürzeren Taktzeiten, mehr Verbindungen in Randzeiten und einer höheren Flexibilität für Fahrgäste führt. Gleichzeitig profitieren Betreiber von einer besseren Anpassungsfähigkeit an Nachfrageänderungen und einer effizienteren Nutzung der Triebfahrzeuge.
- Ein weiterer Schwerpunkt liegt auf der Komplexitätsreduktion in den Entwicklungsprozessen. Durchgängige Entwicklungswerkzeuge und -verfahren sowie die frühzeitige Simulation und Optimierung von Abläufen über den gesamten Entwicklungs- und Produktlebenszyklus hinweg tragen wesentlich zur Effizienzsteigerung bei.
- Das aktuelle Vorhaben verfolgt einen ganzheitlichen Ansatz, um die Inbetriebnahme neuer Technologien zu beschleunigen und gleichzeitig bestehende Infrastrukturen zukunftsfähig zu machen.

Die Abkürzung MBPLE steht für: Model Based Product Line Engineering.

2. Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Projekt MBPLE4Mobility wurde mit Mitteln des Bundesministeriums für Wirtschaft und Klimaschutz gefördert. Es wurde gemeinschaftlich durchgeführt von fünf industriellen Partnern (davon drei KMU) und zwei akademischen Projektpartnern. Verbundpartner waren

- Siemens Mobility – SMO
- AVL mit den drei Tochtergesellschaften
 - AVL Deutschland – AVL-DE
 - AVL Schrick – AVL-S
 - AVL Software and Functions – AVL-SFR
- Codewerk – CW
- HMG
- MID
- Friedrich-Alexander-Universität Erlangen-Nürnberg – FAU
- Hochschule München – HM

Im März 2023 wurde die Firma pure-systems als assoziierter Partner aufgenommen.

Die Leitung des Verbundprojektes lag bei SMO. Das Projekt war zunächst für eine Laufzeit von drei Jahren ausgelegt und wurde am Ende um sechs Monate kostenneutral verlängert.

Die Laufzeit des Projektes überlappte sich mit der Corona-Pandemie.

3. Planung und Ablauf des Vorhabens

Die Arbeiten in MBPE4Mobility waren in acht Arbeitspakete (AP) aufgeteilt, wobei SMO an folgenden Arbeitspaketen beteiligt war:

- Arbeitspaket 1: Definition von integrierten Modellierungskonzepten entlang des V-Modells
- Arbeitspaket 2: Analyse, welche speziellen Aspekte der Entwicklung sich mit MBSE- und PLE-Verfahren unterstützen lassen
- Arbeitspaket 3: Evaluierung von MBSE mit PLE anhand eines Beispiels (Proof of Concept) – Use Case Ladefunktion
- Arbeitspaket 4: Entwicklung von Werkzeugen/Prozessen
- Arbeitspaket 6: Validierung der MBSE- und PLE-Ansätze am Beispiel von 3 Anwendungen im Bereich Automotive – Use Case Vehicle Control Unit
- Arbeitspaket 8: Projektmanagement

Eine detaillierte Darstellung der einzelnen Aufgaben der Arbeitspakete findet sich in der MBPLE4Mobility Vorhabensbeschreibung.

No	Arbeitspaketbeschreibung	
1	Definition von integrierten Modellierungskonzepten zur systematischen und strategischen Wiederverwendung von Entwicklungsartefakten (MBSE-PLE)	SMO AVL FAU HM HMG CW MID
2	Analyse wie spezielle Aspekte der Entwicklung mit MBSE-PLE unterstützt werden können: 1. Anforderungsmanagement 2. Review 3. Auswirkungsanalyse 4. Schaltungstechnik 5. Simulation 6. Safety-Engineering 7. Performance-Engineering 8. Security 9. Modellbasierte Interfaces 10. Model Factory	SMO + + AVL HMG + AVL MID + AVL MID FAU HM + HM + AVL FAU HM + AVL FAU + AVL FAU CW + MID + CW + HMG FAU
3	Evaluierung von MBSE mit PLE anhand eines einfachen Beispiels 1. Beispiel Schienenverkehr 2. Beispiel Automotive	SMO FAU HM HMG CW AVL HM
4	Entwicklung von Werkzeugen zur Unterstützung der entwickelten MBSE-PLE Methoden und Prozesse 1. Definition der Anforderungen an die Werkzeuge/Prozesse – Gemeinsamkeiten/Unterschiede feststellen 2. Entwicklung Werkzeuge/Prozesse für gemeinsame Aspekte 3. Anpassung auf individuelle Anforderungen SMO/Anwendungsfall 4. Anpassung auf individuelle Anforderungen CW/Anwendungsfall 5. Anpassung auf individuelle Anforderungen AVL/Anwendungsfall	SMO AVL FAU HM HMG CW SMO AVL FAU HM HMG MID SMO CW CW AVL
5	Exemplarischer Nachweis der Anwendbarkeit der MBSE-PLE Produktlinienentwicklung am Beispiel der GoA2 Funktionsmodule der Fahrzeugsteuerung	SMO FAU HMG CW
6	Validierung der MBSE-PLE Ansätze am Beispiel von 3 Anwendungen im Bereich Automotive 1. MBPLE für HV-Energiespeicher 2. MBPLE für die Ladefunktion und Interoperabilität 3. MBPLE für Fahrerassistenzfunktionen	AVL HM
7	Überprüfung der Eignung der Verfahren zur Steigerung der Innovation und zur Reduktion der Entwicklungskosten, Entwicklungsrisiken und der Projektdurchlaufzeiten	SMO AVL FAU HM HMG CW
8	Projektmanagement	SMO AVL FAU HM HMG CW MID

Abbildung 1 Arbeitspakete aus der Vorhabensbeschreibung

Ursprünglich wurde das Projekt mit sechs Meilensteinen über eine Laufzeit von 36 Monaten geplant. Nach der bewilligten kostenneutralen Verlängerung um sechs Monate passte man die Termine für die Meilensteine 5 und 6 im Jahr 2024 an die verlängerte Laufzeit des Projektes an.

Meilenstein	Beschreibung	Termin
M1	Wesentlich Konzepte wurden erarbeitet. Proof of Concept und Werkzeugentwicklung können gestartet werden.	Monat 6
M2	Proof of Concept ist erbracht.	Monat 12
M3	Werkzeuge und methodische Aspekte sind bereit für den Beginn des Nachweises.	Monat 18
M4	Konzepte sind überarbeitet.	Monat 30
M5	Start der Eignungsprüfung.	Monat 36
M6	Die Arbeiten des Projektes sind abgeschlossen. Die definierten Ziele für die Nachweise für die Erreichung der TRL-Ziele wurden erbracht (TRL = Technology Readiness Level)	Monat 42

Tabelle 1: Übersicht Meilensteine

4. Wissenschaftlicher und technischer Stand zu Beginn des Vorhabens

i. Verwendete Konstruktionen, Verfahren und Schutzrechte

Beim Start des Forschungsprojekts waren sowohl die MBSE- als auch die PLE Methodiken noch nicht weit entwickelt. Bereits vor Projektstart war der Vorteil der Weiterentwicklung dieser Methoden und der sich draus ergebende wirtschaftliche Vorteil absehbar. Zu Beginn des Vorhabens lagen noch keine Schutzrechte vor, auf die aufgebaut werden konnte.

Der wissenschaftliche und technische Stand, an den angeknüpft wurde ergab sich weitgehend aus Siemens-internen Dokumenten und -internem Know-How, welches für vorgehende kommerzielle Projekte aufgebaut wurde.

ii. Verwendete Fachliteratur und Informationsdienste

Im Rahmen dieser Arbeit wurde auf externe Fachliteratur weitgehend verzichtet. Die Inhalte basieren primär auf internem Wissen, internen Entwicklungsunterlagen sowie projektspezifischen Dokumentationen der Siemens Mobility GmbH.

5. Zusammenarbeit mit anderen Stellen

Siemens Mobility arbeitete mit den in Abschnitt I.2 erwähnten Projektpartnern und Assoziierten Partnern zusammen. Es gab keine weitere Zusammenarbeit mit anderen Stellen.

II. Eingehende Darstellung

1. Verwendung der Zuwendung und erzielte Ergebnisse im Einzelnen

Im Folgenden werden die erzielten Ergebnisse pro Arbeitspaket dargestellt. Anschließend wird auf die Erreichung der Meilensteine eingegangen, danach erfolgt eine Darstellung der Ergebnisse im Hinblick auf die Erreichung der Forschungsziele bzw. der Technology Readiness Level.

Einzelergebnisse nach Arbeitspaketen

In den folgenden Kapiteln sind die Einzelergebnisse nach Arbeitspaketen sortiert.

1.1. Methodik (AP 1 und AP 2)

Requirements Engineering (AP 1.1 und AP 2.1)

Feature Trees sind essenziell, um Variabilität im Anforderungsmanagement zu meistern. Eine Methodik wurde entwickelt, die Folgendes berücksichtigt:

- Variabilität kann sowohl durch Kunden als auch interne technische Lösungen getrieben werden
- Vertrieb und Engineering benötigen unterschiedliche Sichten (z.B. bzgl. Granularität) auf die Sichten
- Es existieren verschiedene Arten von Variabilität, von einfachen Parametern bis hin zu komplexeren Strukturen
- Variabilität kann zu verschiedenen Zeitpunkten während der Entwicklung oder des Betriebs festgelegt werden
- Anforderungen müssen mit den Features verknüpft werden

Use Cases eignen sich besonders zur Spezifikation von Abläufen, zudem können textuelle Anforderungen damit ergänzt werden. Die entwickelte Methodik erlaubt sowohl eine textuelle als auch formale Darstellung in Form von Modellen. Dadurch konnten einzelne Elemente der Use Cases mit den Features Trees verknüpft werden.

Eine besondere Herausforderung für SMO ist, dass Kunden Züge nicht nach Features kaufen, sondern über öffentliche Ausschreibungen mit CbC-Kommentierung. Dafür wurde eine Methodik und ein Tool entwickelt, das CbC-Kommentierung basierend auf Feature-Modellen ermöglicht und so die Lücke zwischen Ausschreibungen und Produktlinienentwicklung geschlossen hat.

Einige der Konzepte wurden gemeinsam mit der Hochschule München auf dem Tag des Systems Engineering in Paderborn und Würzburg vorgestellt.

Model-Review (AP 2.2)

Reviews beziehen sich für gewöhnlich auf Dokumente und/oder den implementierten Code. Durch den Einsatz von MBSE und damit durch die Verwendung von Modellen ergibt sich eine weitere Möglichkeit des Reviews: Das Modellreview.

Im Rahmen des Arbeitspaketes wurde ermittelt, welche Vor- und Nachteile ein Modellreview gegenüber einem Dokumentenreview bietet. Das betrachtete Modell dient dabei als Grundlage für die automatische Generierung des Dokuments.

Die Untersuchung ergab, dass sich über das Modellreview das Tracing besser prüfen lässt als über das Dokumentenreview. Das liegt daran, dass die Verknüpfungen zwischen Modellelementen in der Regel über eigene Modellelemente (Beziehungen) im Modell hinterlegt sind. Sie lassen sich leichter nachverfolgen, da Modellierungswerkzeuge bequeme Sprungfunktionen anbieten. In einem Dokument existieren solche Sprungfunktionen nicht immer. Es lassen sich zwar Verweise einfügen, allerdings ist dies nicht für alle Artefakte möglich, insbesondere wenn sie nicht Bestandteil des in Review befindlichen Dokuments sind.

Ein Nachteil des Modellreviews ist jedoch die Zugänglichkeit zum Modellierungswerkzeug. Ebenso beinhaltet ein Modell viele Attribute, die nicht im Fokus eines Reviews sind. Allerdings bieten sie die Möglichkeit, eine angepasste Sicht auf die Modellelemente zu geben. Je nach Rolle eines Reviewers ist eine Anpassung des Reviewumfangs möglich. In einem Dokument ist solche eine Anpassung in der Regel nur über eine vordefinierte Dokumentenstruktur möglich.

Abschließend wurde ein Vorgehen spezifiziert, mit dem sich Rollen und Umfänge in einem Modellreview festlegen lassen. Ein dazugehöriges Meta-Modell für den Reviewvorgang selbst wurde spezifiziert.

Auswirkungsanalyse (AP 2.3)

Änderungen in einer Fahrzeugsteuerungsapplikation müssen genauestens analysiert und dokumentiert werden. Eine Auswirkungsanalyse (Impact Analysis) ist ein Dokument, das solch eine Änderung detailliert beschreibt.

Im Rahmen von AP 2.3 wurde untersucht, wie durch MBSE das Schreiben einer Auswirkungsanalyse unterstützt werden kann. Es wurde festgestellt, dass sich folgende Informationen für die von einer Änderung betroffenen Entwicklungsartefakte auf einfache Art und Weise mit der Hilfe von Modellen ermitteln lässt:

- Die Namen der betroffenen Teilsysteme der Fahrzeugsteuerung
- Die Namen von betroffenen Funktionen
- Die Namen der betroffenen Softwarekomponenten
- Die IDs der betroffenen Anforderungen
- Die Namen der betroffenen Schnittstellen
- Die Bezeichner der betroffenen Safety-Elemente

Für die Ermittlung der Informationen wurden entsprechende Tabellen und Abhängigkeitsgraphen (siehe Abbildung 2) in MagicDraw erstellt und konfiguriert (siehe Abbildung 3).

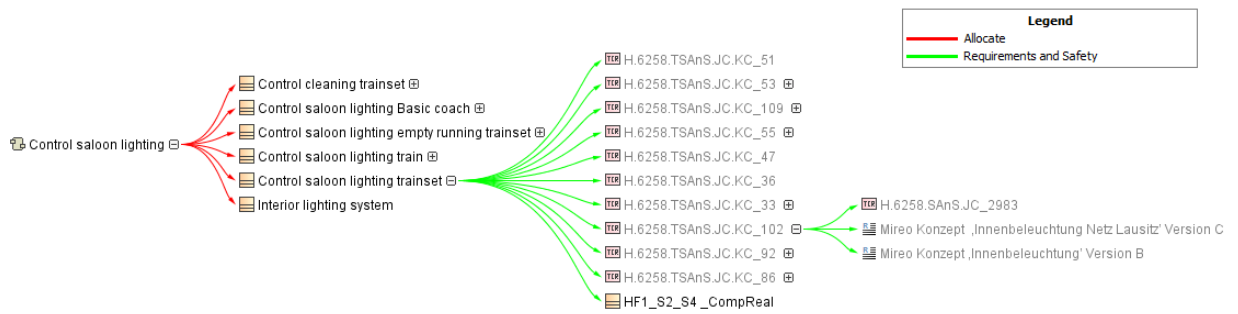


Abbildung 2 Abhängigkeitsgraph in MagicDraw

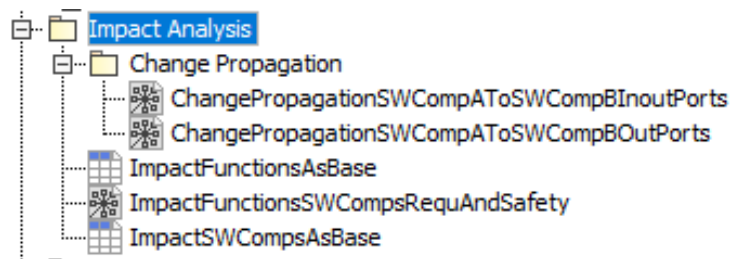


Abbildung 3 Tabellen und Abhängigkeitsgraphen für die modellgestützte Ermittlung von Informationen für eine Auswirkungsanalyse

Das Vorgehen wurde anhand zweier Änderungen erfolgreich pilotiert und anschließend für alle Modelle der Fahrzeugsteuerung ausgerollt.

Hardware-Entwurf / Schaltungstechnik (AP 1.3 und AP 2.4)

Eine umfassende Untersuchung von Methoden zur Verbindung von Schaltungstechnik und Feature-Modellen hat keine praktikable Lösung gefunden. Gründe dafür waren insbesondere Einschränkungen in den etablierten Schaltungstechnik-Tools und Prozessen. Eine tiefgehende Evaluierung hatte massive Änderungen in den Tools und Prozessen erfordert, die im Rahmen des Projektes nicht möglich gewesen wären. Die gewonnenen Erkenntnisse, welche Arten von Variabilität bei der Schaltungstechnik eine besonders große Rolle spielen und wie diese realisiert werden, können aber bei einer zukünftigen Neugestaltung der Prozesse (z.B. bei einem Toolwechsel) von großem Nutzen sein.

Simulation (AP 2.5)

Erstellung von virtuellen Fahrzeugen:

Dieser Punkt wurde vollständig umgesetzt. Es wurde ein Konzept für die MBPLE Test- und Entwicklungsumgebung entwickelt.

Simulation der Zugkommunikation

Im Rahmen des Projekts wurde ein Konzept für die Simulation der Zugkommunikation umgesetzt. Es wurde ein Workflow erstellt, aus dem die Schnittstellen direkt im Simulationswerkzeug importiert werden können. In Teilen wurde es auch im Rahmen des MBPLE Projekts für die Restfahrzeug-Simulation der Test- und Entwicklungsumgebung umgesetzt und eingesetzt

TrainSim ATP

Dieser Punkt wurde voll umgesetzt und war eines der Arbeitsschwerpunkte des Projekts. Der erste Teil bestand darin vorhandene Simulationstechnologien in den unterschiedlichen Siemens Mobility Bereichen zu untersuchen. Zum Beginn des Projektes existierten zwei völlig unterschiedliche und Inkompatible Systeme. Auf Seite des Gesamtfahrzeugs ein Produkt namens TrainSim und auf Seite der Zugsicherungssysteme ein System namens TUS. Nach ersten Untersuchungen wurden bestimmt, dass es am zielführendsten ist auf eine TrainSim-Umgebung für das komplette Fahrzeug aufzusetzen und für die Streckenausrüstungsanteile Simulationen von den Kollegen der RI einzusetzen, da diese direkt aus der Domäne kommen. Auf die Onboardgeräte der Zugsicherung bezogen bedeutet das:

- Konventionelle Schaltungstechnik (Störschalter, Schleifen, Wegimpulsgeber,...) werden mittels TrainSim Technologie simuliert
- Restbussimulation mittels TrainSim Technologie
- GPS Simulation mittels TrainSim Technologie
- Radarsimulation wird neu entwickelt, um es bestmöglich in die Bewegungssimulation integrieren zu können.
- Für den Balisenkanal kommt eine Technologie namens „Balsim“ von RI zum Einsatz. Hierfür wurde ein Plugin für das TrackSim Custom Device entwickelt, um es in die Veristand Welt von TrainSim zu integrieren
- Für die RBC Kommunikation kommt eine Technologie namens STAC Mobile Adapter ebenfalls von RI zum Einsatz. Diese Adapter kommunizieren mit dazu passenden virtuellen Maschinen für die RBC Strecken-Zentrale-Simulation

Ein weiteres großes Thema war die Entwicklung der Schnittstelle für die Streckensimulation in Bezug auf die Anforderungen der Zugsicherung. Hierbei ging es darum die Engineeringdaten der Strecke automatisiert zu verarbeiten und daraus die Inputdaten für die serverbasierte Streckensimulation zu erzeugen und automatisiert weiterzuverarbeiten.

Das Thema wurde in Teilen auch im Rahmen einer sehr guten Bachelorarbeit zum Thema „Konzipierung und Realisierung eines Hardware-In-The-Loop-Testsystems für den European Vital Computer (EVC) eines Schienenfahrzeugs basierend auf einer existierenden Restfahrzeugsimulation“ umgesetzt. Das komplette Konzept wurde später auch als Test- und Entwicklungsumgebung umgesetzt. Dabei hat sich das Konzept bewährt. Nun ist der Einsatz in mehreren Projekten geplant.

Streckensimulation

Die Streckensimulation ist ein völlig neues Konzept, das lange als Idee in der Schublade lag. Aber erst ein großes Forschungsprojekt hat die Umsetzung erst ermöglicht. Mit der Umsetzung wurde direkt zu Projektbeginn begonnen.

Als Architektur wurde eine serverbasierte Lösung gewählt. Diese Lösung ermöglicht flexible Erweiterungen und auch den Einsatz von mehreren Fahrzeugen. Dies ist auch Projekt und Fahrzeugtyp übergreifend möglich. Das Ziel ist ein massenhafter Betrieb von virtuellen Fahrzeugen (Test- und Entwicklungsumgebungen) zu ermöglichen.

Unten gezeigt ist die gesamte geplante Gesamtarchitektur.

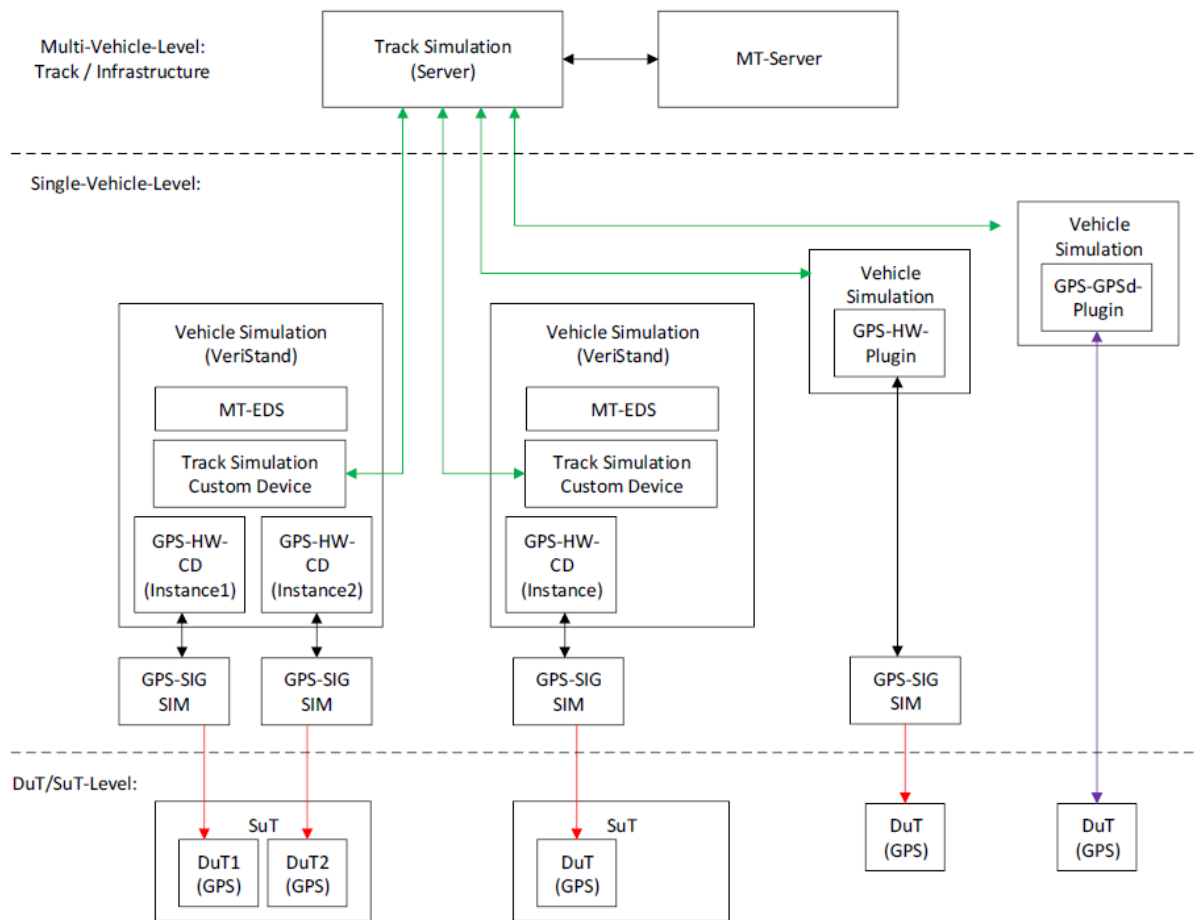


Abbildung 4 Geplante Gesamtarchitektur Streckensimulation

Hier können verschiedene Arten von Test- und Entwicklungsumgebungen an die serverbasierte Streckensimulation angebunden werden. Dabei ist egal, ob ein Testobjekt ein Stück Hardware (bzw. mehrere Geräte) ist, oder einfach nur ein virtualisiertes Gerät bzw. eine zu testende Applikation ist.

In der folgenden Darstellung wird die Kommunikation zwischen den unterschiedlichen Arten von Test- und Entwicklungsumgebungen und der serverbasierten Streckensimulation mittels gleicher Schnittstellen genauer gezeigt.

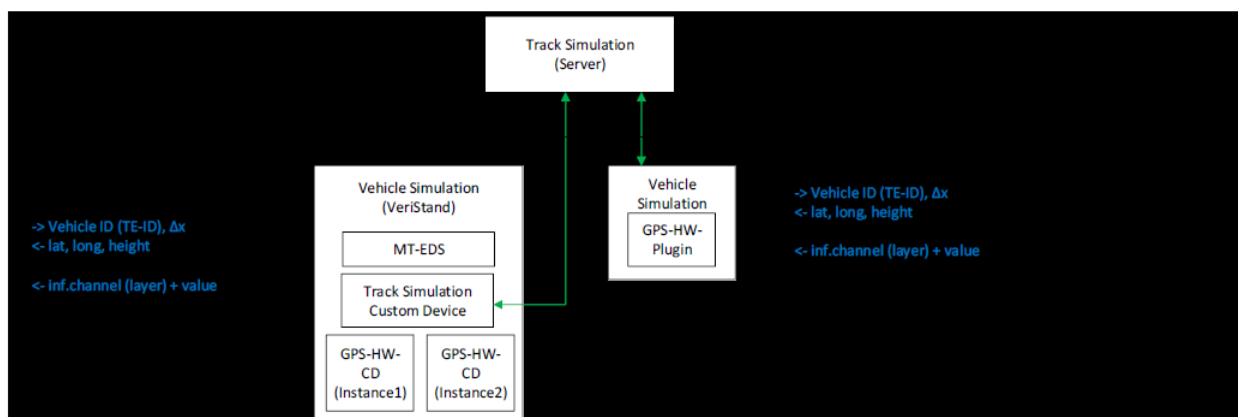


Abbildung 5 Serverbasierte Streckensimulation

Bei der Umsetzung wurde festgestellt, dass die nötigen Latenzen mit einem zentralen Server nicht eingehalten werden können. Deswegen wurde ein zusätzlicher, laborlokaler Server entwickelt, der die schnelle Kommunikation übernimmt. Gleichzeitig fungiert dieser Server als Gateway eines Labors zum zentralen Streckensimulationsserver. Dieser logische Server besteht wiederum aus mehreren einzelnen Applikationen, die jeweils in eigenen Containern laufen. Dies erhöht die Verfügbarkeit der Gesamtlösung und erlaubt flexible Hostingkonzepte. Nachfolgend die Umgesetzte Architektur.

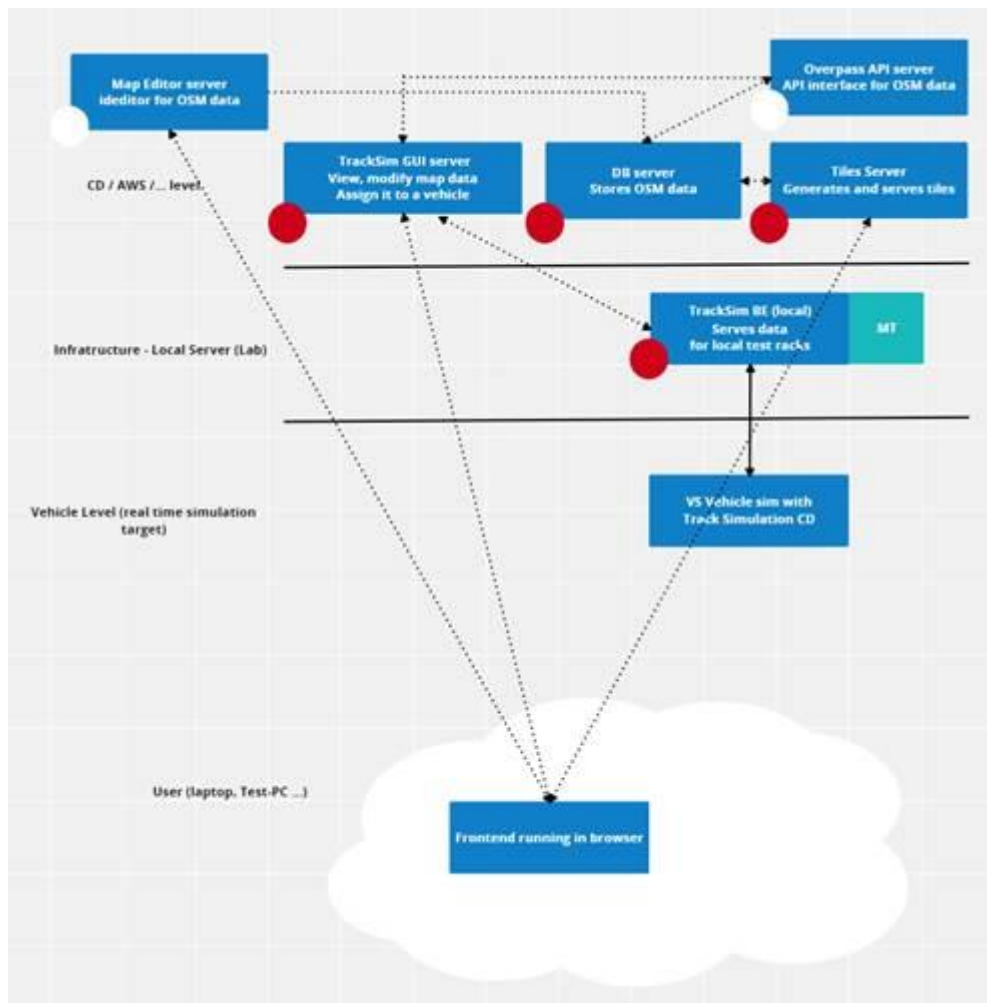


Abbildung 6 Architektur serverbasierte Streckensimulation

In Summe war das Thema serverbasierte Streckensimulation ein voller Erfolg, durch den frühen Beginn und frühe Erfolge ist das Thema bereits produktiver Einsatz in nachfolgenden Projekten

Sehr gute beobachten wir eine gute Skalierbarkeit und eine volle Cloud-kompatibilität. Dadurch dass die Simulation flexibel erweiterbar ist und durch die konsequente Verwendung webbasierte Frameworks ergibt sich, ganze neue Welt an Möglichkeiten der Weiterentwicklung.



Abbildung 7 Screenshot der Streckensimulation

Virtuelle Fahrzeuge (virtueller Laborbetrieb)

Dieses Arbeitspaket kann in zwei große Teile unterteilt werden. Zum einen die Erstellung eines virtuellen Fahrzeugs auf Basis der Entwicklungen im MBPLE Projekt mit dem primären Einsatzziel im TrainSim ATP HiL für die Entwicklung, Integration und Test von Zugsicherungssystemen mit ATO zusammen mit der Fahrzeugleittechnik (Fahrzeugsteuerung). Zum anderen ein großes Arbeitspaket rund um Cloud-Umgebung und Virtualisierung mit dem Ziel virtuelle Fahrzeuge in komplett virtuellen Umgebungen betreiben zu können.

Zum Anteil TrainSim Mireo MBPLE:

Nachdem eine Fahrzeugbasis (Mireo Smart) festgelegt wurde, wurde auf Basis der Entwicklungsunterlagen ein virtuelles Fahrzeug aus diesen entwickelt. Dazu wurden nach TrainSim Konzept die Build-Pipelines aufgesetzt. Dies ist eine Siemens Mobility Rolling stock Standardlösung. Dies sollte die Basis für die im MBPLE zu entwickelnden Technologien und Prozesse sein. In diese Basis wurden die GoA2 Anpassungen gemacht. Dies betraf insb. die Integration der Zugsicherungssysteme inkl. ATO. Hier mussten die E-Montageunterlagen des Fahrzeugs (Stromlaufpläne) und Schnittstellen der Subsysteme (GSDs) importiert werden. Die Subsysteme wurden in ihrem Verhalten modelliert, damit die Fahrzeug-Leittechniksoftware (Fahrzeugsteuerungssoftware) eine Testbasis für ihre Entwicklung hatte.

Dieses virtuelle Fahrzeug wurde dann in zwei Ausprägungen im Labor betrieben. Die erste Variante war für die Leittechnikentwicklung (TrainSim MBPLE TC), ein HiL System mit echten Fahrzeug-Leittechnikkomponenten (Fahrzeugsteuerung) und kompletter Restfahrzeugsimulation. Die zweite Variante war für die Entwicklung, Integration und Test von Fahrzeugleittechnik mit Betriebsleittechnik (Zugsicherung). Diese Variante nennen wir TrainSim MBPLE ATP.

Nachfolgend sind Fotografien der Test und Entwicklungsumgebung. Das große Rack Links beinhaltet die Betriebsleittechnik (Zugsicherung) inkl. der dafür benötigten Simulationshardware. Das rechte Rack beinhaltet die Fahrzeugetechnik (Fahrzeugsteuerung) inkl. der dafür benötigten Simulationshardware. Das rechte Bild zeigt die Rückseite des großen Racks.

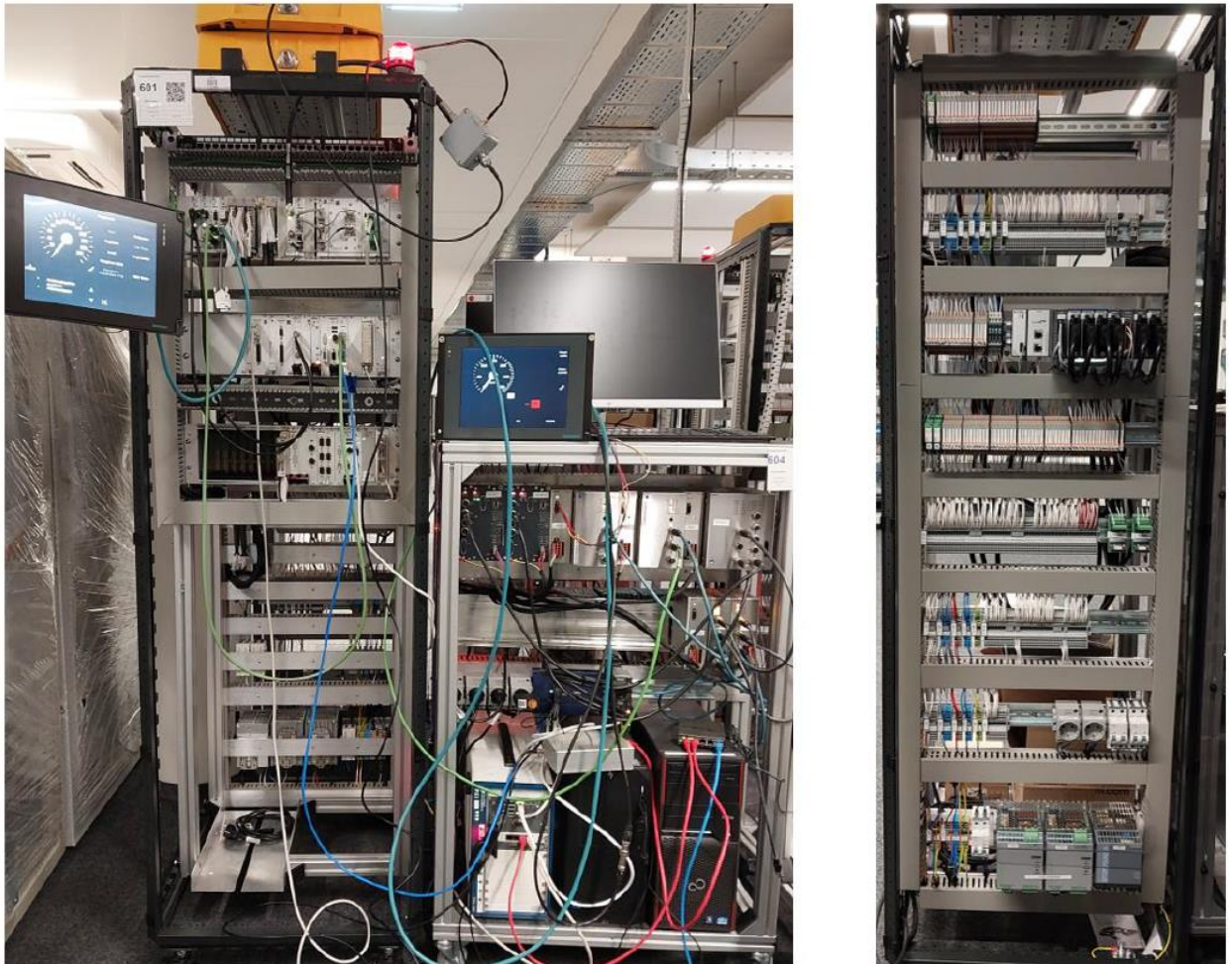


Abbildung 8 Teststände, Vorderseite (links), Rückseite (rechts)

Dank TrainSim Konzept können beide Racks allein oder zusammen je nach Bedarf benutzt werden. Genau hierfür sind die beiden Simulationsprojekten TrainSim MBPLE TC und TrainSim MBPLE ATP.

Da dieser erste große Block sehr viel mehr Ressourcen in Anspruch genommen hat als geplant, wurde im zweiten Block weniger erreicht als vorgenommen. Es wurden alle Themen bearbeitet, aber keines komplett zu Ende geführt:

- TrainSim Cloud: Zum Ende des MBPLE Projekts existiert eine fertige Architektur und eine erste Demoversion. Zum Abschluss der Arbeiten, wie im Rahmen des Projekts vorgesehen fehlen noch rund ein Jahr Entwicklungszeit. Das Projekt wird fortgeführt und hat mittlerweile einen sehr guten Stand.
- Virtualisierung SPCS: Nicht durch SIM in Bearbeitung. Wurde von einem Konsortialpartner übernommen

- Virtuelle (subsystem-)Steuergeräte: Konzept auf Basis co-simulations Standards entwickelt. Erste (SMO RS EN interne) Steuergeräte in der Umsetzung. Abschluss nach MBPLE Projekt.

Safety-Engineering (AP 2.6)

Für den Aspekt Safety Engineering wurden vor allem zwei Punkte detailliert bearbeitet:

- Durch Variabilität in den Systemen steigen auch die Anforderungen in Bezug auf die Validierung und Begutachtung. Um die dadurch entstehenden Aufwände im Rahmen zu halten, wurde ein Verfahren zur Trennung von sicherheitsrelevanten und nicht-sicherheitsrelevanten SW-Anteilen auf einer einzelnen Steuerung entwickelt. Dies basiert auf Eigenschaften, welchem vom Basissystem erbracht werden müssen, und zusätzlichen Checklisten. Auch wenn sich diese Methode als wirtschaftlich nicht sinnvoll herausgestellt hat, liefert sie dennoch wertvolle Hinweise für eine Weiterentwicklung des Systems.
- Für den Sicherheitsnachweis kommen vor allem Gefährdungsbäume zum Einsatz. Die Methodik wurde dahingehend erweitert, dass nun auch Varianten modelliert werden können. Die Verknüpfung mit Feature Trees erfolgt dabei nach einer ähnlichen Methode wie bei anderen Elementen der Systemarchitektur (siehe Abbildung 9).

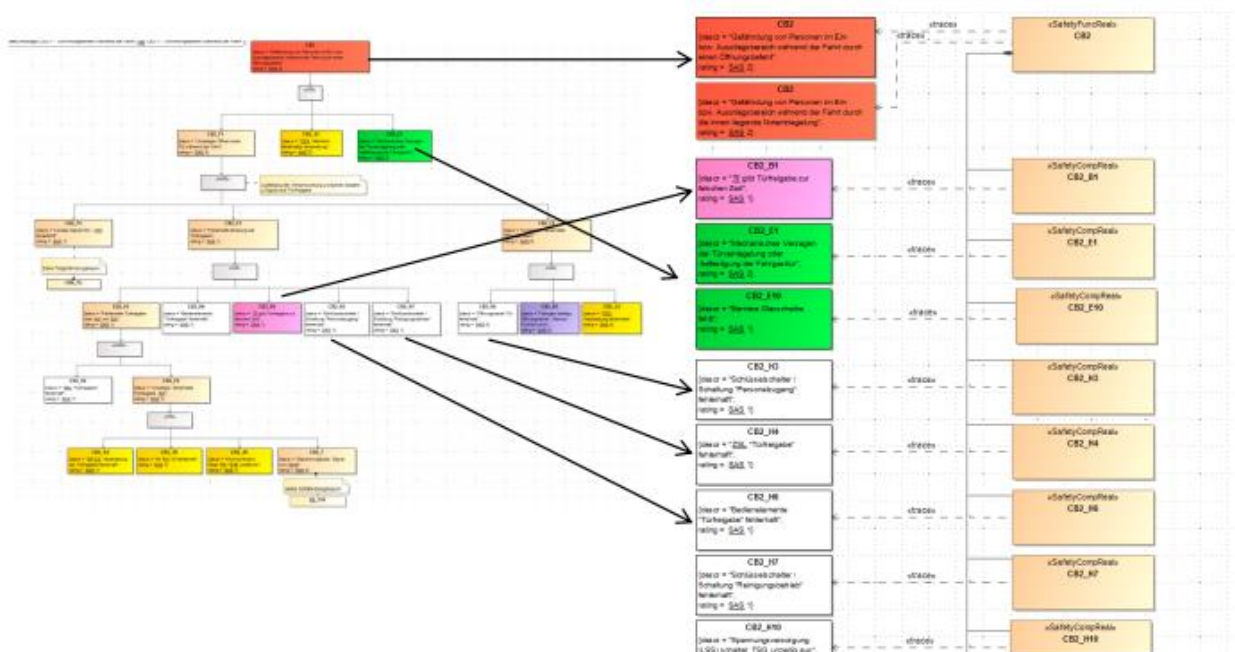


Abbildung 9 Beispiel für einen Gefährdungsbaum und Verknüpfung

Performance Engineering (AP 2.7)

Im Rahmen einer Masterarbeit wurde das Performance Engineering für End-to-End Reaktionszeiten in komplexen verteilten Systemen ausführlich untersucht. Dies umfasste insbesondere:

- Recherche zu Methoden der Performance-Modellierung
- Entwicklung einer angepassten Methodik zur Modellierung von End-to-End-Reaktionszeiten für SMO
- Simulation dieser Reaktionszeiten mit OMNeT++ basierend auf der Modellierung,
- Validierung der Simulationsresultate durch Messungen am realen System

Eine umfassende Analyse zur Berechnung und Budgetierung der Last von Steuerungssystemen wurde nicht durchgeführt. Erste Untersuchungen deuteten darauf hin, dass eine grobe Abschätzung der Auslastung sowie eine fortlaufende Überwachung während der Integrationstests ausreichend sind.

Security (AP 2.8)

Im Rahmen des Arbeitspaketes 2.8 Security wurde eine umfassende Sicherheitsanalyse der Applikation MagicDraw 19 nach ISO 27001 durchgeführt. Die Analyse basierte auf einem detaillierten Fragebogen und untersuchte die drei fundamentalen Säulen der IT-Sicherheit:

1. Vertraulichkeit (Confidentiality):
Die Untersuchung der Vertraulichkeitsaspekte konzentrierte sich auf den Schutz und die Zugänglichkeit der Modelldaten. Folgende Maßnahmen wurden implementiert:
 - Einführung des Need-to-Know-Prinzips für die Zugriffsverwaltung
 - Evaluation und Sicherstellung der verschlüsselten Kommunikation zwischen:
 - MagicDraw 19 (Client)
 - MagicDraw Teamwork Server
 - MagicDraw Teamwork Cloud
2. Integrität (Integrity):
Die Integritätsprüfung fokussierte sich auf den Schutz der Daten vor unberechtigten Zugriffen und Manipulationen. Als zentrale Maßnahme wurde festgelegt:
 - Restriktive Vergabe von Schreibrechten nach dem Bedarfsprinzip
 - Implementierung eines kontrollierten Zugriffsmanagements
3. Verfügbarkeit (Availability):
Die Verfügbarkeit befasst sich mit der Zugänglichkeit des Systems und möglichen Ausfallszenarien. Hierzu wurden folgende Maßnahmen definiert:
 - Monitoring der Systemverfügbarkeit
 - Implementierung von Backup-Strategien
 - Maßnahmen zur Ausfallsicherheit

Modellbasierte Interfaces / Generalisierter Kommunikationsadapter (AP 2.9)

Im Rahmen des Arbeitspaketes 2.9 wurde der Konsortialpartner Codewerk unterstützt. Es wurden geeignete Use Cases für einen generalisierten Kommunikationsadapter im Umfeld des Zugsicherungssystems European Train Control System (ETCS) besprochen. Ebenso wurden notwendige Informationen für die Umsetzung einer Virtualisierungsumgebung zur Verfügung gestellt.

Model Factory (AP 2.10)

Im Rahmen des Arbeitspaketes 2.10 wurde der Konsortialpartner HMG unterstützt. Es wurde ein Workshop mit Experten aus dem Umfeld der Bedienung eines Triebzuges organisiert, um die Methode des Core Modelling zu evaluieren.

Ebenso wurde im Themenbereich des modellbasierten Tests unterstützt: Es gab einen regelmäßigen Austausch und es wurden Dokumentationen und Modelle zur Verfügung gestellt, um das Vorgehen zu evaluieren.

Test (AP 1.4 – AP 1.6 und AP 2.11)

Ein entsprechendes Konzept wurde ausgearbeitet. Dabei wurde explizit die Softwarestruktur der Fahrzeugsteuerung betrachtet, und unterteilt, in Software-Komponenten, Teilsysteme und das System Fahrzeugsteuerung (TCS). Die Teststufen wurden dann wie folgt definiert:

- Software component test -> testet die Komponenten
- Part system software integration test -> testet die Integration von SW-Komponenten
- Part system integration test -> testet die Integration von HW- und SW-Komponenten
- Part system system test -> betrachtet das Teilsystem als Blackbox
- TCS integration test -> testet die Integration von Teilsystemen zum System
- TCS system test -> testet das System

Alle Stufen werden entsprechend beschrieben, und folgende Punkte betrachtet:

- Test object
- Test objective
- Test basis
- Test environment
- Entry criteria
- Test process
- Exit criteria
- Responsible team/role

Zusammenfassend lässt sich sagen, dass durch den gestuften Testprozess bereits getestete Elemente bei einer Verwendung in einem neuen Projekt nicht erneut getestet werden müssen, wenn sich innerhalb dieser Elemente nichts ändert. Es ist dann lediglich die Integration in das nächstgrößere System zu testen. Somit ist es möglich den Testaufwand zu reduzieren, ohne dabei Einbußen bei der Testabdeckung zu generieren.

1.2. Evaluierung von MBSE mit PLE anhand eines Beispiels (AP3)

Für den "Proof of Concept" (PoC) wurden die spezifizierten Methoden für MBSE und PLE eingesetzt und auf deren Tauglichkeit geprüft.

Darüber hinaus wurden folgende Themenbereiche bereits evaluiert:

- 1) Spezifikation von UseCases im Requirements Management Tool DOORS statt im Modellierungstool MagicDraw
- 2) Synchronisation von Feature-Bäumen und -Konfigurationen zwischen dem Requirements Management Tool DOORS und dem Modellierungstool MagicDraw
- 3) Kontextdiagramme mit physikalischen Größen

Im Folgenden wird jeder Bereich näher betrachtet.

1.3. Evaluierung von MBSE mit PLE anhand eines Beispiels (AP3)

1.3.1. Evaluierung der Spezifikation von Feature-Realisierungen in MagicDraw

Für den "Proof of Concept" (PoC) wurden die aus AP1 und AP2 spezifizierten Methoden für die Modellierung und Beherrschung von Variabilität prototypisch bei der Entwicklung eines Teilsystems der Fahrzeugsteuerung angewendet, um sie auf ihre Tauglichkeit zu prüfen.

Es wurde das Teilsystem GC1 ausgewählt. Es ist dafür verantwortlich, die Fahr- und Bremsbedienhandlungen des Triebfahrzeugführers (Tf) entgegenzunehmen und in entsprechende Sollwerte umzuwandeln, die andere Teilsysteme einlesen und verarbeiten. Das Bedienelement kann dabei unterschiedlich ausgeprägt sein kann: Es ist möglich zwei Bedienelemente zu haben, eines für Fahren und eines für Bremsen, oder ein kombiniertes Bedienelement, das beide Funktionen abdeckt. Gemäß der Spezifikation aus AP1 und AP2 wurde ein Feature Tree erstellt, der diese Variabilität beinhaltet. Abbildung 10 zeigt einen Ausschnitt aus dem Feature Tree von GC1, der die beschriebene Variabilität darstellt.

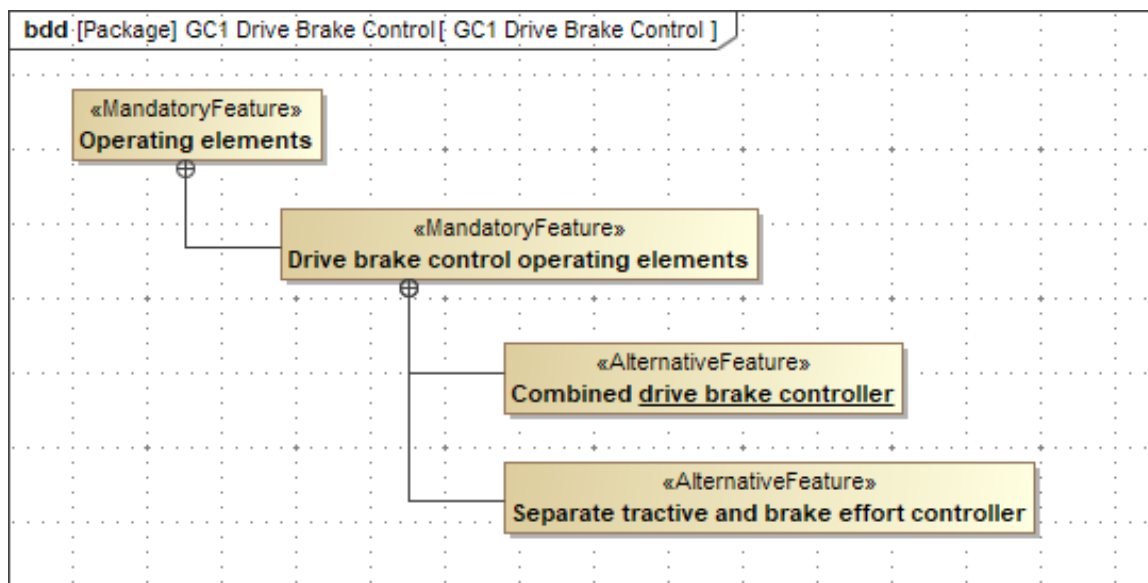


Abbildung 10 Ausschnitt Feature Tree für GC1 Fahr-/Bremshebel

Anschließend wurde beim Design der Architektur von GC1 eine Feature Realization für den kombinierten Fahr-/Bremshebel definiert und mit dem Feature in Bezug gesetzt (siehe Abbildung 11).

2.10 Variability

2.10.1 Train length counter (Optional)

After actuating the Sifa button via "double click" (twice activated within 1.5s), the distance travelled is measured and informed by means of an acoustic signal in the driver's cab (train length counter) after reaching the train length

Binding Time:	<u>TrainsetType</u>
Pre Decision Time:	N/A
Realization Method:	Configure at Engineering-Time

Owned feature realization elements:

Name	Type	SysML Type
[N/A]	<u>Train Length Counter</u>	<u>SwComponentInstance</u>
<u>Train Length Counter</u>	[N/A]	<u>SwComponent</u>
<u>Train Length Counter Command</u>	[N/A]	Interface Block
<u>Train Length Counter Response</u>	[N/A]	Interface Block
<u>TrainLengthCounter</u>	[N/A]	Package
<u>request voice output train length counter</u>	Request voice output (JC3)	S7ExternalProxyPort

Abbildung 13 Ausleitung von Feature-Realisierungen in Dokumente

Darüber hinaus stellen die Beziehungen zwischen Feature Realizations und den Architekturelementen eine wesentliche Hilfe im Modellierungswerkzeug MagicDraw dar, wenn neue Produktlinien abgeleitet („branchen“) oder Informationen zwischen Produktlinien ausgetauscht („mergen“) werden sollen.

Die spezifizierten Methoden für die Beherrschung und Modellierung der Variabilität aus AP1 und AP2 konnten damit erfolgreich angewendet und deren Tauglichkeit bestätigt werden.

1.3.2. Evaluierung der Spezifikation von UseCases in DOORS

Eine Spezifikation von textuellen Anforderungen und UseCases in unterschiedlichen Tools führt zu Konsistenzproblemen:

- Es gibt ein erhöhtes Risiko von Dopplungen bei den Anforderungen
- Textuelle Anforderungen und UseCases können sich widersprechen

Um dieses Problem zu umgehen, wurde im Rahmen von AP1 festgelegt, wie UseCases im Requirements Management Tool DOORS spezifiziert werden können. Im Rahmen von AP3 wurde ein UseCase zur GoA2 Funktion exemplarisch in DOORS spezifiziert und konnte mittels der umgesetzten Import-Funktion aus AP4 in MagicDraw als UseCase Modellobjekt importiert werden.

ID	Requirement	ID Original (DXL)	Feature
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_314	2.2 020102: ATO Initial activation from manual operation	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_314	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_315	2.2.1 Documentation	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_315	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_322	Der Use Case beschreibt die initiale bzw. erstmalige Aktivierung des automatisierten Fahrbetriebs, bei dem die Kontrolle der Traktions-/Bremsvorgaben (und ggf. auch Türöffnung/-schließung) vom Akteur an die ATO übergeben wird. Dies kann im Stillstand und während der Fahrt erfolgen. Abgrenzung: Dieser Use Case behandelt nur die reine Aktivierung des automatisierten Fahrbetriebs. Die anschließende automatisierte (An-) Fahrt von einem entsprechend des Journey Profils vorgegebenen Startpunkt bis zum nächste End-/Haltepunkt ist in den nachfolgenden Use Cases dargestellt.	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_322	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_316	2.2.2 Actors	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_316	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_347	Driver The driver is the person operating the train. The exact activity layout is defined by the respective railway company. He is often only responsible to a limited extent for the other technical aspects of the trainset.	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_347	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_317	2.2.3 Trigger	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_317	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_323	Der Akteur will den automatisierten Fahrbetrieb durch Starten einer automatischen Fahrt beginnen.	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_323	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_318	2.2.4 Precondition	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_318	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_327	Der ATO liegt ein Journey Profile (JP) und ein Segment Profile (SP) für den Streckenabschnitt vor. TODO: kann entfallen, siehe Precondition (11) aus übergeordnetem UseCase?!	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_327	
H.6387. FAnS.JC .ATOove rETCS_F ahrbetri eb_326	Die MA (von ETCS) ist ausreichend vorhanden, um * innerhalb des Bahnhofsbereichs: Zug kann vollständig den Bahnsteig passieren/verlassen * außerhalb des Bahnhofsbereichs: Zug-spezifische minimale Strecke durch MA freigegeben * Anfahrt auf einen Haltepunkt: eine ausreichende Bremsdistanz zum Anhalten durch die ATO während der Fahrt TODO: kann entfallen, siehe Precondition (6) aus übergeordnetem UseCase?!	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_326	
H.6387. FAnS.JC .ATOove	Fall „Zugverband im Stillstand“: Der Fahr-/Bremshebel steht in Neutralstellung.	H.6387.FAnS.JC.ATOoverET CS_Fahrbetrieb_325	Combined DBC

Abbildung 14 Beispielhafter GoA2 UseCase in DOORS

Abbildung 14 zeigt den beispielhaften GoA2 UseCase, der im Rahmen von AP3 in DOORS spezifiziert wurde.

Durch die Spezifikation des UseCase in DOORS konnte die Konsistenz der Anforderungen erhöht werden. Darüber hinaus ergab sich ein Vorteil beim Tracing der einzelnen UseCase Schritte auf übergeordnete Anforderungen. Dies ist in MagicDraw nicht ohne weiteres möglich, konnte aber durch die Auslagerung der UseCase-Spezifikation nach DOORS ermöglicht werden.

1.3.3. Evaluierung der Synchronisation von Feature-Bäumen und Feature-Konfigurationen zwischen DOORS und MagicDraw

Gemäß den Vorgaben aus AP1 wurden in MagicDraw Feature-Konfigurationen für die jeweiligen Plattformen der Fahrzeugsteuerung erzeugt. Da die Beziehungen zwischen Features und Anforderungen wichtig für den Entwicklungsprozess sind, ergab sich die Notwendigkeit die Features und Feature-Konfigurationen von MagicDraw nach DOORS zu synchronisieren. Das wurde in AP3 für ein Module Kit der Fahrzeugsteuerung umgesetzt. Durch eine neue Funktion in DOORS ist es möglich zuvor aus MagicDraw exportierte Feature-Konfigurationen zu importieren. Danach steht nicht nur die Feature-Konfiguration in DOORS zur Verfügung, es können auch die Features mit den Anforderungen verlinkt werden.

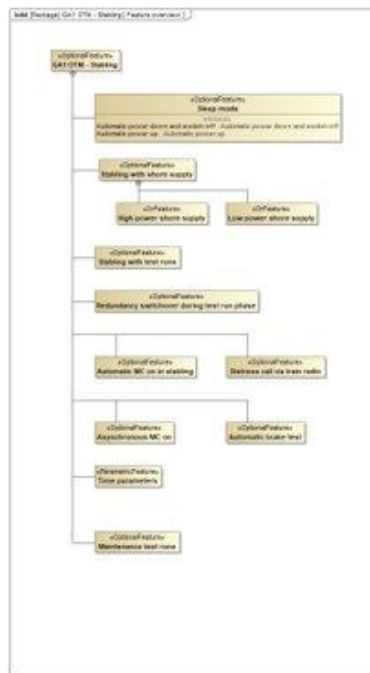


Abbildung 15 Feature-Baum in MagicDraw

Abbildung 15 zeigt einen exemplarischen Feature-Baum aus MagicDraw. Dieser wurde nach DOORS in ein dafür vorgesehenes Modul synchronisiert (siehe Abbildung 16)

ID	Feature Name
647	GA1 OTM - Stabling
648	Automatic brake test
649	Distress call via train radio
650	Maintenance test runs
651	Stabling with test runs
652	Sleep mode
653	Automatic MC on in stabling
654	Asynchronous MC on
655	Redundancy switchover during test run phase
656	Time parameters
657	Stabling with shore supply
658	Low power shore supply
659	High power shore supply
660	FF Power Storage
661	GA1 OTM - Washing
662	Washing at standstill
663	Washing mode automatic
664	Washing mode towed
665	Washing mode manual

Abbildung 16 Feature-Konfiguration und Feature-Bäume in DOORS

Aus der Synchronisation von Features und Feature-Konfigurationen resultierte direkt ein Vorteil: In DOORS konnte jede Anforderung durch eine Verlinkung dem jeweiligen Feature zugeordnet werden. Der Feature-Name musste nicht mehr von Hand an der Anforderung eingetragen

werden. Abbildung 17 zeigt eine Anforderung in DOORS und den Namen des Features, mit dem die Anforderung verknüpft ist. Der Feature-Name wird im Attribut *Feature* dargestellt. Dieser musste vorher von Hand an jeder Anforderung eingetragen werden. Durch das neue Verfahren kann der Name des verknüpften Features automatisch eingeblendet werden. Das führte zu einer höheren Konsistenz beim späteren Export der Anforderungen und Import in MagicDraw. In MagicDraw wird der Bezug zwischen Anforderungen und Features gemäß den Vorgaben aus DOORS wiederhergestellt. In der Vergangenheit gab es Probleme dabei, da der Feature-Name nicht immer richtig geschrieben wurde.

ID	Requirement	Feature
H.8820. TSAoS.J C.GA1.1 _66	<input checked="" type="checkbox"/> If stabling-related diagnostic messages occur in reposing mode, TCMS must (in addition to the distress call via RDT) send a distress call by SMS via the train radio to the operation control center.	Distress call via train radio

Abbildung 17 Bezug einer Anforderung zu einem Feature in DOORS

Schließlich ist es nun auch möglich, in DOORS die Anforderungen einzusehen, die mit einem bestimmten Feature in Bezug stehen. Hierzu muss man lediglich die Verknüpfungen an einem Feature inspizieren. Die nächste Abbildung zeigt ein Feature in DOORS. Über das gelbe Dreieck können alle eingehenden Verknüpfungen an dem Feature angesteuert werden. Dies sind alle Anforderungen mit Bezug zum Feature.

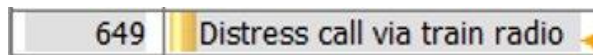


Abbildung 18 Features mit Verknüpfungen zu Anforderungen in DOORS

Durch die Verknüpfung von Anforderungen mit Features in DOORS ist es nun bereits während dem Requirements Engineering möglich, Anforderungen zu einem bestimmten Feature einzusehen und umgekehrt ein Feature über eine Anforderung ausfindig zu machen.

1.3.4. Evaluierung der Kontextdiagramme mit physikalischen Größen

Kontextdiagramme werden auf Systemebene der Fahrzeugsteuerung verwendet, um Schnittstellen zwischen den einzelnen Teilsystemen der Fahrzeugsteuerung darzustellen (Abbildung 19).

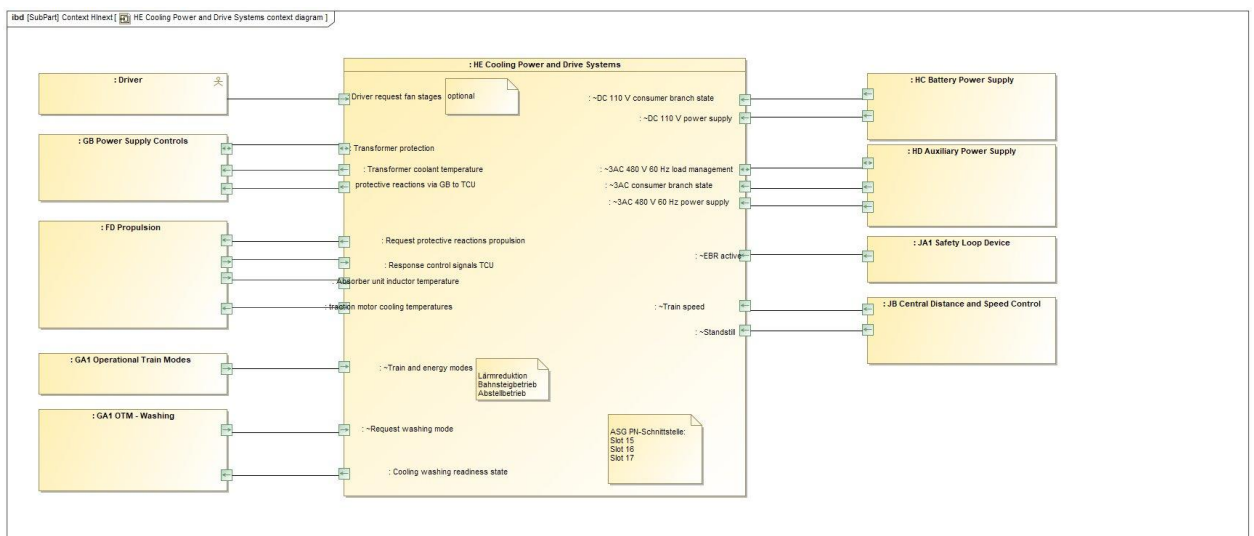


Abbildung 19 Exemplarisches Kontextdiagramm eines Teilsystems der Fahrzeugsteuerung

Da es sehr viele Schnittstellen zur Abstimmung gibt, ist es wichtig, diese zu dokumentieren. Solange Schnittstellen nicht abgestimmt und dokumentiert sind, ist das Risiko von Problemen bei der Umsetzung sehr hoch. Dabei spielt es jedoch auch eine Rolle, wie eine Schnittstelle ausgeprägt ist: Es gibt SW-Schnittstellen (Information), aber auch elektrische Schnittstellen, oder auch andere, physikalische Schnittstellen (siehe Abbildung 20).



Abbildung 20 Schnittstellentypen

Bei den Abstimmungen ist es wichtig zu wissen, was genau über die Schnittstelle übertragen wird. Da dies in Kontextdiagrammen bisher nicht ersichtlich war, wurden entsprechende Schnittstellentypen definiert und prototypisch in Kontextdiagrammen angewendet (siehe Abbildung 21). Dabei erhält jeder Typ eine unterschiedliche Farbe, um eine bessere Unterscheidung zu ermöglichen.

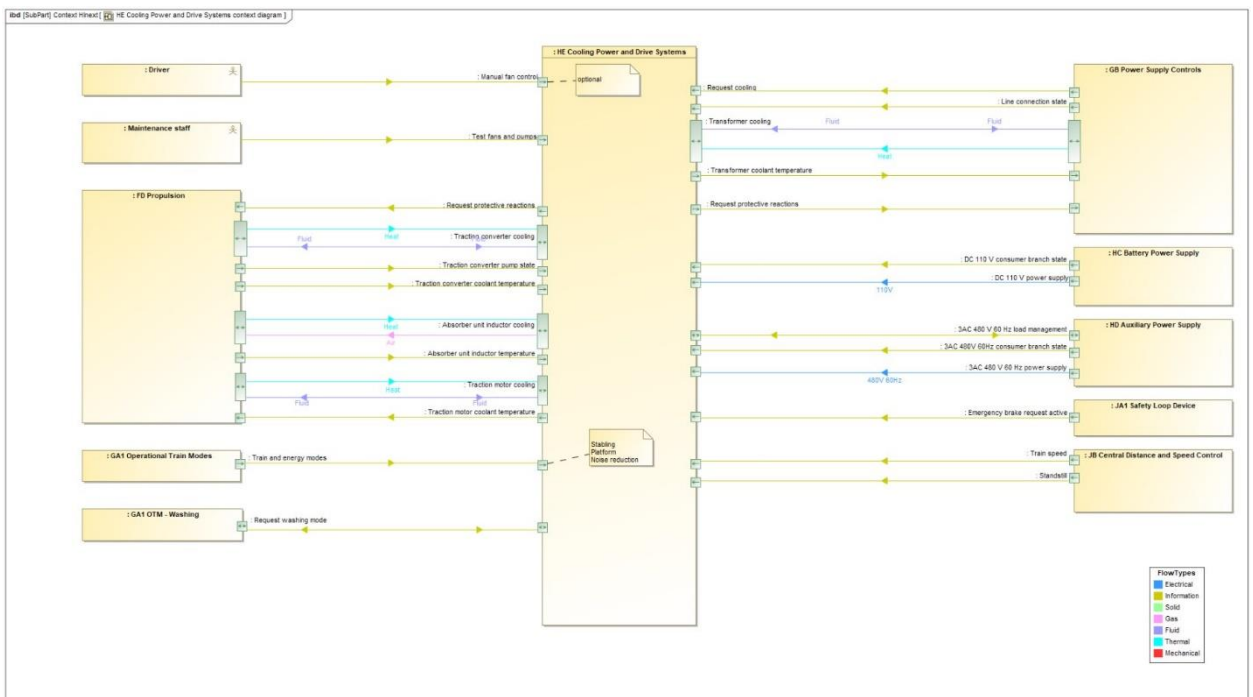


Abbildung 21 Exemplarische Hervorhebung von Schnittstellentypen im Kontextdiagramm

Durch die Spezifikation des konkreten Schnittstellentyps und die farbliche Hervorhebung in den Kontextdiagrammen waren die Schnittstellenabstimmungen fokussierter und klarer.

1.3.5. Fazit

Der PoC konnte erfolgreich abgeschlossen werden: Mit den Methoden des MBSE und PLE war es möglich, Modelle exakt zu beschreiben und dabei auch auf die Variabilität einzugehen. Es konnte AP5 und damit die exemplarische Umsetzung von GoA2 gestartet werden.

Darüber hinaus wurden auch Methoden erfolgreich evaluiert, die es ermöglichen, die Variabilität bereits bei der Anforderungserstellung zu berücksichtigen.

Schließlich wurde auch eine verbesserte Darstellung von Schnittstellen zwischen unterschiedlichen Teilsystemen einer Fahrzeugsteuerung erfolgreich evaluiert.

1.4. Entwicklung von Werkzeugen/Prozessen (AP4)

Die Entwicklung von Werkzeugen und Prozessen orientierte sich an den Vorgaben aus AP1 und AP2. Es wurde ein Modelldrift implementiert, der im Tool SonarQube die Unterschiede zwischen Modellen anzeigt. Anschließend wurde auch ein Plugin zur Anzeige von sogenannten „Variability Smells“ in einem Modell implementiert. Außerdem wurde die Generierung komplexer Softwarekomponenten auf Basis der Modellierung mit sogenannten „Inline Blocks“ umgesetzt. Hinzu kamen auch Themen, die beim Einsatz von MBSE und PLE essenziell sind, um einen guten Entwicklungsprozess möglich zu machen: Es wurde eine zyklenfreie Modellstruktur definiert und umgesetzt und der Umstieg auf eine effiziente Cloud-Infrastruktur vorbereitet.

In den folgenden Kapiteln wird jedes dieser Themen näher beleuchtet.

1.4.1. Entwicklung der Visualisierung des Modelldrift in SonarQube

Bei einer Produktlinienhierarchie aus der PLE-Methodik ist es wichtig zu wissen, wie weit unterschiedliche Produktlinien voneinander entfernt. Unterschiede zwischen einer Produktlinie und ihren Ableitungen sind besonders von Interesse. Die Unterschiede zwischen den einzelnen Produktlinien sind aber auch relevant.

Kommt neben PLE auch MBSE zum Einsatz, so ist es möglich die Unterschiede zwischen den Modellen der einzelnen Produktlinien zu visualisieren. Dies wurde in Form eines Modelldrifts ermittelt und dargestellt.

Der Modelldrift wird immer zwischen den SysML-Modellen zweier Produktlinien ermittelt. Die Ermittlung des Drifts macht nur dann Sinn, wenn die Produktlinien verwandt sind, das bedeutet, sie teilen Modellelemente mit demselben Universally Unique Identifier (UUID). In einer strategisch erzeugten Produktlinienhierarchie wird das durch den Einsatz von Branch- und Merge-Funktionen im Modellierungstool ermöglicht.

Die ausgewählten SysML Modelle werden eingelesen und vorgefiltert: Es sind nicht immer alle Modellelemente von Interesse. Je nach Stakeholder rücken andere Modellelemente bei der Ermittlung des Modelldrifts in den Vordergrund. Zum Beispiel können die Feature-Bäume unterschiedlicher Produktlinien von Interesse sein. Dadurch lassen sich Fragen nach gemeinsamen Features beantworten, aber es können auch ungewollte Unterschiede aufgedeckt werden.

Für die Visualisierung des Modelldrifts wurde SonarQube ausgewählt, da es bereits im Einsatz ist und hier einige Funktionen für den Import von Daten existierten. Der Fokus wurde auf Elemente des Feature-Baums gelegt. Abbildung 22 zeigt einen Ausschnitt des Modelldrifts. Zu erkennen ist, dass bspw. Kommentare unterschiedlich sind, aber auch die Elementnamen können abweichen, was beim selben Element nicht wünschenswert ist.

Issues 16

comment is not identical "{LANGUAGE DE} Diese optionalen Unterzustände sind notwendig, wenn das Traktionssystem von der Hauptenergieversorgung getrennt werden kann, ohne diese damit insgesamt auszuschalten; so kann z. B. die Bordnetzversorgung weiter betrieben werden. Diese Möglichkeit besteht z. B. bei Nahverkehrstrriebzügen mit Versorgung aus eine Schronschiene, bei denen Antriebs- und Hilfsbetriebeurrichter parallel mit den Stromabnehmern verbunden sind. {LANGUAGE EN} These optional sub-nodes become necessary if the traction system can be isolated separately from the main power supply without isolating the whole trainset; by this e.g. the on-board power supply can still be run. This possibility exists e.g. for regional trains with 3rd rail supply where traction and auxiliary converter are connected in parallel to the pantograph. {/LANGUAGE}" <-> "{LANGUAGE DE} Diese optionalen Unterzustände sind notwendig, wenn das Traktionssystem von der Hauptenergieversorgung getrennt werden kann, ohne diese damit insgesamt auszuschalten; so kann z. B. die Bordnetzversorgung weiter betrieben werden. Diese Möglichkeit besteht z. B. bei Nahverkehrstrriebzügen mit Versorgung aus eine Schronschiene, bei denen Antriebs- und Hilfsbetriebeurrichter parallel mit den Stromabnehmern verbunden sind. {LANGUAGE EN} These optiona... Why is this an issue? Consistency

Maintainability safety +

Open Not assigned 3min effort • 3 years ago

name is not identical "Default energy source" <-> "Energy Source" Why is this an issue? Consistency

Maintainability safety +

Open Not assigned 3min effort • 3 years ago

Abbildung 22 Darstellung des Modelldrifts in SonarQube

Obwohl die Darstellung der Unterschiede negativ eingestuft wurde, ist die Funktion als sehr sinnvoll bewertet worden. Sie befindet sich zum Zeitpunkt der Erstellung dieses Berichts in der Weiterentwicklung und soll zeitnah im Unternehmen ausgerollt werden.

1.4.2. Entwicklung eines Plugins zur Anzeige von „Variability Smells“

Eine mit dem Modelldrift vergleichbare, aber etwas andere Herangehensweise bei der Identifikation von Abweichungen zwischen Produktlinien einer Produktlinienhierarchie verfolgt die Anzeige von sogenannten „Variability Smells“. Statt explizit eine gefilterte Menge an Modellelementen miteinander zu vergleichen, wird hier der Ansatz verfolgt komplette Modelle miteinander zu vergleichen und die Unterschiede direkt im Modellierungstool MagicDraw anzuzeigen. Dadurch wird es ermöglicht bspw. überflüssige Modellelemente oder aber auch unnötige Abweichungen zwischen Produktlinien zu identifizieren, was unter dem Begriff „Variability Smells“ zusammengefasst wird.

Abbildung 23 zeigt die Funktionsweise. Dabei ist erwähnenswert, dass mehr als zwei Produktlinien miteinander verglichen werden können. Jedes Modellelement erhält eine Färbung entsprechend seiner Zugehörigkeit zu den Produktlinien. Insgesamt gibt es so viele Färbungen wie es Kombinationen der miteinander verglichenen Produktlinien gibt.

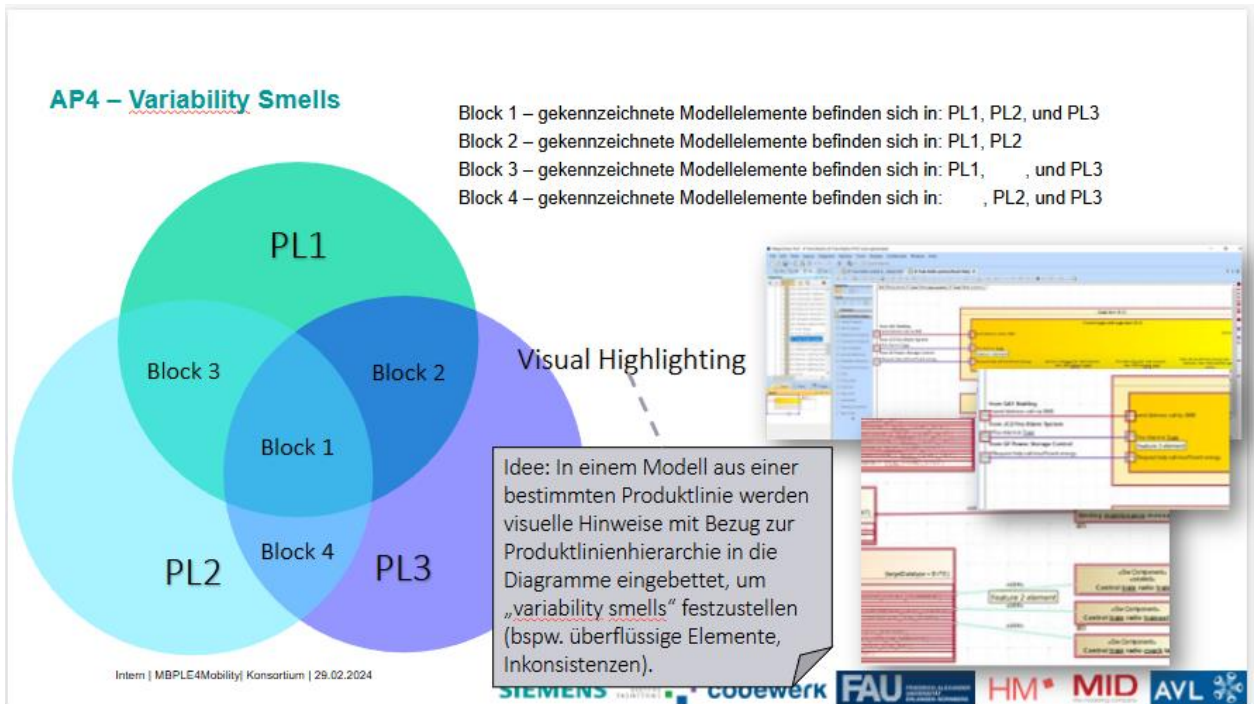


Abbildung 23 Anzeige von Variability Smells in MagicDraw

Der Vorteil dieser Herangehensweise liegt in der Darstellung: Die direkte Anzeige in MagicDraw ist besser als die Darstellung in SonarQube. Insbesondere wenn lediglich zwei Produktlinien miteinander verglichen werden. Bei mehreren Produktlinien ist die Wahl der Farben für die Kombinationen wichtig. Dabei sollte man ähnliche Farben vermeiden. Ein Nachteil liegt allerdings darin, dass komplette Modellinhalte verglichen werden, wobei je nach Stakeholder nur eine Untermenge relevant ist. Außerdem müssen bei Produktlinien, die aus mehreren Modellen bestehen, mehrere Vergleiche durchgeführt werden. Hier ist der Ansatz über den Modelldrift im Vorteil, da hier alle Modelle zweier Produktlinien innerhalb eines Vergleichs miteinander verglichen werden können.

1.4.3. Entwicklung der „Inline Blocks“ für komplexe Softwarekomponenten

Die im Rahmen des Projekts umgesetzte Funktion der „Inline Blocks“ wurde exemplarisch auf eine sehr komplexe SWC angewendet, die zum Teilsystem JD1 der Fahrzeugsteuerung gehört, welches für die Integration der Zugsicherung (bspw. ETCS) zuständig ist. Abbildung 24 zeigt die umgesetzte Zerlegung dieser SWC in mehrere Inline Blocks.

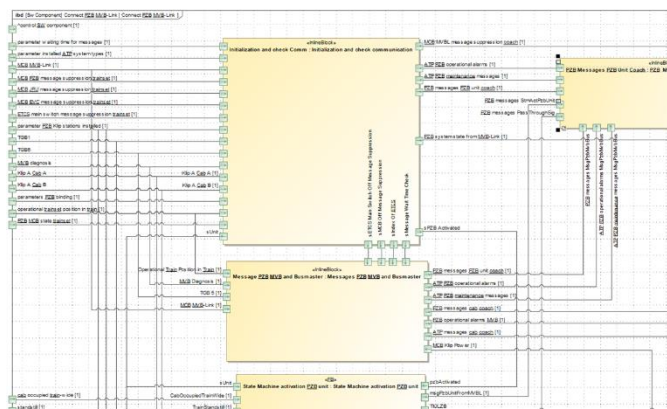


Abbildung 24 Exemplarische Zerlegung einer komplexen Softwarekomponente in Blöcke

Die Modellierung war umfangreich, da die SWC viele Informationen verarbeitet. Allerdings vereinfachte sie das Verständnis der SWC. Statt tausende Zeilen Code zu lesen und zu verstehen, existierte nun ein Modell der SWC mit ihren Funktionen. Zusätzlich ermöglichte die Modellierung die Generierung eines Code-Bereichs für jeden Inline Block innerhalb des Code-Rumpfs der SWC. Das verbesserte die Nachvollziehbarkeit des Codes. Allerdings müssen für Inline Blocks die Ein- und Ausgänge im Variablen-Bereich des SCL-Codes der SWC definiert werden, die Anzahl an generierten Codezeilen nimmt damit zu. Dieser Zusammenhang ist in Abbildung 25 dargestellt. Die für den Inline Block „State machine activation PZB unit“ zusätzlich generierten Eingänge sind gelb gekennzeichnet. Die Ausgänge sind blau gekennzeichnet.

```

sStmActPzbUnit : STRUCT // State Machine activation PZB unit
  sActivatePzbUnit : STRUCT // ActivationPzbUnit
    activateAtpSys : BOOL := FALSE; // activation command specific ATP system
  END_STRUCT;
  sOccuCabTrnWide : BOOL := FALSE; // CabOccupiedTrainWide
  trainStandstill : STRUCT // TrainStandstill
    value : BOOL := FALSE; // Value
    valid : BOOL := FALSE; // Valid
  END_STRUCT;
  sDriveDir : INT := DRVDIR_NONE; // DrivingDirection
  sUnit : STRUCT // sUnit
    opState : INT := UNIT_STATE_NOT_EXISTENT; // opState
    level : INT := UNIT_LEVEL_UNDEF; // level
    mode : INT := UNIT_MODE_UNDEF; // mode
    commState : INT := STATE_COMM_UNKNOWN; // commState
    pwrAvailable : BOOL := FALSE; // pwrAvailable
  END_STRUCT;
  sPzbAct : STRUCT // pzbActivated
    sPzbActivated : BOOL := FALSE; // PZB Activated
  END_STRUCT;
  sActivePzbUnitQ : STRUCT // ActivationPzbUnit
    activateAtpSys : INT := ACT_NOT_RDY; // activation state specific ATP system
  END_STRUCT;
  sInvlOutTulZb : STRUCT // TIULZB
    tuUicNumber : DWORD := DWORD#16#0; // TuUicNumber
    tuSlWtgNdue : BYTE := BYTE#16#0; // TuSlWtgNdue
    l2BActv3 : BYTE := BYTE#16#0; // L2BActv3
    dummy_301 : BYTE := BYTE#16#0; // Dummy_301
    dummy_302 : BYTE := BYTE#16#0; // Dummy_302
  END_STRUCT;
  sActStateUnit : STRUCT // activationStateUnit
    sActivationStateUnit : INT := PZB_ACT_NOT_RDY; // sActivationStateUnit
  END_STRUCT;
  sMsgPzbUnitFromMvbl : STRUCT // msgPzbUnitFromMvbl
    influenceFault : BOOL := FALSE; // influence disturbed
    directSwNotZero : BOOL := FALSE; // driving direction switch unoccupied cab not in 0
  END_STRUCT;

  pzbActivIncorrectly : BOOL := FALSE; // PZB not activated correctly
  cmdActivatePzb : BOOL := FALSE; // TCS activated PZB
  pzbActive : BOOL := FALSE; // PZB reports activated
  pzbCommErr : BOOL := FALSE; // PZB communication error
  valid : BOOL := FALSE; // valid
  END_STRUCT;
END_STRUCT;

```

The diagram shows a block titled "State Machine activation PZB unit". It has several input ports on the left: sUnit, CabOccupiedTrainWide [1], TrainStandstill [1], DrivingDirection [1], and ActivationPzbUnit [1]. It has output ports on the right: sPzbActivated, msgPzbUnitFromMvbl [1], and TIULZB : TU_LZB. Below the diagram, a legend indicates that yellow boxes represent the Simatic name of the IN Ports and blue boxes represent the Simatic name of the OUT Ports.

Abbildung 25 Generierte Ein- und Ausgänge für Inline Block

In Abbildung 26 ist der Aufruf des Inline Blocks „State machine activation PZB unit“ dargestellt. Dabei handelt es sich *nicht* um einen üblichen Funktionsaufruf, da der Block direkt eingebettet ist (inline). Der/die Entwickler/in schreibt den Funktionsanteil des Blocks an die Stelle zwischen dem Einlesen der Eingänge und dem Schreiben der Ausgänge; in Abbildung 26 ist dies durch die Zeile mit dem Strichpunkt (;) dargestellt.

```

(* INLINE BLOCK
INSTANCE: State Machine activation PZB unit
TYPE: State Machine activation PZB unit

DESCRIPTION:
This is the rough documentation of State Machine activation PZB unit

INPUT:
ActivationPzbUnit: Activation ATP system (generic) (in)
CabOccupiedTrainWide: Cab train-wide occupied (out)
TrainStandstill: Train standstill (out)
DrivingDirection: Target driving direction (out)
sUnit: sUnit (out)
sPzbActivated: PZB Activated (out)

OUTPUT:
ActivationPzbUnit: Activation ATP system (generic) (out)
TIULZB: TIU_LZB (out)
sActivationStateUnit: sActivationStateUnit (out)
sMsgPzbUnitFromMvbl: PZB messages PZB unit (out *)
sStmActPzbUnit.sActivatePzbUnit := cmdActivatePzbUnitMvbl;
sStmActPzbUnit.sOccuCabTrnWide := cabOccuTrn;
sStmActPzbUnit.sTrainStandstill := standstill;
sStmActPzbUnit.sDriveDir := driveDir;
sStmActPzbUnit.sUnit := sIniCheckComm.sUnit;
sStmActPzbUnit.sPzbAct := sIniCheckComm.sPzbActivated;
(*PROTECTED REGION ID Id_19_0_1_afd029a_1701748081039_403152_299561-InlineInstanceImplementation) ENABLED START*)
;
(*PROTECTED REGION END*)
cmdActivatePzbUnitMvblQ := sStmActPzbUnit.sActivePzbUnitQ;
sInvlOutTulZb := sStmActPzbUnit.sInvlOutTulZb;

```

The diagram shows the inline block call within a SW component. The component has input ports: cab occupied train-wide [1], standstill [1], driving direction [1], and activation PZB unit [1]. The inline block call is: sUnit : State Machine activation PZB unit;. The component has output ports: sPzbActivated, msgPzbUnitFromMvbl [1], and TIULZB : TU_LZB. The output ports of the inline block are connected to the output ports of the SW component.

Abbildung 26 Einbettung des Inline Blocks in den Code-Rumpf der SWC

Der generierte Bereich enthält darüber hinaus noch ausführliche Kommentare, um ihn besser verstehen zu können. Dies fördert auch die Nachverfolgbarkeit zwischen Modell und Code.

Ein weiterer Vorteil besteht in der Generierung der Dokumentation: Anstelle eines mehrere Seiten langen Textes kann nun eine strukturierte Sicht auf die komplexe SWC ausgeleitet werden. Das Vorgehen ist dabei identisch zu übergeordneten Ebenen: Es gibt für die komplexe SWC eine strukturelle Sicht (ibd) mit einer Beschreibung zu allen Inline Blocks und eine Verhaltenssicht mit einer Beschreibung zum Verhalten. Abbildung 27 zeigt den Anfang der exemplarischen Ausleitung der komplexen SWC „Control operating element cleaning basic head“. Es folgt zuerst das ibd, also die strukturelle Sicht. In weiteren Kapiteln werden die einzelnen Beschreibungen der Inline Blocks ausgegeben. Abschließend folgt die Dokumentation des Verhaltens zur SWC.

4.1.1.7 Control operating element cleaning basic head «SwComponent»

Basic component information:

Simatic name: KC_CtrlIntCleanOpCab_SC

Component ID: 59775

Rough documentation:

Die Softwarekomponente "Steuerung Bedienelement Innenreinigung" empfängt die Bedienhandlungen des Leuchtdrehtasters Innenreinigung, wertet diese aus und plausibilisiert sie. Anschließend werden die interpretierten Bedienhandlungen des Bedienelements als Steuerbefehle an die übergeordneten Softwarekomponenten der Innenreinigungsteuerung übermittelt.

Die Softwarekomponente ist auch für die Integration und Ansteuerung der Statusmeldeleuchte des Bedienelements verantwortlich.

4.1.1.7.1 Detailed Design

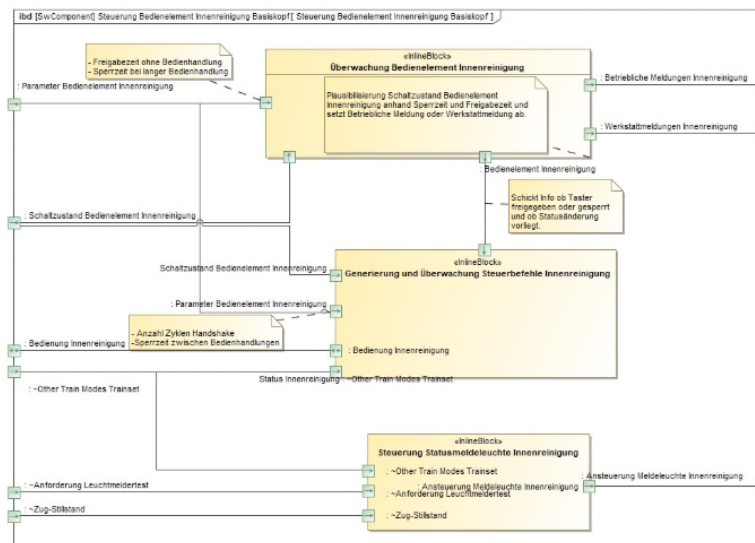


Figure 1 SW Component Bedienelement Innenreinigung Basiskopf detailed design

Here Text if the documentation field of the ibd is filled with additional description.

--- DCC: ---	--- Version: --- Freigabedatum: ---
--- © Siemens Mobility GmbH 2023	Seite 69 von 119

Abbildung 27 Beispielhafte Ausleitung der Dokumentation einer komplexen SWC mit Inline Blocks

Bei der Anwendung von Inline Blocks ist es wichtig festzulegen, ab wann eine SWC als komplex angesehen wird. Diese Festlegung wurde im Rahmen des Projekts erarbeitet und festgehalten. Auch wenn Inline Blocks die Verständlichkeit und Nachvollziehbarkeit erhöhen, macht ihre Anwendung bei einfachen SWCs keinen Sinn, da der zusätzliche

Modellierungsaufwand und die zusätzlich generierten Codezeilen für einfache Komponenten keinen Mehrwert erzeugen.

1.4.4. Entwicklung einer zyklenfreien Modellstruktur

Ein wichtiger Aspekt beim Einsatz von MBSE ist die eingesetzte Modellstruktur. Kleine Projekte können in der Regel in einem einzigen Modell umgesetzt werden. Bei größeren Projekten wie der Umsetzung der GoA2 Funktion in der Fahrzeugsteuerung eines Zuges sind allerdings viel mehr Modelle notwendig. Eine Produktlinie kann über 100 Modelle besitzen, die miteinander in Verbindung stehen. Daher ist eine definierte Modellstruktur sehr wichtig, um die Komplexität zu beherrschen, die sich aus den Modellabhängigkeiten ergibt.

Im Rahmen des MBPLE Projekts wurde die zugrundeliegende Modellstruktur analysiert und auf Schwachstellen analysiert. Abbildung 28 stellt einen Ausschnitt aus der erfassten Modellstruktur dar. Durch die Spezifikation war es möglich, die Modelle zu kategorisieren und eine allgemeine Sicht auf die Abhängigkeiten zu erstellen.



Abbildung 28 Ausschnitt der analysierten Modellstruktur

Danach wurde die Modellstruktur analysiert und Modellstrukturrichtlinien festgelegt:

- es wurden Modelltypen und deren Inhalt definiert
- es wurden die möglichen Abhängigkeiten zwischen den unterschiedlichen Modelltypen definiert

Das Ergebnis ist in Abbildung 29 dargestellt. Es wurden als Modelltypen Profile, Libraries, Collections und Working models festgelegt. Jeder Modelltyp erfüllt einen bestimmten Zweck und beinhaltet definierte Elemente. Darüber hinaus ist klar definiert, welche Abhängigkeiten erlaubt sind. Dabei wurde insbesondere darauf geachtet, dass keine Modellzyklen entstehen können.

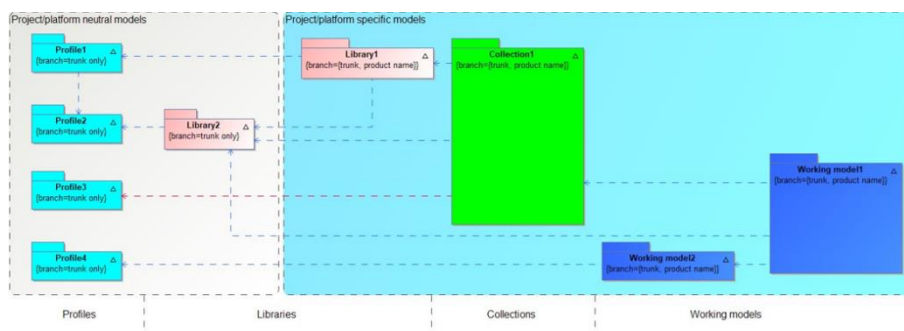


Abbildung 29 Ableitung von Modellstrukturrichtlinien

Die Analyse offenbarte eine Schwachstelle in der Modellstruktur: Zyklen in den Modellabhängigkeiten.

Ein Zyklus ergibt sich durch eine gegenseitige Abhängigkeit von Modellen: Ein Modell nutzt die Inhalte eines anderen Modells, während das andere Modell ebenfalls auf die Inhalte des ursprünglichen Modells zugreift. Ein solcher Zyklus kann sich auch über mehrere Modellabhängigkeiten ergeben und fällt daher nicht immer sofort auf. In einer Modellstruktur müssen Zyklen vermieden werden.

Der Großteil der identifizierten Zyklen konnte durch einen einfachen Umbau der Abhängigkeiten nach den Modellstrukturrichtlinien behoben werden. Es existierte jedoch eine zyklische Abhängigkeit, die einen großen Modellumbau zur Folge hatte. Dafür mussten die betroffenen Modellanteile aus ihren ursprünglichen Modellen in ein jeweils neu erstelltes Modell umgezogen werden. Es wurden pro Produktlinie bis zu 40 neue Modelle hierzu eingeführt. Der Modellumbau einer Produktlinie beinhaltete folgende Arbeiten:

1. Erstellen der neuen Modelle
2. Umzug der betroffenen Modellanteile in die neuen Modelle
3. Aktualisierung aller Modellabhängigkeiten

Da solch ein manueller Umbau viel Zeit benötigt hätte, wurde der Prozess so aufgesetzt, dass er fast ausschließlich automatisch durchgeführt werden konnte. Es wurden entsprechende Funktionen in MagicDraw implementiert, die mittels einer vorgegebenen Konfiguration die Modellstruktur anpassen. Das Vorgehen wurde erfolgreich auf die gesamte Produktlinienhierarchie angewendet.

Der Modellumbau führte nicht nur zu einer klaren und fehlerfreien Modellstruktur. Er verbesserte auch die Performance beim Öffnen und Bearbeiten von Modellen in MagicDraw. Damit ist über die gesamte Produktlinienhierarchie eine spürbare Verbesserung eingetreten. Darüber hinaus ermöglichte der Umbau auch eine fehlerfreie Migration von einer klassischen Server-Infrastruktur in eine Cloud-Infrastruktur.

1.4.5. Vorbereitung Cloud-Infrastruktur für das Modellierungswerkzeug MagicDraw

Die Vorteile von MBSE- und PLE-Methoden bei der Entwicklung von komplexen Systemen, wie bspw. der Fahrzeugsteuerung eines Triebzuges, zeigten sich durchweg im Rahmen des MBPLE Projekts. Ein Aspekt, der dabei aber bisher nur wenig in Betracht gezogen wurde, ist die Bereitstellung des Modellierungstools und dessen Infrastruktur über einen längeren Zeitraum hinweg. Zum Beispiel kann ein Triebzug durchaus eine vorgegebene Lebenszeit von 30 Jahren haben. Die zum Einsatz kommenden SW-Entwicklungssysteme müssen in dieser Zeit verfügbar sein, um bspw. wichtige SW-Patches durchführen zu können. Hierzu zählen auch Modelle, aus denen Code generiert wird. Daher ist es notwendig, das verwendete Modellierungswerkzeug inkl. seiner Datenbank für den langfristigen Einsatz vorzubereiten. In diesem Kontext ist es wichtig, das Modellierungstool inkl. seiner Infrastruktur auf dem neuesten Stand zu halten.

Eine der wichtigsten Neuerungen der letzten Jahre ist der Umstieg von einer klassischen Server-Infrastruktur hin zu einer Cloud-Infrastruktur. MagicDraw unterstützt ab der Version 2021 nur eine Cloud-Lösung (Teamwork Cloud) als Infrastruktur. Die klassische Serverlösung (Teamwork Server) ist mit der Version MagicDraw 19 ausgelaufen. Eine Änderung dieser Art stellt Unternehmen vor größere Probleme, insbesondere wenn MBSE intensiv im Einsatz ist: Eine Aktualisierung ist notwendig, um weiterhin Support zu bekommen, allerdings ist dafür ein Umzug der Daten von einem klassischen Server hin zur Cloud notwendig.

Um auf solche tiefgreifenden Änderungen vorbereitet zu sein, bietet es sich an, ein Testsystem (Staging) aufzusetzen. Im Staging können vorab alle Änderungen umgesetzt und getestet werden. Dadurch können frühzeitig Probleme identifiziert und angesprochen werden.

Im Rahmen des MBPLE Projekts wurde ein Staging für MagicDraw Teamwork Cloud aufgesetzt, um eine Migration von der klassischen Teamwork Server Infrastruktur hin zur Cloud Infrastruktur zu testen. Der Fokus lag dabei auf zwei wesentlichen Punkten:

1. Die Performance der zukünftigen Infrastruktur
2. Eine konsistente und fehlerfreie Migration der Daten von der alten in die neue Infrastruktur

Um die zukünftige Infrastruktur richtig auszulegen, wurde initial die Performance des Staging beobachtet, während einzelne Produktlinien aus der alten Serverinfrastruktur in die neue Cloud-Infrastruktur migriert wurden. Es wurden erfolgreich mehrere Testläufe durchgeführt und dabei unterschiedliche Stellschrauben in der Cloud evaluiert. Schließlich konnte eine geeignete Konfiguration gefunden werden, welche die Performance bei der Migration maximiert.

Es zeichnete sich allerdings bereits ab, dass sich die Migration über einen längeren Zeitraum ziehen wird: Durch den PLE-Ansatz existieren mehrere Produktlinien, die migriert werden müssen. Daher ist eine Nutzung der produktiven Cloud-Infrastruktur durch Entwickler während einer Migration gegeben. Aufgrund von Projektterminen ist es nämlich nicht möglich, den Zugang zur produktiven Cloud-Infrastruktur während einer Migration vollständig zu sperren. Darüber hinaus lässt sich die Migration einer Produktlinie nicht pausieren. Um dieses Szenario zu beleuchten, wurden Lasttests im Staging durchgeführt: Während einer Migration einer Produktlinie sollten mehrere Nutzer gleichzeitig auf die Cloud-Infrastruktur zugreifen. Die Ladezeiten der Modelle wurden protokolliert. Erste Lasttests offenbarten Verbesserungspotential. Es wurden zusätzliche Einstellungen identifiziert, die einen positiven Effekt auf die Performance der Cloud-Infrastruktur während einer Migration haben. Ein abschließender Lasttests mit einer repräsentativen Anzahl an Personen, die während einer Migration gleichzeitig unterschiedliche Modelle bearbeiteten, lieferte positive Ergebnisse.

Neben der Performance spielt auch die Konsistenz der migrierten Daten eine wichtige Rolle. Da MBSE in vollem Umfang eingesetzt wird (Codegenerierung und Dokumentengenerierung), ist eine Prüfung der migrierten Daten möglich. Dadurch wurde ein Problem bei der Migration der Daten offenbart: Es gab eine unbeabsichtigte Änderung der UUID von definierten Modellelementen, siehe Tabelle 2.

Branch	TWS Element ID	TWC Element ID
trunk	_19_0_2_a560294_1587745234280_173521_275769	_19_0_2_a560294_1587745234280_173521_275769
Branch1	_19_0_2_a560294_1587745234280_173521_275769	_19_0_4_b0e027f_1743579453953_804435_3268
Branch2	_19_0_2_a560294_1587745234280_173521_275769	_19_0_4_b0e027f_1743579754588_400400_2993

Tabelle 2: Unbeabsichtigte UUID-Änderung bei Migration in Cloud-Infrastruktur

Es konnte daraufhin bewertet werden, welche Auswirkungen diese Änderungen haben und welche Maßnahmen existieren, um sie zu verhindern. Da die Auswirkungen sehr gering sind, wurde gemeinsam mit allen Stakeholdern beschlossen keine Maßnahmen anzuwenden und lediglich die Entwickler/-innen zu informieren.

Abschließend konnte über das Staging der komplette Migrationsprozess evaluiert und alle Probleme gelöst werden. Die Migration der Produktlinienhierarchie von einer klassischen Serverinfrastruktur hin zu einer neuen Cloud-Infrastruktur konnte damit beginnen.

1.5. Exemplarischer Nachweis der Anwendbarkeit der MBPLE-Methoden am Beispiel der GoA2 Funktionsmodule der Fahrzeugsteuerung (AP5)

Das Ziel von AP5 war es, die Methoden des MBPLE Projekts in der prototypischen GoA2-Entwicklung im Fahrzeugsteuerungsanteil eines Triebzuges zu erproben. Dafür sollte eine funktionsfähige Produktlinie der Fahrzeugsteuerung eines Triebzuges als Ausgangsbasis ausgewählt werden. Mit Methoden des MBSE und PLE sollte eine Ableitung der Produktlinie erzeugt und um eine prototypische GoA2-Funktion erweitert werden. Die Umsetzungen sollten so vollzogen werden, dass sie mit den Methoden des MBPLE in andere Produktlinien überführt werden kann.

In den folgenden Kapiteln wird zuerst die GoA2-Funktion und danach die Aufgaben der Fahrzeugsteuerung eines Triebzuges kurz beschrieben, damit der Kontext verständlich ist. Danach wird auf die Fahrzeuganforderungs- und Fahrzeugarchitekturspezifikation für GoA2 eingegangen, die mit MBPLE-Methoden erstellt wurde. Anschließend erfolgt eine kurze Einführung in die Produktlinienhierarchie der Fahrzeugsteuerung und die Bestimmung einer geeigneten Produktlinie als Ausgangsbasis. Von besonderem Interesse ist die Erstellung einer Ableitung dieser Produktlinie. Es werden dann die Änderungen an der Systemarchitektur der Ableitung erläutert, die für die GoA2-Funktion notwendig sind. Danach wird auf die Änderungen der betroffenen Teilsysteme eingegangen und auf die Erstellung einer ersten Version der Fahrzeugsteuerungsapplikation inkl. erster Tests an einem Hardware-in-the-Loop-Teststand (HIL). Abschließend wird der Teststandsaufbau erläutert, inkl. der Inbetriebnahme der Geräte und den damit einhergehenden Herausforderungen.

1.5.1. Grades of Automation (GoA)

Es existieren verschiedene Stufen des automatischen Betriebs bei Triebzügen (Grades of Automation – GoA). Abbildung 30 stellt die Stufen dar.

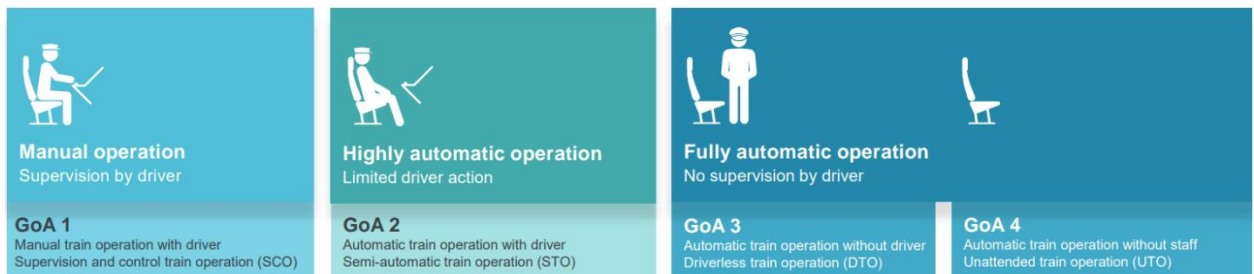


Abbildung 30 Grades of Automation

GoA1 entspricht dem manuellen Betrieb des Triebzuges durch den Triebfahrzeugführer (Tf). Dabei wird der Triebzug vom System überwacht, welches dem Tf Empfehlungen geben kann.

GoA2 definiert bereits einen stark automatisierten Betrieb: Der Tf initiiert die automatische Fahrt, woraufhin das System den Fahrbetrieb aufnimmt. Der Tf überwacht den Betrieb und kann jederzeit eingreifen. Der Tf ist auch für die Erkennung von Hindernissen auf der Strecke verantwortlich. Ebenso ist der Tf für die Abfertigung am Bahnsteig zuständig. Das System übernimmt also lediglich die Fahrt von einer Haltestelle zur nächsten.

GoA3 und GoA4 definieren einen automatischen Betrieb eines Triebzuges ohne Tf. Beim GoA3-Betrieb ist noch Zugpersonal an Bord, wohingegen für GoA4 auch kein Zugpersonal mehr an Bord ist.

Im MBPLE-Projekt lag der Fokus auf einer prototypischen Umsetzung von GoA2 im Fahrzeugsteuerungsanteil eines Triebzuges mit den erarbeiteten MBSE- und PLE-Methoden. In

der Architektur der Funktion gibt es zwei wesentliche Bereiche: Trackside ATO (ATO-TS) und On-board ATO (ATO-OB). Abbildung 31 zeigt beide Teile.

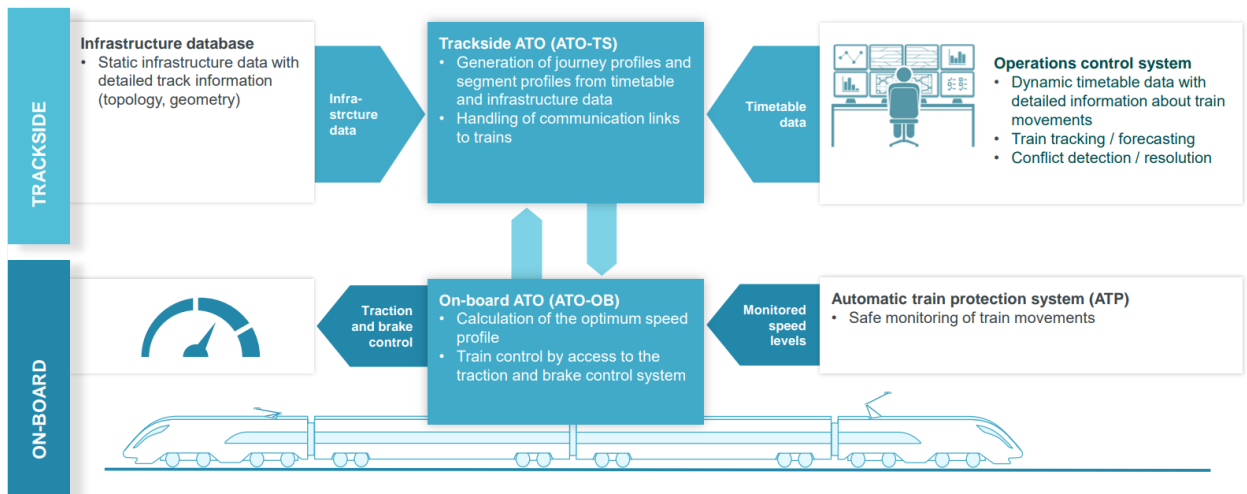


Abbildung 31 Trackside ATO (ATO-TS) und On-board ATO (ATO-OB)

Die ATO-TS stellt der ATO-OB statische Informationen zur Strecke zur Verfügung, auf der sich der Triebzug befindet. Dabei handelt es sich bspw. um Informationen zu Haltestellen oder Höchstgeschwindigkeiten. Darüber hinaus werden auch dynamische Informationen, wie Fahrplandaten, der ATO-OB zur Verfügung gestellt. Die ATO-OB verarbeitet diese Daten und wird vom Tf aktiviert, um automatisch auf der Strecke zu fahren. Das Gesamtsystem wird dabei von der Zugsicherung (ATP) überwacht. Das Betriebsszenario für eine automatisiert Fahrt ist im Detail in Abbildung 32 dargestellt.

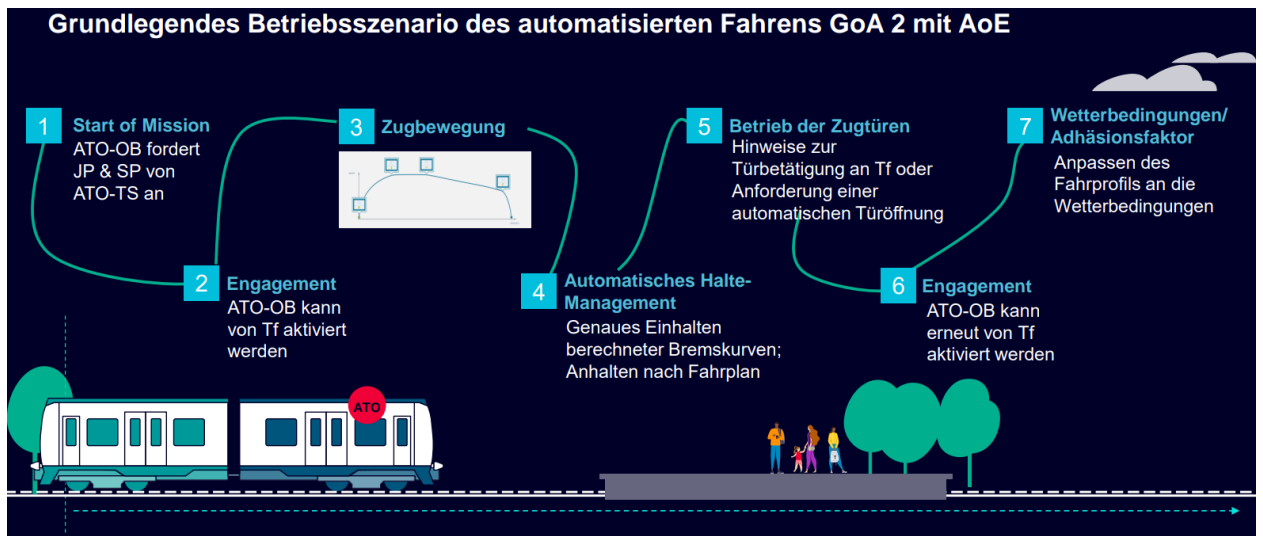


Abbildung 32 Betriebsszenario GoA2

Den Start des Betriebsszenario (1) markiert die Übermittlung der streckenrelevanten Daten (statisch und dynamisch). Danach kann der Tf das ATO-OB aktivieren (2), woraufhin das System den Zug in Bewegung versetzt (3). Gemäß den Fahrplan- und Streckendaten ermittelt das ATO-OB Sollwerte, um an der nächsten Haltestelle so genau wie möglich anzuhalten (4). Optional kann das ATO-OB zusätzlich eine automatische Öffnung der Zugtüren am Bahnsteig anfordern (5). Nach der Abfertigung kann der Tf das System erneut aktivieren (6), sodass der nächste Streckenabschnitt automatisiert befahren wird. Während des Betriebs können weitere

dynamische Informationen vom ATO-TS an das ATO-OB übermittelt werden, die einen Einfluss auf das Fahrprofil haben (bspw. Wetterdaten).

1.5.2. Aufgaben der Fahrzeugsteuerung eines Triebzuges

Ein Triebzug besteht aus mehreren Systemen. Eines davon ist die Fahrzeugsteuerung. Sie integriert alle anderen Systeme, wie bspw. die Klimasteuerung, den Antrieb, die Bremse, aber auch die Bedienelemente. Sie vernetzt die Systeme des Triebzugs miteinander, ermöglicht den Datenaustausch oder steuert sie.

Die Fahrzeugsteuerung teilt sich in mehrere Teilsysteme auf. Jedes Teilsystem ist damit betraut ein anderes System bzw. eine andere Funktion in den Triebzug zu integrieren. Im MBPLE Projekt sind die folgenden Teilsysteme relevant:

- GC1 Drive Brake Control
- GC3 Automatic Train Operation
- JD1 Automatic Train Protection

Das Teilsystem GC1 Drive Brake Control ist für das Einlesen der Sollwerte für das Fahren und Bremsen vom Tf zuständig. Es integriert dafür das entsprechende Bedienelement. GC1 leitet die Sollwerte auch an andere Teilsysteme weiter, die dann die Vorgaben umsetzen.

Das Teilsystem GC3 Automatic Train Operation ist für die Integration des ATO-OB im Triebzug zuständig. Es übermittelt den Triebzugzustand an das ATO-OB und empfängt dessen Zustand sowie dessen Sollwertvorgaben, sofern ein automatischer Fahrbetrieb vom Tf aktiviert wurde.

Das Teilsystem JD1 Automatic Train Protection integriert das Zugsicherungssystem, welches die Zugbewegungen durch das ATO-OB überwacht.

1.5.3. Fahrzeuganforderungsspezifikation (FAnS)

Die Fahrzeuganforderungsspezifikation (FAnS) beschreibt alle Anforderungen zum Fahrzeug. Sie beinhaltet textuelle Anforderungen, aber auch UseCases. Da das GoA2 Betriebsszenario einen komplexeren Ablauf darstellt, wurden hauptsächlich UseCases für die Spezifikation der Anforderungen verwendet. Als Ausgangslage dient die zugrundeliegende Norm ERTMS/ATO. Die UseCases wurden zentral spezifiziert und mit allen relevanten Stakeholdern durchgesprochen. Abbildung 33 stellt die spezifizierten UseCases dar.



Abbildung 33 UseCase Diagram FAnS ATO GoA2

Ein weiteres Arbeitspaket, welches seinen Weg in die FAnS fand, ist der Feature-Baum zur GoA2 Funktion. Durch die Spezifikation der Features war es möglich alle Optionen und Möglichkeiten früh im Entwicklungsprozess darzustellen. Die teils geforderte Variabilität hat einen großen Einfluss auf die nachfolgende Umsetzung der Architektur. Abbildung 34 stellt den Feature-Baum dar.

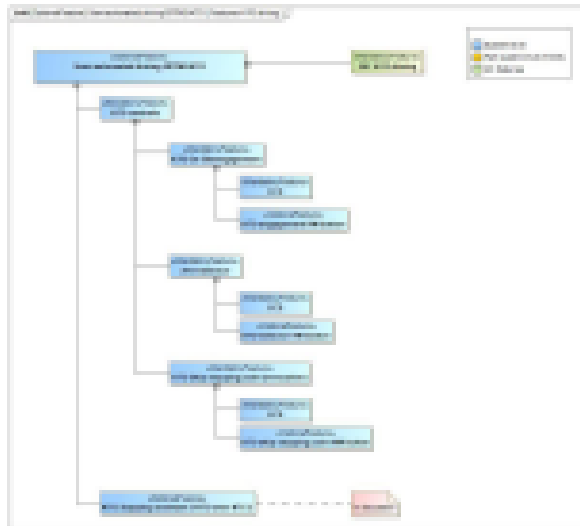


Abbildung 34 Feature-Baum zu GoA2

Abschließend wurde mit den Methoden des MBSE eine FAnS aus dem entsprechenden Modell ausgeleitet, um die UseCases und den Feature-Baum übersichtlich in Dokumentenform darzustellen. Das Dokument konnte sehr gut zum Review verteilt werden. Die zentrale Pflege der Anteile im Modell verhinderte Inkonsistenzen bei der Durchsprache.

1.5.4. Fahrzeugarchitekturspezifikation (FArS)

An die FAnS anschließend wurde die Fahrzeugarchitekturspezifikation (FArS) spezifiziert. Dafür wurde das System in Subsysteme aufgeteilt und detaillierte Abläufe für die UseCases in Form von Verhaltensdiagrammen spezifiziert. Jedem Subsystem wurden Aktivitäten aus den UseCases zugeordnet, sodass sich eine klare Aufteilung der Verantwortlichkeiten ergibt. Die Spezifikation der Diagramme mit den Methoden des MBSE offenbarte erneut viele Vorteile: Eine zentrale Datenbasis als Grundlage verhinderte Inkonsistenzen: Alle Stakeholder waren auf demselben Informationsstand und Änderungswünsche wurden an einer Stelle eingebracht. Darüber hinaus konnte erneut ein Dokument aus dem Modell ausgeleitet werden, das die relevanten Inhalte zusammenfasste. Durch das Dokument war es möglich ein bereits Review der Inhalte durchzuführen, um sie endgültig abzustimmen. Die FArS ist ein wichtiges Dokument für die nachfolgenden Entwicklungsprozesse.

1.5.5. Ableitung einer MBPLE-Produktlinie

Für die Anwendung der MBPLE-Methoden im Kontext der Fahrzeugsteuerung, sollte in einem ersten Schritt eine eigene, vollständige Produktlinie innerhalb der Produktlinienhierarchie erstellt werden.

Die Identifikation einer geeigneten Produktlinie richtet sich vorrangig nach ökonomischen Gesichtspunkten. Es wurde darauf geachtet, dass die Absprungbasis möglichst viele der benötigten Funktionen zur Verfügung stellt. Bspw. ist es für GoA2 von Vorteil, wenn die Absprungbasis bereits das Zugsicherungssystem European Train Control System (ETCS) unterstützt. Besonders im Fokus waren konfigurierbare Triebzugplattformen, da sie sehr variabel sind und viele Funktionen bieten. Die Variabilität konnte sehr gut über die Feature

Trees eingesehen und bewertet werden. Durch sie konnte schnell eine geeignete Absprungbasis gewählt werden.

Anschließend wurden die MBPLE-Produktlinie als neuer Zweig (Branch) der Absprungbasis aufgesetzt. Im Modellierungswerkzeug MagicDraw wurde dazu für alle relevanten Modelle ein entsprechender Branch eingerichtet.

Da sich Entwicklungsartefakte einer Produktlinie in der Regel in unterschiedlichen Tools befinden, ist darauf zu achten, dass in allen Tools ein entsprechender Branch realisiert wird. Bei gängigen Konfigurationsmanagementsystemen ist das gegeben, ebenso bei Modellierungswerkzeugen wie MagicDraw. Bei Anforderungs- oder Testmanagementsystemen kann es notwendig sein, Anpassungen vorzunehmen.

Um gezielt eine Ableitung zu erzeugen ist es außerdem notwendig, eine Konfigurationsbeschreibung über alle Entwicklungsartefakte zu haben. Dadurch ist man in der Lage, einen Branch zu einem spezifischen Versionsstand einzurichten.

Für die Arbeiten im MBPLE-Projekt existierte eine solche Konfiguration in der ausgewählten Absprungbasis. Die Erstellung der neuen Produktlinie war damit gezielt und auf einfache Art und Weise möglich.

Um die Richtigkeit der neuen Produktstruktur zu verifizieren, wurde anschließend ein Akzeptanztest durchgeführt, bei dem die wesentlichen Funktionen der Fahrzeugsteuerung geprüft werden. Da zu diesem Zeitpunkt noch kein geeigneter Teststand zur Verfügung stand, wurde ein Teststand der Absprungbasis verwendet und mit der ersten Version der Fahrzeugsteuerungssoftware aus der MBPLE-Produktlinie geladen. Die Akzeptanztests wurden erfolgreich bestanden. Abbildung 35 zeigt die positiven Testergebnisse.

5.1 Übersicht

5.2 Ergebnisse nach Anforderungen

Anforderung ID	Anforderung Name	Testfall ID	Testfall Name	Testfall Ergebnis	Anforderung Ergebnis
H.6258.TS AnS.JC.G A3_10	H.6258.TSAnS.JC.GA3_10	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A3_56	H.6258.TSAnS.JC.GA3_56	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A3_57	H.6258.TSAnS.JC.GA3_57	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A3_6	H.6258.TSAnS.JC.GA3_6	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A3_7	H.6258.TSAnS.JC.GA3_7	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A3_9	H.6258.TSAnS.JC.GA3_9	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A4_10	H.6258.TSAnS.JC.GA4_10	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A4_20	H.6258.TSAnS.JC.GA4_20	RTL2_TC-1	Akzeptanztest	passed	passed
H.6258.TS AnS.JC.G A4_3	H.6258.TSAnS.JC.GA4_3	RTL2_TC-1	Akzeptanztest	passed	passed

Abbildung 35 Testergebnisse aus dem Akzeptanztest der neuen Produktlinie

Zu den getesteten Funktionen gehören zum Beispiel:

- Betriebszustände
- Führerraum besetzen
- Zugkonfiguration bestätigen
- Stromabnehmer anlegen
- Hauptschalter betätigen
- Führerraumwechsel
- Fahren
- Bremsen
- Bedienung der Türen

Die positiven Ergebnisse aus dem Akzeptanztest belegen bereits initial, dass sich MBSE- und PLE-Methoden bewähren. Eine konsequente und formale Spezifikation mittels MBSE und die Spezifikation der Variabilität mittels PLE ermöglichen das gezielte Erzeugen von Produktlinien, die bereits zu ihrem Lebensbeginn funktionsfähig sind und erweitert werden können.

1.5.6. Anpassungen der Systemarchitektur

Nachdem erfolgreich die MBPLE-Produktlinie aufgesetzt wurde, war das nächste Ziel die prototypische Umsetzung der GoA2-Funktion. Hierzu wurden zuerst notwendige Anpassungen an der Systemarchitektur erfasst und umgesetzt.

Ein wichtiger Aspekt dabei ist die Schnittstellenspezifikation des Zugsicherungssystems European Train Control System (ETCS). Zum Zeitpunkt des MBPLE-Projekts gab es noch kein ETCS, das den Standard ESBO 10 vollumfänglich implementierte. Daher wurde entschieden, ein ETCS mit der Schnittstellenspezifikation nach ESBO 9.2 zu verwenden und zu integrieren. Damit einher gingen diverse Änderungen an der SW-Schnittstelle zwischen dem ETCS und der Fahrzeugsteuerung. Aber auch Änderungen an den Schaltungsdiagrammen.

Ein weiterer wichtiger Punkt ist die sogenannte ETCS-Projektierung. Für ein ETCS-System müssen viele Fahrzeugparameter festgelegt werden. Dazu zählen bspw. Werte wie die maximale Höchstgeschwindigkeit, aber auch komplexere Parameter wie Bremskurven. Diese wurden zusammen mit Stakeholdern definiert und umgesetzt.

Anschließend wurde festgelegt, wie genau die Schnittstelle zwischen der ATO-OB und der Fahrzeugsteuerung umgesetzt werden soll. Hierfür entstand eine erste Version einer Schnittstellenspezifikation. Grundlage hierfür waren sowohl die FARs, als auch die ATO-OB Schnittstellenspezifikation des Standards ERTMS/ATO. Darauf aufbauend wurde die ProfiNet (PN) Anbindung der ATO-OB umgesetzt, sowie die Gegenseite in der Fahrzeugsteuerung.

Abschließend wurden die Informationen der Systemarchitektur für die Anpassung der Fahrzeugsimulation zur Verfügung gestellt. Die Fahrzeugsimulation stellt den Systemen an einem Teststand relevante Eingaben zur Verfügung, damit diese die geforderte Funktion umsetzen können.

1.5.7. Erweiterung der MBPLE-Produktlinie um GoA2

Das Teilsystem GC3 dient der Integration der ATO-OB in die Fahrzeugsteuerung. Im Rahmen des MBPLE-Projekts wurde es prototypisch umgesetzt, gemäß den Vorgaben des Standards ERTMS /ATO-OB. Um die Integration der ATO-OB strategisch wiederverwendbar zu machen, wurde es zentral in der Wurzel der Produktlinienhierarchie der Fahrzeugsteuerung (Plattform

Basic Train Control, BTC) umgesetzt und von dort aus in die MBPLE-Produktlinien via MBPLE-Methoden integriert.

Für GC3 wurde dafür ein Feature Tree im BTC erzeugt (siehe Abbildung 36).



Abbildung 36 Feature-Baum zu GC3

Es wurde außerdem die Architektur von GC3 ausdetailliert (Verhalten und Struktur, siehe Abbildung 37 und Abbildung 38), sowie Feature Realizations definiert, die im Bezug zu den Architekturelementen stehen.

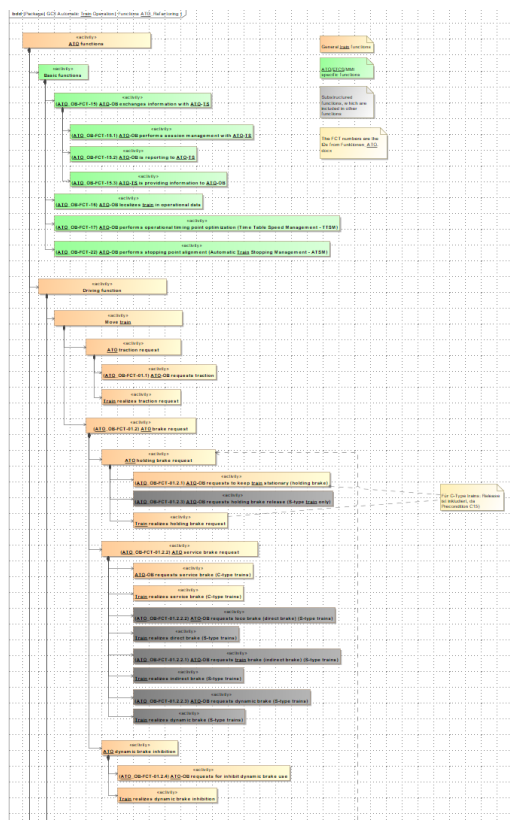


Abbildung 37 Ausschnitt der Verhaltensübersicht des Teilsystems GC3

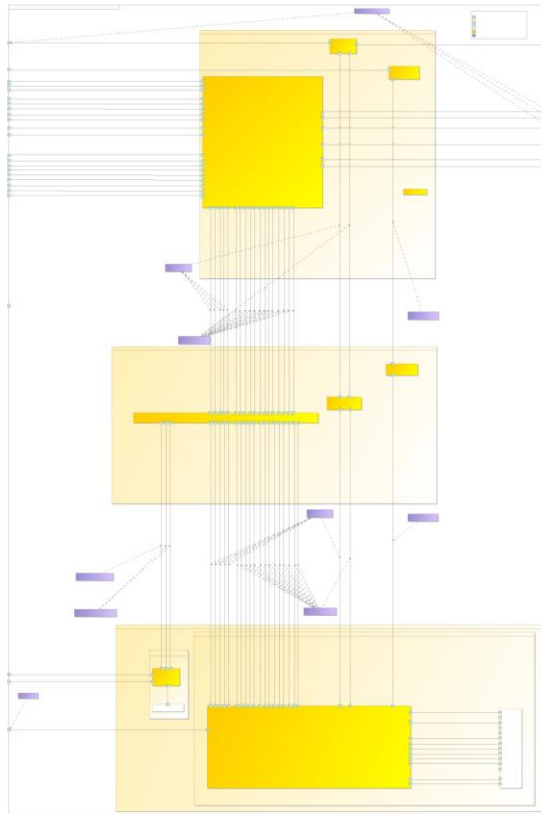


Abbildung 38 Ausschnitt der Struktursicht des Teilsystems GC3

Der Code für die GC3-Softwarekomponenten (SWCs) wurde ebenfalls im BTC implementiert und als Bibliothek zur Verfügung gestellt. Die SWC-Hüllen (Schnittstellendefinitionen) wurden mit dem Codegenerator auf Basis des Modells erzeugt. Die Logik wurde von Hand umgesetzt.

Anschließend wurde das Teilsystem GC3 aus dem BTC initial in die MBPLE-Produktlinie integriert. Bei der initialen Integration wurden die GC3-Modellelemente aus dem BTC in die entsprechenden GC3-Modelle der MBPLE-Produktlinie übertragen (merge). Dabei bleibt die UUID der Elemente vorhanden, sodass diese zu einem späteren Zeitpunkt zwischen BTC und MBPLE-Produktlinie abgeglichen werden können und somit zukünftige Updates ermöglichen. Danach wurden gemäß der Feature-Konfiguration in der MBPLE-Produktlinie alle obsoleten GC3-Architekturelemente entfernt. Somit stand in der MBPLE-Produktlinie eine vollständige, konfigurierte Architektur des Teilsystems GC3 zur Verfügung.

Abschließend wurden in der MBPLE-Produktlinie für das Teilsystem GC3 die Softwarekomponenteninstanzen generiert und die dazugehörige BTC-Bibliothek eingebunden. Somit stand eine erste Version der wiederverwendeten Umsetzung des Teilsystems GC3 in der MBPLE-Produktlinie zur Verfügung und die Fahrzeugsteuerungsapplikation konnte gebaut werden.

Aktualisierungen der Architektur wurden über einen erneuten Abgleich der Modelle vom BTC in die MBPLE-Produktlinie übertragen. Änderungen am Code der Softwarekomponenten wurden als neue Version der GC3-Bibliothek vom BTC zur Verfügung gestellt.

Das Vorgehen erwies sich als sehr effizient. Obwohl ein Update Zeit in Anspruch nimmt, überwiegt der Vorteil einer zentralen Entwicklung des Teilsystems GC3. Bspw. lässt sich der Stand mit den Methoden des MBPLE ohne Schwierigkeiten in andere Produktlinien innerhalb der Produktlinienhierarchie integrieren. Damit steht die Umsetzung allen Fahrzeugsteuerungs-Plattformen und -Projekten zur Verfügung. Ebenso führt die zentrale Entwicklung schneller zu

einem höheren Reifegrad der Software, da Feedback nur an einer Stelle eingearbeitet wird. Dies spricht sehr stark für die Anwendbarkeit der MBPLE-Methoden.

Abschließend sei noch erwähnt, dass zu den wiederverwendbaren Entwicklungsartefakten auch die Unit-Tests der Softwarekomponenten zählen. Die Unit-Tests wurden einmalig im BTC geschrieben und erfolgreich durchgeführt, was Abbildung 39 verdeutlicht. Dadurch konnte frühzeitig eine hohe Qualität des Codes gewährleistet werden. Außerdem wurden Aufwände reduziert, da die Unit-Tests nicht in den Produktlinien wiederholt werden mussten. Integrations- und Systemtests müssen jedoch in allen Produktlinien durchgeführt werden.

Test Result

0 failures (0%) 43 tests (100%)
Total: 0.08 sec.

All Tests Add description

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
features.GC3_AtoCobOn_SC	35 ms	0		0		3		3	
features.GC3_AtoTrIn_SC	12 ms	0		0		1		1	
features.GC3_AtoTrIn_SC	3 ms	0		0		1		1	
features.GC3_AtoTrIn_SC	0.26 sec	0		0		15		15	
features.GC3_AtoTrIn_SC	0.51 sec	0		0		21		21	
features.GC3_REAL_TO_WORD_FC	21 ms	0		0		2		2	

Abbildung 39 GC3 Unit-Testergebnisse

1.5.8. Umsetzung der ProfiNet-Anbindung des ATO-OB

Parallel zur Entwicklung des Fahrzeugsteuerungsanteils für GoA2 mit den Methoden des MBPLE startete die Umsetzung der PN-Schnittstelle des ATO-OB durch ein anderes Team. Um die Schnittstelle möglichst frühzeitig zu testen, wurde dafür ein eigener Teststand für Integrationstests aufgebaut (siehe Abbildung 40 und Abbildung 41).

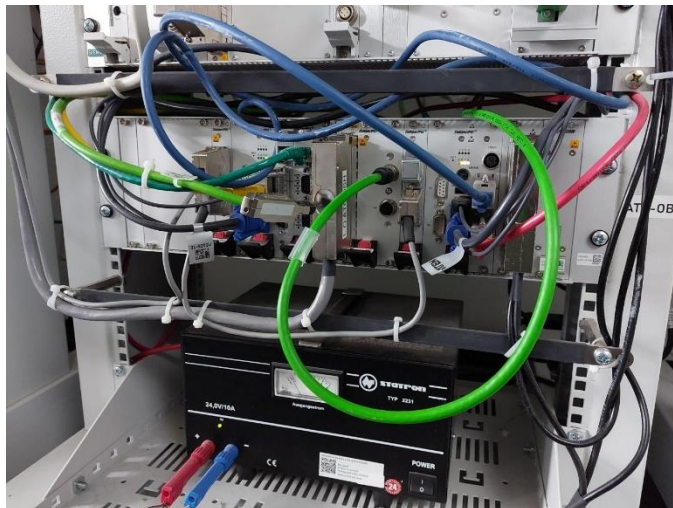


Abbildung 40 Detailansicht Teststand für ATO-OB Integrationstests

Ziel war es, sicherzustellen, dass die Datenpakete erfolgreich über ein PN-Netzwerk zwischen der ATO-OB und dem Fahrzeugsteuerungsrechner (SP CS) gemäß der Schnittstellendefinition ausgetauscht werden können.



Abbildung 41 ATO-OB Integrationsteststände

Die Tests wurden erfolgreich abgeschlossen. Änderungen an der Schnittstelle wurden in der Fahrzeugsteuerung zentral im BTC eingepflegt und konnten über ein Update der MBPLE-Produktlinie zur Verfügung gestellt werden.

1.5.9. Test der GoA2-Funktion an einem Teststand

Für den Abschluss des Arbeitspaketes 5 war ein manueller Test der GoA2-Betriebsszenarios an einem Teststand geplant. Dafür sollte die Fahrzeugsteuerungsapplikation auf eine reale SP CS geladen werden und mit einem echten ATO-OB und ETCS vernetzt werden. Mittels einer Fahrzeugsimulation sollten fahrzeugrelevante Daten (Pneumatik, Fortbewegungsdaten) den Systemen bereitgestellt werden. Eine Streckensimulation sollte statische Streckendaten und dynamische Fahrplandaten zur Verfügung stellen, damit das GoA2-Betriebsszenario durchgeführt werden kann.

Da die Vorbereitungen zu dem Aufbau des Teststands und insbesondere dessen Inbetriebnahme sehr umfangreiche Arbeitspakete sind, wird auf die einzelnen Arbeitspakete im Detail eingegangen. Den Beginn macht die erarbeitete, rudimentäre Testspezifikation für den manuellen Test des GoA2-Betriebsszenarios. Danach folgt eine Beschreibung des Teststands, der in 4-facher Ausführung aufgebaut wurde. Anschließend wird die Inbetriebnahme der Fahrzeugsteuerungssoftware auf dem SP CS besprochen. Danach folgt die Inbetriebnahme der ATO-OB. Den Abschluss der Inbetriebnahmen macht das ETCS. Es wird noch auf die Simulation der Strecke eingegangen, bevor ein Fazit zum Test und insbesondere zur Inbetriebnahme das Kapitel zu AP5 abschließt.

1.5.9.1. Testspezifikation für manuelle Tests

Für den manuellen Test des GoA2-Betriebsszenarios wurde folgende grobe Testspezifikation aufgesetzt:

Testfall 1	Kommunikation SP CS <-> ETCS über angepasste Schnittstelle
Testziel(e)	SP CS und ETCS sind erfolgreich gestartet und haben sich am ProfiNet-Ring angemeldet und Statusdaten gesendet.

Testfall 2	Kommunikation SP CS <-> ATO-OB über PN-Schnittstelle
Testziel(e)	SP CS und ATO-OB sind erfolgreich gestartet und haben sich am ProfiNet-Ring angemeldet und Statusdaten gesendet.

Testfall 3	Kommunikation der Streckendaten zwischen ATO-TS <-> ATO-OB
Testziel(e)	ATO-TS und ATO-OB sind erfolgreich gestartet und die Streckendaten wurden erfolgreich vom ATO-TS zum ATO-OB übertragen.

Testfall 4	Kommunikation TrackSim (RBC) <-> ETCS mit Streckendaten
Testziel(e)	ETCS und RBC-Kommunikationsanteil der Streckensimulation sind erfolgreich gestartet und RBC-Daten wurden erfolgreich ausgetauscht.

Testfall 5	Kommunikation Engagement Button <-> ETCS <-> ATO-OB <-> SP CS
Testziel(e)	Alle Systeme sind erfolgreich hochgefahren (SP CS, ETCS, ATO-OB, ATO-TS, Fahrzeug- und Streckensimulation) und die Streckendaten sind verfügbar. Das Fahrzeug wurde in Fahrbereitschaft gebracht, sodass das GoA2-Betriebsszenario gestartet werden kann (Start-of-Mission) und das System während der simulierten Fahrt entlang der simulierten Strecke beobachtet werden kann.

1.5.9.2. Teststandaufbau

Der Teststand besteht aus vielen Komponenten. Der Aufbau ist eine komplexe Aufgabe. Abbildung 42 zeigt eine logische Sicht auf den Aufbau mit den wichtigsten Komponenten (SP CS, ETCS, ATO-OB, ATO-TS, Simulation).

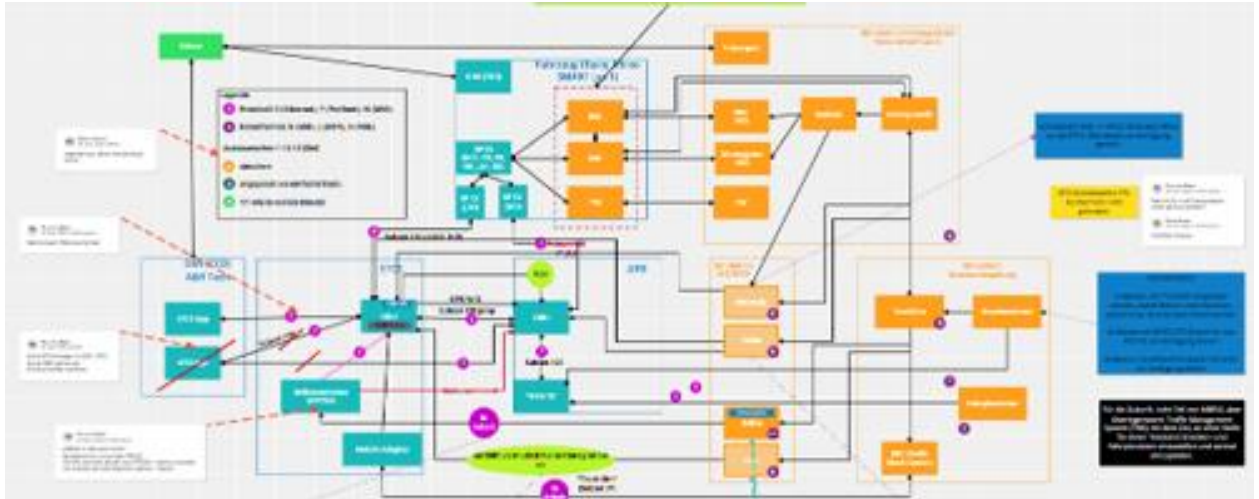


Abbildung 42 Logische Sicht auf die Anteile des Teststands

Um diese Komplexität beherrschbar zu machen, wurde der Teststandaufbau in zwei Phasen unterteilt:

- 1) Die erste Phase fokussierte sich auf den Fahrzeugsteuerungsanteil (SP CS) und die Fahrzeugsimulation.
- 2) Die zweite Phase befasste sich mit dem Aufbau des ETCS und der ATO-OB und ATO-TS und dessen Integration in das bestehende System aus Phase 1

Die Komponenten für die erste Phase konnten trotz der Corona-Pandemie und der damit einhergehenden weltweiten Lieferprobleme fast ausschließlich mit kurzer Wartezeit geliefert werden. Aufgrund der vorherrschenden Erfahrungen mit dem Aufbau von Testständen für die Fahrzeugsteuerung konnten die Bauteile für Phase 1 zeitnah eingebaut und verkabelt werden. Abbildung 43 zeigt den anfänglichen Zustand des Teststands aus Phase 1.

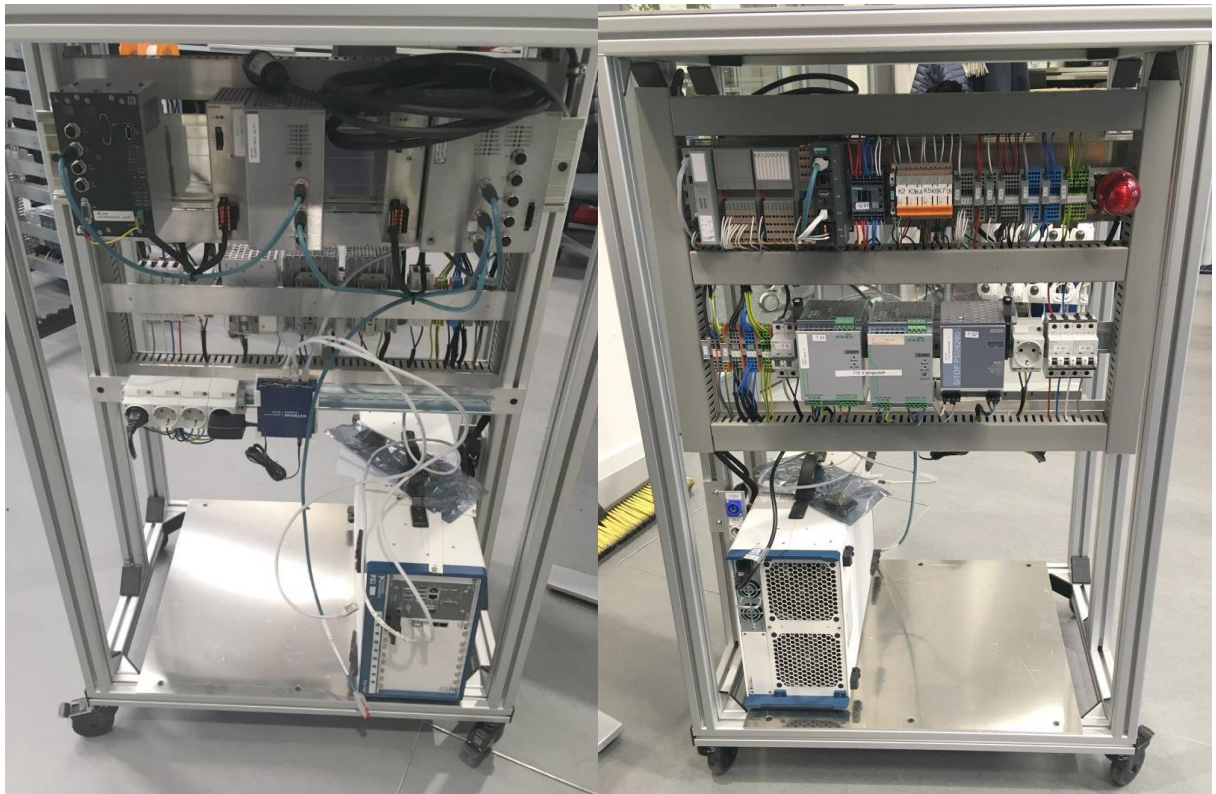


Abbildung 43 Aufbau Teststand Phase 1

Die Lieferverzögerungen aufgrund der Corona-Pandemie haben insbesondere die Komponenten der 2. Phase (ETCS, ATO-OB, ATO-TS) betroffen. Es musste bis zu einem Jahr gewartet werden, bis die Komponenten vor Ort waren und der Aufbau durchgeführt werden konnte. Daher ergaben sich Verzögerungen. Abbildung 44 zeigt den noch nicht verdrahteten Teststands rack aus Phase 2.



Abbildung 44 Aufbau Teststand Phase 2 - unverdrahtet

1.5.9.3. Inbetriebnahme Fahrzeugsteuerung

Für die Inbetriebnahme des Fahrzeugsteuerungsanteils sind ein Teststand nach Phase 1 und die Fahrzeugsteuerungssoftware erforderlich. Der Teststands Aufbau nach Phase 1 und die Herleitung/Erstellung der Fahrzeugsteuerungssoftware wurden bereits in den vorherigen Kapiteln erläutert. In diesem Zusammenhang relevant sind noch die Stromlaufpläne. Sie werden für die Erstellung der Fahrzeugsimulation benötigt und hängen mit der Fahrzeugsteuerungssoftware zusammen. Für die GoA2-Umsetzung wurden die Stromlaufpläne der Ausgangsbasis wiederverwendet und erweitert. Allerdings erwiesen sich die Anpassungen komplizierter als ursprünglich angenommen. Die Stromlaufpläne konnten aber schließlich fertiggestellt werden. Allerdings ergaben sich erneut Verzögerungen. Abbildung 45 zeigt einen Ausschnitt aus den angepassten Stromlaufplänen.

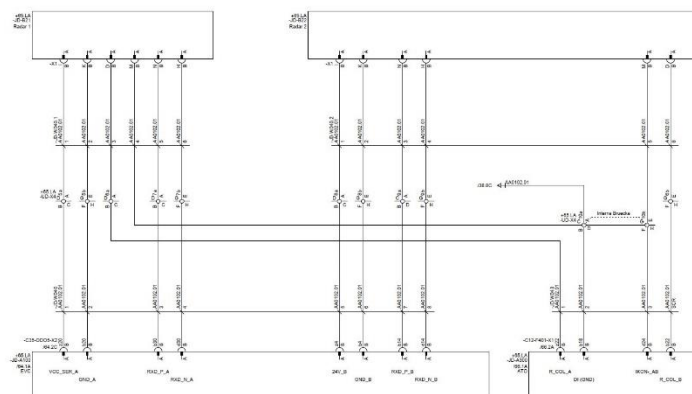


Abbildung 45 Ausschnitt aus angepasstem Stromlaufplan für GoA2

Zur Inbetriebnahme der Fahrzeugsteuerung wurde schließlich die Fahrzeugsteuerungssoftware erfolgreich auf den SP CS im Teststand geladen und mit der angepassten Fahrzeugsimulation eingeschaltet und hochgefahren. Über virtuelle Bedienfelder konnte der Triebzug erfolgreich angeschaltet und in Fahrbereitschaft versetzt werden. Eine Fahrt mit Fahrzeugsimulation war möglich.

1.5.9.4. Inbetriebnahme ATO-OB und ATO-TS

Für die Inbetriebnahme des ATO-OB ist ein Teststand nach Phase 2 notwendig. Aufgrund von Lieferverzögerungen und Verzögerungen beim Aufbau und der Verdrahtung der ATO-OB und des ETCS, war der Teststand nach Phase 2 erst zu einem verspäteten Zeitpunkt bereit. Die SW-Lieferung für das ATO-OB und auch die ATO-TS war zu dem Zeitpunkt schon vorhanden. Dadurch konnte unmittelbar nach Bereitstellung mit dem Laden der ATO-OB-Software begonnen werden. Die Inbetriebsetzung verlief erfolgreich. Allerdings konnte beim Test ein Problem bei der Anmeldung des ATO-OB im PN-Ring festgestellt werden. Da vorherige Integrationstests durch das ATO-OB-Team erfolgreich verlaufen sind, musste eine tiefgreifendere Analyse durchgeführt werden. Aufgrund der fortgeschrittenen Projektlaufzeit und der anstehenden Inbetriebsetzung des ETCS musste die Analyse des Problems verschoben werden. Abbildung 46 zeigt eine noch nicht angeschlossene ATO-OB.

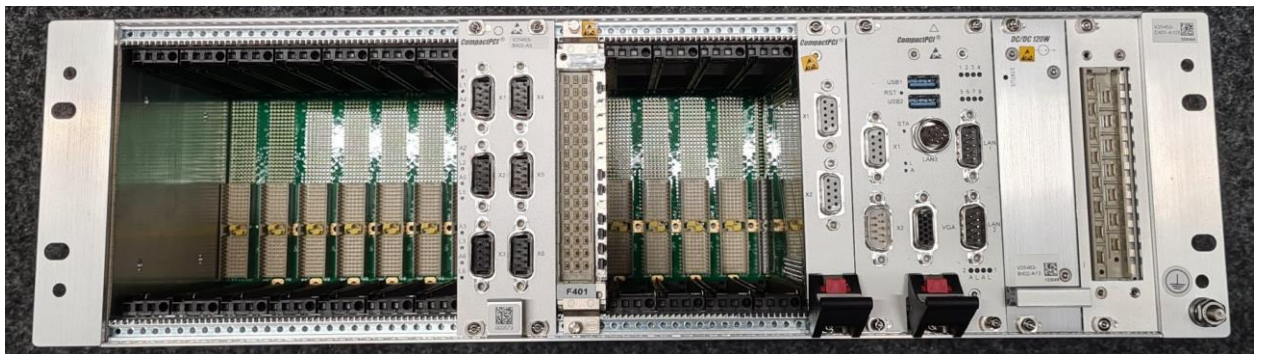


Abbildung 46 ATO-OB

Die ATO-TS wird auf in einer virtuellen Umgebung (VM) installiert und über Ethernet mit der ATO-OB verbunden. Die Streckendaten werden über eine definierte Schnittstelle der ATO-TS zur Verfügung gestellt. Es wurden Daten zu einer ETCS Level 2 Strecke zur Verfügung gestellt.

1.5.9.5. Inbetriebnahme ETCS

Das ETCS besteht im Wesentlichen aus zwei Komponenten: Dem European Vital Computer (EVC) und dem Non-Vital Computer (NVC). Beide Komponenten werden an das PN-Netzwerk angeschlossen. Die Inbetriebnahme erfolgt ausschließlich über den NVC.

Während der Inbetriebnahme ergaben sich diverse Probleme, welche die ursprüngliche, starke Verzögerung durch die späte Lieferzeit weiter verschärft haben. Es musste bspw. der NVC getauscht werden, da ein Defekt vermutet wurde. Darüber hinaus konnte eine wichtige Steckkarte nicht vor Ort aktualisiert werden. Sie musste dafür eingeschickt werden. Letztendlich ist es gelungen, den NVC zu installieren und zu konfigurieren und anschließend den EVC zu installieren und zu konfigurieren. Allerdings ergaben sich bei der Inbetriebsetzung noch Verbindungsprobleme: Der EVC war nicht im PN-Netzwerk sichtbar. Aufgrund der durch die Verzögerungen stark fortgeschrittenen Projektlaufzeit, konnte nicht mehr rechtzeitig die



Abbildung 49 Aufgebauter GoA2 Teststand

Die Anwendbarkeit der MBPLE-Methoden konnte allerdings trotzdem erfolgreich nachgewiesen werden. Eine formale Spezifikation der Architektur mittels MBSE gekoppelt mit der Beherrschung der Variabilität mittels Feature Trees aus dem Bereich des PLE erwiesen sich als geeignete Methoden, um effizient und zielgerichtet Produktlinien zu entwickeln. Mit den MBPLE-Methoden lassen sich lauffähige, stabile Produktlinien zügig aufbauen. Darüber hinaus wird eine zentrale Entwicklung von Funktionen wie bspw. GoA2 ermöglicht, sowie deren systematische Integration in unterschiedliche Produktlinien. Dadurch lassen sich Entwicklungskosten senken und es kann schneller eine höhere Qualität erreicht werden. Dies trägt dazu bei, dass Funktionen wie GoA2, welche einen positiven Einfluss auf den CO²-Verbrauch haben, schneller eine breitere Anwendung finden.

Die im Projekt erzielten Ergebnisse können auch für weitere Projekte eingesetzt werden. Dazu zählen nicht nur die für GoA2 erarbeiteten Anforderungen, Architekturen und Schnittstellen, sondern auch die Teststände. Es wurden insgesamt 4 Teststände aufgebaut und verdrahtet. Obwohl sie leider nicht im Rahmen des Projekts in Betrieb gesetzt werden konnten, sind sie in nachfolgenden Projekten im Einsatz. Wir sind damit in der Lage, die Entwicklungszeiten und die Produktzyklen zu verkürzen und die Varianzen in den Plattformen besser zu beherrschen.

1.6. Eignungsprüfung (AP7)

In dem Forschungsprojekt MBPLE konnten während der Projektlaufzeit die Anwendbarkeit der MBPLE-Methoden erfolgreich nachgewiesen werden.

Auch erwiesen sich formale Spezifikation der Architektur mittels MBSE gekoppelt mit der Beherrschung der Variabilität mittels Feature Trees aus dem Bereich des PLE als geeignete Methoden, um effizient und zielgerichtet Produktlinien zu entwickeln.

Thema	TRL zu Projektbeginn	TRL zu Projektende
Durchgehende (Anforderungen, Architektur, Simulation und Test) MBPLE Umsetzung von GoA2 im Schienenverkehr	3	5
Parallele Einführung von komplexen Innovationen in drei Fahrzeugplattformen nachgewiesen	3-4	5
Durchgängiges MBPLE für den gesamten Prozess von den Anforderungen bis zur Validierung	3	5

Tabelle 3: Eignungsprüfung SMO

Mit diesem Ergebnis ist Siemens Mobility in der Lage, die Entwicklungszeiten und die Produktzyklen zu verkürzen und die Varianzen in den Plattformen besser zu beherrschen.

1.7. Projektmanagement (AP8)

Arbeitspaket 8 umfasst das Projektmanagement, welches sich über die gesamte Projektlaufzeit erstreckte.

1.7.1. Zusammenarbeit im Konsortium

Über die Projektlaufzeit fand eine ausgesprochen gute und produktive Zusammenarbeit der Konsortialpartner statt. Konsortialtreffen fanden alle 4-6 Wochen statt, themenspezifische Meetings im zwei Wochen Takt. Die Treffen fanden online und in Präsenz statt.

1.7.2. Meilensteine

Über die Projektlaufzeit fand eine ausgesprochen gute und produktive Zusammenarbeit der Konsortialpartner statt. Konsortialtreffen fanden alle 4-6 Wochen statt, themenspezifische Meetings im zwei Wochen Takt. Die Treffen fanden online und in Präsenz statt.

Die folgenden Meilensteine waren geplant und erfolgreich umgesetzt.

Meilenstein	Beschreibung	Termin	Status
M1	Wesentlich Konzepte wurden erarbeitet. Proof of Concept und Werkzeugentwicklung können gestartet werden.	Monat 6	Erreicht
M2	Proof of Concept ist erbracht.	Monat 12	Erreicht
M3	Werkzeuge und methodische Aspekte sind bereit für den Beginn des Nachweises.	Monat 18	Erreicht
M4	Konzepte sind überarbeitet.	Monat 30	Erreicht
M5	Start der Eignungsprüfung.	Monat 36	Erreicht
M6	Die Arbeiten des Projektes sind abgeschlossen. Die definierten Ziele für die Nachweise für die Erreichung der TRL-Ziele wurden erbracht (TRL = Technology Readiness Level)	Monat 42	Erreicht

Tabelle 4: Übersicht Meilensteine

Alle Meilensteine wurden erfolgreich erreicht.

Bei allen Meilensteinen traf sich das Konsortium vor Ort bei einem anderen Konsortialpartner. Hier konnten die Ergebnisse gut dem TÜV und den anderen Konsortialpartner präsentiert werden. Neben dem fachlichen Austausch blieb auch immer genug Zeit für Vorstellung der Arbeit der austragenden Partner und dem persönlichen Austausch.

2. Wichtigste Positionen des zahlenmäßigen Nachweises

Auf Grund der Corona-Pandemie und dem damit verschobenen Reise- und Meetingverhalten sind in Summe deutlich weniger Reisen angefallen als ursprünglich geplant. Im Projektverlauf haben sich für die Arbeitsmeetings und Konsortialtreffen Teams-Meetings als sehr produktiv rausgestellt. Bei den Meilensteintreffen waren persönliche Treffen zielführend. Sowohl in der Präsentation der Arbeitsergebnisse, als auch für den persönlichen Austausch.

Während des Projekts hat sich herausgestellt, dass die Teststände deutlich teurer waren als ursprünglich kalkuliert. Der Grund dafür sind zum Teil extrem gestiegene Preise der Chips und der Einzelkomponenten.

Den höchsten Anteil an den Projektkosten haben die Personalkosten eingenommen. Gefolgt von den Kosten für Fremdleistungen. Die externen Mitarbeiter konnten uns mit Know-How langfristig unterstützen, welches im normalen Projektbetrieb nicht zwingend benötigt wird. Somit gab es einen Wissenstransfer, welches Siemens Mobility auch langfristig für die folgenden Projekte weiter zur Verfügung steht.

In Summe konnte das Projekt aber erfolgreich im geplanten abgewickelt werden und der Kostenrahmen eingehalten werden.

3. Notwendigkeit und Angemessenheit der geleisteten Arbeit

Das Forschungsprojekt MBPLE4Mobility ermöglichte es Siemens Mobility, gezielt Prozessverbesserungsthemen zu bearbeiten, die im operativen Tagesgeschäft mit Kundenprojekten sonst nicht in diesem Umfang hätten verfolgt werden können. Durch die Projektarbeit wurde nicht nur internes Wissen aufgebaut, sondern auch die methodische Kompetenz innerhalb des Unternehmens gestärkt – ein Vorteil, der sich bereits positiv auf die Akquise neuer Kundenprojekte ausgewirkt hat.

Die im Projekt entwickelten Methodenbeschreibungen und Software-Werkzeuge fließen direkt in den Entwicklungsprozess von Siemens Mobility ein. Dadurch lassen sich sowohl die Qualität als auch der Grad der systematischen Wiederverwendung von Entwicklungsartefakten deutlich steigern.

Diese Fortschritte wären ohne die Förderung durch das Bundesministerium für Wirtschaft und Klimaschutz (BMWK) in diesem Umfang nicht realisierbar gewesen.

4. Voraussichtlicher Nutzen

Siemens Mobility setzt weiter auf die Erkenntnisse von MBPLE4Mobility. Dazu gehört eine permanente Toolentwicklung, dauerhafte Weiterentwicklung der methodischen Konzepte sowie der Simulation und den Testständen. Dies hat sich im Projektverlauf als nützlich erwiesen. Der GoA2 Betrieb wird von vielen Kunden gefordert – hier hat die Vorarbeit von MBPLE4Mobility wertvolle Dienste geleistet. Ein Beispiel dafür ist die S-Bahn München, welche im teilautomatischen Fahrbetrieb ab 2028 Passagiere transportieren soll.

Die Verwaltung der unterschiedlichen Züge einer Produktlinie wird aktuell der Plattform Mireo weiter angewandt und weiterentwickelt.

Mit der Entwicklung der voll autonomen Züge, dem GoA4 Betrieb, werden die Grundlagen des GoA2 Betriebs konsequent weiterentwickelt. Hier sieht Siemens, aber auch die Kunden wie die Deutsche Bahn, großes Potential in den nächsten Jahren. Mit der Entwicklung wird versucht, Probleme wie z.B. den Fahrermangel entgegenzuwirken. Auch die Dekarbonisierung und das energieeffiziente Fahren wird so gefördert. Zusammen mit weiteren Konsortialpartnern ist Siemens in dem Forschungsprojekt „Automated Train“ beteiligt. Ein weiteres Forschungsprojekt

„Remote Train“, bei dem Triebfahrzeuge über eine Leitwarte ferngesteuert werden können, ist in Planung.

5. Update Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Über den Fortschritt der methodischen Entwicklung in anderen Forschungsprojekten, Universitäten oder Konkurrenten der Bahntechnik von Siemens Mobility ist nichts bekannt. Siemens stellt aber fest, dass der Wunsch vom teil- oder vollautonomen Fahren bei den Kunden stetig an Präsenz gewinnt und zunehmend auch gefordert wird. Die Gründe dafür sind vielseitig: z.B. Fahrermangel, energieoptimiertes Fahren, bessere Einhaltung der Fahrpläne und Entlastung der Fahrer während der Fahrt.

6. Erfolgte und geplante Veröffentlichungen des Ergebnisses

Das Projekt wurde im Rahmen der Konferenz TdSE (Tag des System Engineerings) in den Jahren 2022 (Paderborn) und 2023 (Würzburg) vorgestellt. Das Vorhaben und die methodische Umsetzung stießen auf große Aufmerksamkeit. Weitere Veröffentlichungen sind vorstellbar und sinnvoll, aber der Zeitpunkt ist momentan noch nicht bekannt.