

KI-basiertes, modulares Assistenzsystem für funktionale Systemarchitekturen

S. Schleifer, A. Lungu, B. Kruse, L. Schwarz, S. Goetz, S. Wartack

Getrieben durch eine steigende Produktkomplexität gewinnt Model-Based Systems Engineering (MBSE) zunehmend an Bedeutung. Der Einsatz von künstlicher Intelligenz (KI) hat das Potential in der Systemmodellierung Zeit- und Kosteneinsparungen zu realisieren. Voraussetzung hierfür ist jedoch eine durchgängige KI-Integration in den Modellierungsprozess. In diesem Beitrag wird ein modulares Assistenzsystem vorgestellt, dass Natural Language Processing (NLP) und Large Language Modelle (LLMs) in die manuelle Verhaltensmodellierung funktionaler Systemarchitekturen integriert. Die Teilmodule folgen der zugrunde liegenden Modellierungsmethode und sind vollständig in das Modellierungswerkzeug eingebunden. Der modulare Aufbau erlaubt eine gezielte Nutzung der KI-Unterstützung und integriert den Systementwickler frühzeitig, um die Qualität der KI-generierten Modellbestandteile zu gewährleisten.

AI-based, modular assistance system for functional system architectures

Driven by increasing product complexity, Model-Based Systems Engineering (MBSE) is becoming increasingly important. The use of artificial intelligence (AI) has the potential to deliver time and cost savings in system modelling. However, this requires end-to-end AI integration into the modelling process. This article presents a modular assistance system that integrates natural language processing (NLP) and large language models (LLMs) into the manual behaviour modelling of functional system architectures. The sub-modules follow the underlying modelling method and are fully integrated into the modelling tool. The modular structure allows for the targeted use of AI support and involves the system developer at an early stage to ensure the quality of the AI-generated model components.

1 Einleitung und Motivation

Steigende Systemkomplexität getrieben durch fortschreitende Kundenansprüche und Veränderungen im Marktumfeld, gepaart mit immer kürzer werdenden Entwicklungszyklen, sind zentrale Herausforderungen in der Automobilindustrie. Damit einher geht eine Vielzahl an Systemanforderungen, die in einem Fahrzeugprojekt realisiert werden müssen. Model-Based Systems Engineering (MBSE) ermöglicht die Handhabung dieser Systemkomplexität. Die damit verbundenen, üblicherweise manuellen Modellierungstätigkeiten sind jedoch besonders zu Beginn zeitintensiv. Daher werden eine Automatisierung und Unterstützung mittels künstlicher Intelligenz (KI) angestrebt. Damit lassen sich natürlichsprachliche Anforderungen effizient verarbeiten und automatisch in formale Modelle überführen, was zu Zeitersparnissen und erhöhter Konsistenz in der Entwicklung komplexer Systeme führen kann. Vor diesem Hintergrund wird in diesem Beitrag ein Assistenzsystem zur Unterstützung und Beschleunigung von manuellen Modellierungstätigkeiten im MBSE vorgestellt.

2 Stand der Technik

Der praktischen Anwendung von MBSE liegt der Dreiklang aus Modellierungssprache, -methode und -werkzeug zugrunde [1]. Die Systems Modeling Language (SysML) ist eine weitverbreitete Sprache in der Systementwicklung [2] und bietet Diagramme für die Anforderungs-, Struktur- und Verhaltensmodellierung. Letztere wird genutzt, um Abläufe im Systemmodell zu beschreiben. Darunter fallen das Use Case Diagramm zur Beschreibung von Funktionalitäten, das Aktivitätsdiagramm für die Ablaufbeschreibung, das Sequenzdiagramm für Kommunikationsabläufe und der Zustandsautomat für eine zustandsbasierte Modellierung [3].

Eine verbreitete Modellierungsmethode ist das V-Modell (VDI 2206) [4], das die Entwicklung mechatronischer Systeme unterteilt in die Systemdekomposition (linker Schenkel des „V“), die disziplinspezifische Implementierung (Spitze des „V“) und die Systemintegration (rechter Schenkel des „V“). Unterstützt werden die Abläufe durch die Validierung und Verifizierung [4]. Für komplexe technische Systeme erfolgt eine Aufteilung in unterschiedliche Systemebenen, um so die Entwicklung vom Gesamtsystem über Subsysteme zu den Komponenten hin zu gliedern. Innerhalb der Systemebenen wird der RFLP-Ansatz genutzt, bei dem man sich über Anforderungssicht (R), funktionaler Sicht (F), logischer Sicht (L) und physikalischer Sicht (P) unter Berücksichtigung einer Referenzarchitektur schrittweise einer Lösung nähert [5]. Der Übergang von der R- zur F-Sicht beinhaltet die Formalisierung natürlichsprachlicher Anforderungen mittels Verhaltensmodellen.

Natural Language Processing (NLP), als Teilgebiet der künstlichen Intelligenz (KI), kann hierbei maßgebend unterstützen. Zentrale Methoden sind das Erkennen von Wortarten (POS-Tagging), die Identifikation bestimmter Satzelemente (Named-Entity Recognition) und die Erstellung von Embeddings. Dies sind vektorisierte Wörter oder Sätze, die entweder über statische oder selbstlernende Modelle erzeugt werden [6]. Mit dem Aufkommen von Large Language Modellen (LLMs) hat sich die Leistungsfähigkeit der Sprachverarbeitung stark erweitert, was auch für das MBSE vielversprechende Möglichkeiten bietet [7]. Im Gegensatz zu bisherigen NLP-Ansätzen zeichnen sich LLMs durch ihre allgemeine Anwendbarkeit aus. Generative LLMs sind in der Lage neuen Text auf Basis bereitgestellter Eingaben (Prompts) zu erzeugen. [6]. Die Erstellung geeigneter Prompts nennt man Prompt Engineering. Darunter fällt das One-/Few-Shot Prompting, bei dem das LLM zusätzlich zur Aufgabe ein oder mehrere Beispiele erhält, um die Ausgabe zu verbessern [8]. Beim Prompt

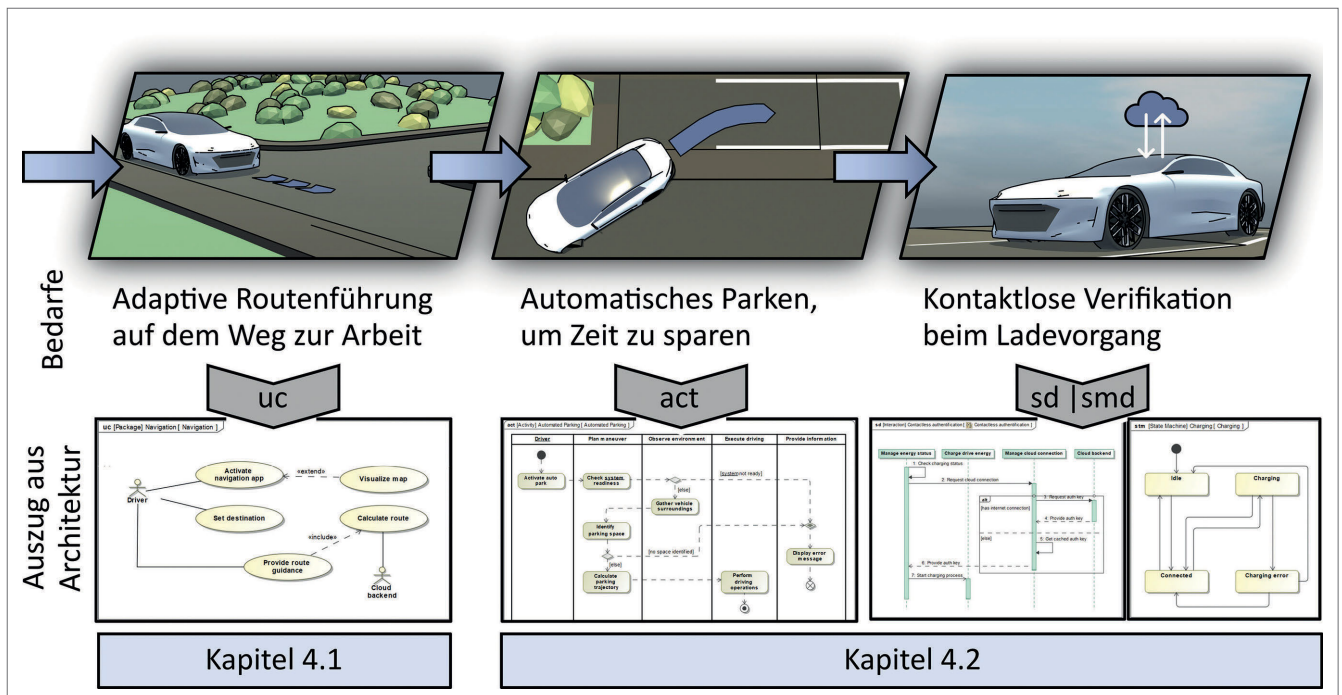


Bild 1: Auszugsweise Darstellung einer exemplarischen User Journey in der Automobilentwicklung. uc: Use Case Diagramm, act: Aktivitätsdiagramm, sd: Sequenzdiagramm, smd: Zustandsautomat. Bild: Lehrstuhl für Konstruktionstechnik (KTmfk)

Chaining wird eine Aufgabe in mehrere Prompts unterteilt, um deren Komplexität zu reduzieren [9].

Der erfolgreiche Einsatz von NLP in der Anforderungsmodellierung und SysML-Diagrammerstellung konnte bereits in einigen Forschungsarbeiten gezeigt werden. Beispielsweise können Embeddings und Clustering Algorithmen genutzt werden, um Use Cases aus User Stories zu extrahieren und über einen regelbasierten NLP Ansatz in Use Case Diagramme zu überführen [10]. Deep Learning Techniken können genutzt werden, um einfache Aktivitätsdiagramme zu erstellen [11]. Gupta et al. [12] stellen einen NLP Ansatz für die Erstellung von Zustandsautomaten vor [12]. Für Sequenz- und Use Case Diagramme wurde bereits gezeigt, dass sich diese durch LLMs in besserer Qualität erzeugen lassen als durch NLP [13, 14]. Die Nutzung von LLMs zeigt auch für Zustandsautomaten vielversprechende Ergebnisse [15]. Generell können Tätigkeiten im MBSE durch den Einsatz von LLMs und Prompt Engineering unterstützt werden [7].

3 Forschungsproblem

Aus dem Stand der Technik wird ersichtlich, dass bisherige Ansätze für die automatische Systemmodellerstellung einzelne Aspekte der Architekturentwicklung nur isoliert betrachten. Damit bleibt auch die Einbindung existierender Ansätze in eine durchgängige Modellierungsmethode unberücksichtigt, was sich insbesondere an der fehlenden Abstimmung der Ein- und Ausgänge aufeinanderfolgender Ansätze bemerkbar macht. Gleichermaßen muss die Einbindung des Menschen berücksichtigt werden, um die KI-Unterstützung effektiv in bestehende Arbeitsabläufe zu integrieren. Zusätzlich treffen bestehende Ansätze vereinfachende Annahmen wie durchgängig homogene Anforderungsspezifikationen [16]. Insofern sollten unterstützende Ansätze folgende Anforderungen berücksichtigen:

- Verarbeitung unstrukturierter Anforderungsspezifikationen

- Erstellung initialer sowie Ergänzung bestehender Systemmodelle
- Anpassbarkeit an Änderungen in Modellierungsmethode oder Anwendungsdomäne

Um die aufgezeigte Forschungslücke zu adressieren, wird in dieser Veröffentlichung die folgende Forschungsfrage beantwortet: Wie muss ein Assistenzsystem zur automatisierten Erstellung von funktionalen Systemarchitekturen aufgebaut sein, um Systementwickler effektiv unterstützen zu können?

Die Beantwortung der Forschungsfrage wird über ein praktisches Anwendungsbeispiel illustriert, das auf eine exemplarische User Journey aus der Automobilentwicklung zurückgreift. Wie in **Bild 1** dargestellt, umfasst die User Journey die Nutzung von Navigationsfunktionen, um einen Zielort anzusteuern. Dort angekommen möchte der Nutzer das Fahrzeug verlassen und einen fahrerlosen Parkvorgang einleiten. Abschließend soll das batterieelektrische Fahrzeug kabellos geladen werden. Für die Produktentwicklung lassen sich daraus Anforderungen ableiten, die die Grundlage für eine Produktarchitektur liefern. Dem Beispiel folgend ergeben sich hieraus mehrere Kundenfunktionalitäten, gegliedert in Features, die technisch umgesetzt werden müssen.

4 Assistenzsystem zur KI-unterstützten Architekturmodellierung im MBSE

Zur Beantwortung der Forschungsfrage wird ein modulares Assistenzsystem vorgestellt, das die Modellierung der vier Verhaltensdiagramme der SysML durch eine KI-Integration unterstützt, siehe **Bild 2**. Der Fokus dieses Beitrags liegt auf der methodischen Verknüpfung einzelner Ansätze zu einem durchgängigen Assistenzsystem. Der modulare Aufbau wird gewählt, um Teilaufgaben der modellbasierten funktionalen Systemarchitekturentwicklung nach Diagrammart und Teilaufgabe zu separieren und dadurch die inhärente Komplexität zu handhaben.

Bild 2: Methodischer Aufbau des modularen Assistenzsystems zur KI-unterstützten Entwicklung funktionaler Architekturen im MBSE. Bild: Lehrstuhl für Konstruktionstechnik (KTmfk)

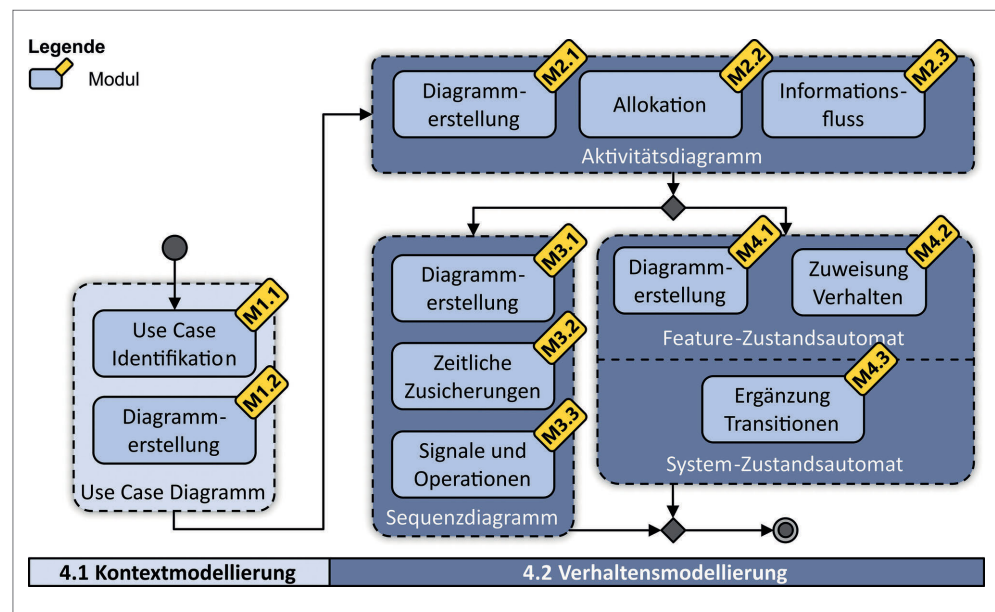
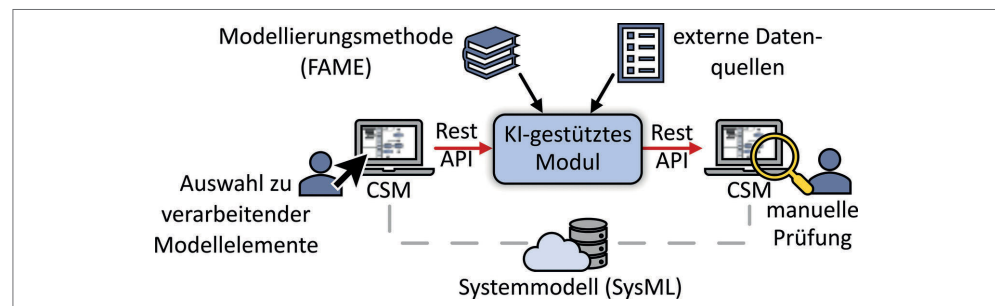


Bild 3: Einbindung eines Moduls in die Entwicklungsumgebung. Bild: Lehrstuhl für Konstruktionstechnik (KTmfk)



Die Erstellung der funktionalen Systemarchitektur, welche auf der Kontext- und Verhaltensbeschreibung basiert, profitiert durch die Anwendung des modularen Assistenzsystems unter anderem durch Zeitersparnisse in der Modellerstellung, da Anforderungsspezifikationen effizient in SysML-Diagramme überführt werden können. Die Modellgenerierung zielt darauf ab Architekturentscheidungen, wie die Einteilung in Subsysteme und die Zuweisung zu Funktionen, zu treffen. Darüber hinaus wird eine erhöhte Modellkonsistenz erreicht, da subjektive Einflüsse auf die Modellierung reduziert und die Wiederverwendung bestehender Modellinhalte begünstigt werden. Die Qualität der KI-generierten Modellinhalte wird in der Praxis durch eine manuelle Prüfung der generierten Modellelemente und Diagramme nach jedem Modulaufruf gewährleistet, siehe **Bild 3**. Dem Nutzer steht das Assistenzsystem als Plugin in Cameo Systems Modeler (CSM) [17] zur Verfügung. Dadurch werden die Eingabedaten des Modulaufbaus, wie Anforderungsspezifikationen und zu detaillierende Modellelemente (z.B. Use Cases), ausgewählt. Die KI-Unterstützung ist über eine RestAPI in das Plugin integriert. Als Sprachmodell wird „gpt-4o“ von OpenAI [18] in einer privaten Instanz eingesetzt. Jedem Modul liegen zusätzlich Schritte einer unternehmensspezifischen Modellierungsmethode (FAME) zugrunde, die in das Prompt Engineering einfließen. Ein vereinfachtes Meta-Modell der Modellierungsmethode findet sich in **Bild 4**. Nachfolgend werden die Module im Kontext des Assistenzsystems vorgestellt, wobei deren detaillierte Umsetzung den referenzierten Veröffentlichungen zu entnehmen ist. Dort sind zudem die individuellen Forschungsfragen der Module zu finden.

4.1 Kontextmodellierung

Zu Beginn der Architekturentwicklung steht die Kontextmodellierung, deren Ziel die Beschreibung aller Use Cases (Funktionalitäten), des zu entwickelnden Systems of Interest (SOI) ist. Dabei werden neben Interaktionen zu Akteuren und externen Systemen außerhalb des SOI auch Beziehungen zwischen Use Cases im Use Cases Diagramm modelliert. [19].

In der Praxis ergibt sich für jedes Feature eine Vielzahl an heterogenen Anforderungen, die über Use Cases gegliedert werden. Die Anforderungen werden beispielsweise aus Vorgängerprojekten und User Journeys ermittelt. Aufgrund unterschiedlicher Granularität der Anforderungen können einzelne Anforderungen zu mehreren Use Cases beitragen. In der praktischen Anwendung entsteht durch die Sichtung der Anforderungsspezifikationen ein hoher manueller Aufwand. Das Assistenzsystem beschleunigt diesen Vorgang über ein Modul für die Use Case Identifikation (M1.1, Bild 2). Eine detaillierte Beschreibung findet sich in [20]. Der Nutzer ruft für ein Feature einen graphbasierten Clustering-Algorithmus auf. Die Anforderungen werden über Embeddings formalisiert und über gewichtete Kanten verbunden. Deren Gewicht setzt sich aus einem funktionalen Ähnlichkeitswert und expliziten Beziehungen zusammen. Ein überlappender Clustering Algorithmus identifiziert die Use Cases durch funktional zusammenhängende Anforderungen. Die manuelle Prüfung wird durch die Ableitung von Keyphrases erleichtert. Weitere Betrachtungen zeigen, dass die Use Case Identifikation vom Einsatz von LLMs durch eine stärkere funktionale Zusammengehörigkeit und besse-

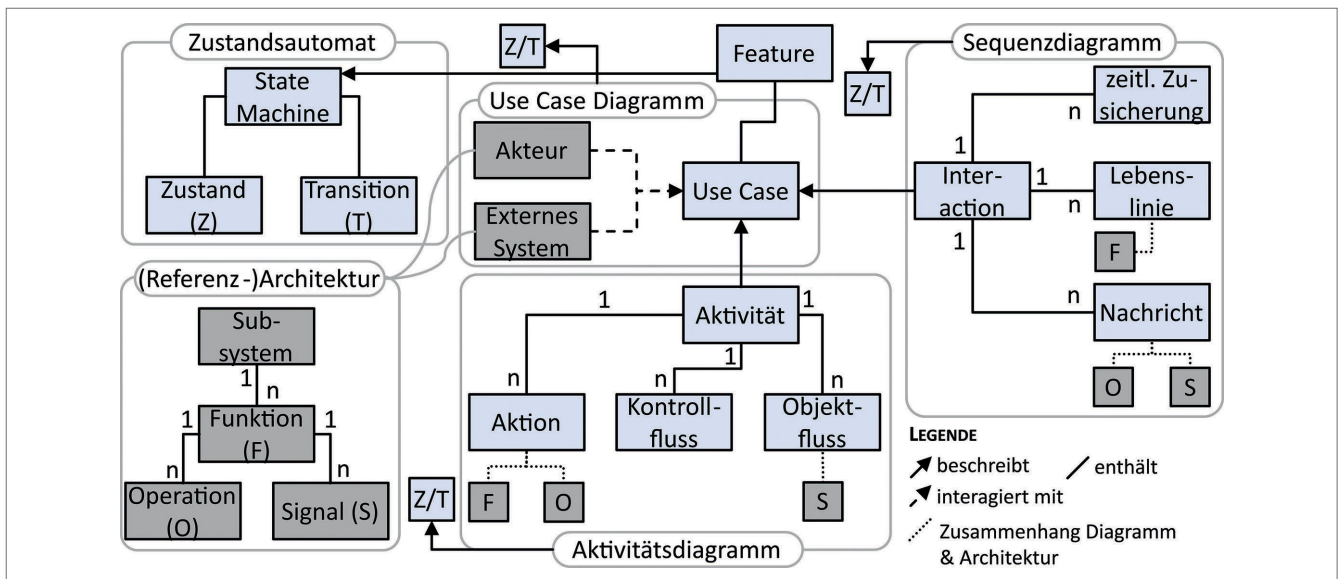


Bild 4: Vereinfachtes Meta-Modell der eingesetzten Modellierungsmethode. Bild: Lehrstuhl für Konstruktionstechnik (KTmfk)

re Verständlichkeit für den Entwickler profitiert. Für das exemplarische Feature „Navigation“ bedeutet das, dass aus der Anforderungsspezifikation fünf Use Cases identifiziert werden. Diese werden über folgende vorläufige Keyphrases beschrieben: „Activate the navigation app“, „Visualize the map“, „User sets destination“, „Compute the route to destination“ und „Guides driver on route“.

Anschließend erfolgt die Ergänzung zum vollständigen Use Case Diagramm für das Feature basierend auf den heterogenen Anforderungen. In der Praxis ist insbesondere mit der vollständigen Beschreibung der Use Cases, unter anderem durch Vor- und Nachbedingungen sowie Auslöser, ein hoher manueller Aufwand verbunden. Darüber hinaus müssen Akteure und Beziehungen konsistent mit den Use Cases verknüpft werden.

Der User Journey folgend, muss zum Beispiel der Fahrer mit dem vorläufigen Use Case „Guides driver on route“ interagieren. Dabei unterstützt Modul M1.2 mit der Detaillierung der einzelnen zuvor identifizierten Use Cases über One-Shot Prompts. Hierbei werden ein prägnanter Use Case Name und eine Dokumentation ergänzt. Dem Beispiel folgend werden diese Use Case Namen generiert (siehe auch Bild 2): „Activate navigation app“, „Visualize map“, „Set destination“, „Calculate route“ und „Provide route guidance“. Die Definition der Vor- und Nachbedingungen sowie Trigger kann durch kuratierte Auswahlmöglichkeiten limitiert werden, um die Ergebnislänge zu verbessern. Die automatische Identifikation der Akteure respektiert die Referenzarchitektur, um das Systemmodell konsistent zu halten. Ferner nutzt das Modul alle Use Cases des Features, um mittels Prompt Chaining Beziehungen zwischen diesen zu identifizieren und das resultierende Use Case Diagramm zu generieren. Abschließend werden die Use Cases im Systemmodell aktualisiert und das KI-generierte Diagramm importiert, welches durch den Nutzer nötigenfalls angepasst und anschließend freigegeben werden kann. [16]

4.2 Verhaltensmodellierung

Der nächste Schritt der Architekturentwicklung ist die Beschreibung interner Abläufe der einzelnen Use Cases über Aktivitätsdiagramme. Dabei werden Aktionen sowie deren logische Abfolge definiert [19]. Für die funktionale Architekturentwicklung

ist die Zuordnung von Aktionen zu Funktionen von großer Bedeutung. Dadurch wird festgelegt welche Ablaufschritte die jeweiligen Funktionen umsetzen müssen. Zusätzlich wird der Informationsaustausch zwischen den Funktionen über Informationsflüsse modelliert.

In der praktischen Anwendung ist die Modellierung der initialen Aktivitätsdiagramme zeitintensiv, da die heterogenen Anforderungen gesichtet und in einen Ablauf überführt werden müssen, der die gewünschte Kundenfunktionalität (z.B. aus der User Journey) abbildet. Hierbei unterstützt das Assistenzsystem durch das Modul M2.1, welches die Grundstruktur des Aktivitätsdiagramms für einen manuell ausgewählten Use Case generiert. Das Modul nutzt die vollständig ausgearbeitete Use Case Beschreibung und die verknüpften Anforderungen als Eingangsdaten. Eine Prompt Chain gliedert die darauf basierende Diagrammerstellung in die Definition der Aktionen und des Kontrollflusses mittels Few-Shot Prompting durch eine textuelle Beschreibung des Diagramms. Dabei werden die umfassenden semantischen Fähigkeiten des LLMs genutzt, um die Modellelemente und deren Beziehungen aus den Anforderungen zu extrahieren. Nachgelagert werden Funktionsaufrufe durch das LLM genutzt, um potenziell auftretende Syntaxfehler aufzulösen [21].

Der User Journey folgend, soll der interne Ablauf des Use Cases „Perform auto park“ durch ein Aktivitätsdiagramm beschrieben werden. Aus den Anforderungen identifizierte Aktionen umfassen unter anderem das Aktivieren der Parkunterstützung, die Prüfung der Sensorverfügbarkeit, Ermittlung des Fahrzeugumfeldes sowie die Pfadplanung und Durchführung der Fahraufgabe. Entscheidungsknoten werden auf Basis der Anforderungen automatisch generiert, um im Fehlerfall den Fahrer zu informieren und den Ablauf des Use Cases abzubrechen.

Anschließend erfolgt das Zuweisen der Subsysteme und deren Funktionen, die an dem Use Case beteiligt sind. Dazu ist die Berücksichtigung der Referenzarchitektur und bestehender Modellelemente essenziell, um Inkonsistenzen im Systemmodell zu vermeiden. Das Modul M2.2 automatisiert diese Aufgabe innerhalb des Aktivitätsdiagramm effizient durch das Nutzen einer Vektordatenbank in Verbindung mit Prompt Chaining. Die Datenbank umfasst Embeddings der Operationen aller Funktionen des Sys-

temmodells und erlaubt die Extraktion von Kandidatenlisten für die Zuweisung von Operationen zu Aktionen. Daraus lassen sich die zu allozierenden Funktionen ableiten und in das Diagramm aufnehmen. Wenn keine passende Operation existiert, werden verbleibende Aktionen über nachgelagerte Prompts auf Funktionen alloziert [21]. Beispielsweise wird die Aktion „Gather vehicle surroundings“ der Operation „Scan vehicle surroundings“ zugeordnet und damit zur Funktion „Observe environment“ alloziert.

Auch die Modellierung des Informationsflusses ist in der praktischen Anwendung durch eine zeitaufwendige und fehleranfällige Suche und Wiederverwendung bestehender Modellelemente, hier Signale, gekennzeichnet. Das Modul M2.3 unterstützt durch die automatische Generierung des Informationsflusses zwischen Aktionen. Die Signale definieren Art und Inhalt des Informationsaustauschs. Das LLM generiert auf Basis des Aktionsnamens einen Signalvorschlag, der mit einer Vektordatenbank abgeglichen wird. Wenn ein passendes, existierendes Signal identifiziert wird, wird der Informationsfluss damit typisiert. Andernfalls wird der initiale Vorschlag des LLMs zur manuellen Prüfung vorgelegt [21]. Der User Journey folgend wird zwischen den Aktionen „Gather vehicle surroundings“ und „Identify parking space“ ein Signal ausgetauscht, das Umgebungsinformationen als Paket zusammenfasst.

Wenn das Verhalten eines Use Cases besonderen Fokus auf Kommunikationsabläufe setzt, beispielsweise die Authentifizierung an einer Ladesäule (siehe User Journey), kann ergänzend oder anstelle des Aktivitätsdiagramms ein Sequenzdiagramm modelliert werden [19], siehe Bild 2.

Ähnlich wie beim Aktivitätsdiagramm kann in der Praxis von einem schnellen Einblick in die Verhaltensabläufe profitiert werden. Zudem müssen die Verhaltensdiagramme untereinander konsistent gehalten werden. Daher automatisiert das Assistenzsystem die Sequenzdiagrammerstellung für einen Use Case (M3.1, Bild 2) über Prompt Chaining. Zunächst werden Lifelines und Nachrichten generiert. Sofern ein Aktivitätsdiagramm besteht, werden diese hieraus übernommen. Lifelines und Nachrichten werden über Few-Shot Prompting zum Sequenzdiagramm zusammengeführt und beispielsweise um bedingte Abläufe ergänzt.

Hierzu wird erneut eine textuelle Beschreibung des Diagramms generiert. Im Rahmen der User Journey wird die Kommunikation zwischen der Ladesäule und dem Fahrzeug zum Starten des Ladevorgangs modelliert. Die Funktionen „Manage energy status“, „Charge drive energy“ und „Manage cloud connection“ aus dem Architekturmodell sowie das externe „Cloud system“ werden als Lifelines genutzt. Exemplarische Nachrichten sind das Anfragen nach einem Authentifizierungsschlüssel aus dem Cloud System. Falls keine Internetverbindung besteht, wird eine lokal gespeicherte Kopie genutzt.

Die konsistente Einbindung zeitlicher Zusicherungen ist in der Praxis ein weiterer manueller Arbeitsschritt. Dieser kann durch das Modul M3.2 beschleunigt werden. Dabei extrahiert eine Prompt Chain zunächst zeitliche Bedingungen aus den Anforderungen und integriert diese anschließend konsistent in das bestehende Sequenzdiagramm. Dem Beispiel folgend muss die Zeitdauer zwischen dem Start der Kommunikation und dem Beginn des Ladevorgangs auf 3 s begrenzt werden.

Im Sequenzdiagramm muss beschrieben werden, ob Nachrichten Informationen (Signale) tragen, oder Verhaltensaufrufe (Operationen) anstoßen. Um in der Praxis Doppelungen zu vermeiden, müssen die bestehenden Modellelemente intensiv geprüft

werden, was mit hohem manuellem Aufwand verbunden ist. Das Assistenzsystem (M3.3) vereinfacht diese Tätigkeit, durch die Verwendung der Vektordatenbanken für Signale und Operationen in Verbindung mit Prompt Engineering Techniken. Basierend auf einem Embedding des Namens der Nachricht wird über die Vektordatenbank eine Kandidatenliste erstellt. Ein Prompt bestehend aus einer detaillierten Beschreibung der Nachricht sowie den Namen und Dokumentationen der Kandidaten entscheidet, ob ein Modellelement aus der Kandidatenliste wiederverwendet werden kann. Wenn kein funktionaler Zusammenhang zwischen den Kandidaten und der Nachricht besteht, wird ein neues Signal bzw. Operation erstellt. Durch die manuelle Prüfung erfolgt eine Absicherung der Ergebnisse. Der User Journey (siehe Bild 1) folgend, ruft die Lifeline „Manage energy status“ über eine Nachricht die Operation „Start charging process“ in der Lifeline „Charge drive energy“ für die kontaktlose Verifizierung mit der Ladeinfrastruktur auf. Zuletzt unterstützt das Assistenzsystem bei der Erstellung von Zustandsautomaten, wenn das Verhalten eines Features unterschiedliche Zustände umfasst [19], siehe Bild 2.

Die manuelle Modellierung des Feature Zustandsautomaten ist erneut mit einem erheblichen zeitlichen Aufwand verbunden, da alle Use Cases des Features betrachtet sowie die übrigen Verhaltensdiagramme konsistent gehalten werden müssen. Diese Tätigkeit erleichtert das Assistenzsystem, über die automatische Generierung eines Zustandsautomaten (M4.1). In der Kombination aus Prompt Chaining und Few-Shot Prompting werden im Feature zunächst Zustände und Transitionen basierend auf den Use Cases und deren Beschreibung identifiziert und anschließend durch eine generierte textuelle Darstellung des Diagramms zusammengeführt. Über eine folgende Simulation kann die Vermeidung von Syntaxfehlern und somit die Ausführbarkeit des generierten Zustandsautomaten sichergestellt werden, da Fehlermeldungen der Simulation als zusätzlicher Input für das Prompting verwendet werden können. Das Modul M4.2 automatisiert mittels Prompt Engineering die Zuordnung von Verhaltensdiagrammen zu den Zuständen, beispielsweise beim Betreten eines Zustandes. Ebenso werden dadurch die Transitionen im Rahmen von Auslösern, Ereignissen und Bedingungen detailliert. Ein einfacher Zustandsautomat des Features „Charging“ umfasst beispielsweise die Zustände „Idle“, „Charging“, „Connected“ und „Error“. Transitionen, wie die Authentifizierung, ermöglichen den Wechsel vom Grundzustand „Idle“ in den „Connected“ Zustand.

Abschließend müssen die Auswirkungen der bisherigen Modellierung auf den Zustandsautomat des Gesamtsystems geprüft werden. Die manuelle Durchführung birgt aufgrund der Vielzahl von Einflussfaktoren Fehlerpotenzial. Das Modul M4.3 des Assistenzsystems unterstützt durch die automatisierte Ergänzung um Transitionen und Zustände basierend auf den Vor- und Nachbedingungen der Use Cases des Features über geeignete Prompt Engineering Techniken. Zusätzlich werden Zustände des Gesamtsystems als Bedingungen in das Feature ergänzt. So muss sich beispielsweise das Fahrzeug in einem aktiven Zustand befinden, damit der Ladevorgang gestartet werden kann.

5 Zusammenfassung und Ausblick

Das modulare Assistenzsystem ermöglicht eine effiziente und qualitativ hochwertige Einbindung in die Systemarchitekturentwicklung, wie es beispielsweise für das Use Case Diagramm in einer ersten Nutzerstudie bereits gezeigt werden konnte [16]. Die

damit verbundene Arbeitsentlastung legt den Grundstein für kürzere Entwicklungszyklen, was insbesondere in der Automobilproduktion in Wettbewerbsvorteilen resultieren kann. Die separaten Module für die Unterstützung der Kontext- und Verhaltensmodellierung funktionaler Systemarchitekturen sind aufeinander abgestimmt. Beispielsweise wird die KI-generierte Use Case Beschreibung für die Erstellung der Aktivitäts- und Sequenzdiagramme genutzt, was eine durchgängige Verknüpfung ermöglicht und dem methodischen Vorgehen entspricht. Der modulare Aufbau gewährleistet eine einfache Anpassung an geänderte Rahmenbedingungen sowie die erleichterte manuelle Prüfung der KI-generierten Modellinhalte, da ein schrittweiser Modellaufbau forciert wird. Abschließend ist hervorzuheben, dass durch die Nutzung von LLMs eine höhere Flexibilität und verbesserte semantische Fähigkeiten als in bestehenden NLP-basierten Ansätzen erreicht werden. Nichtsdestotrotz muss das modulare Assistenzsystem hinsichtlich seiner industriellen Anwendbarkeit noch intensiver in Nutzerstudien untersucht werden sowie die Anpassbarkeit an andere Anwendungsdomänen oder geänderte Modellierungsmethoden geprüft werden.

Danksagung: Die Autoren bedanken sich bei der Audi AG für die Unterstützung und Finanzierung des Forschungsprojekts

Literatur

- [1] Delligatti, Lenny: SysML Distilled: A Brief Guide to the Systems Modeling Language. 1. Auflage: Addison-Wesley Professional, 2013 — ISBN 978-0-321-92786-6
- [2] Ma, Junda; Wang, Guoxin; Lu, Jinzhi; Vangheluwe, Hans; Kiritsis, Dimitris; Yan, Yan: Systematic Literature Review of MBSE Tool-Chains. In: Applied Sciences Bd. 12, Multidisciplinary Digital Publishing Institute (2022), Nr. 7, S. 3431 — DOI 10.3390/app12073431
- [3] Weikens, Tim; Soley, Richard Mark: Systems Engineering mit SysML/UML: Anforderungen, Analyse, Architektur. 3., überarb. und aktualisierte Aufl. Heidelberg: dpunkt.verl, 2014 — ISBN 978-3-86490-091-4
- [4] VDI/VDE 2206:2021-11: Entwicklung mechatronischer und cyber-physischer Systeme. Berlin: Beuth
- [5] Krog, Jonas; Akbas, Caner; Nolte, Bastian; Vietor, Thomas: Development of a Functional and Logical Reference System Architecture in Automotive Engineering. In: Systems Bd. 13, Multidisciplinary Digital Publishing Institute (2025), Nr. 3, S. 141 — DOI 10.3390/systems13030141
- [6] Jurafsky, Daniel; Martin, James H.: Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models, 2024. — Online Manuskript veröffentlicht am 20.08.2024. Abrufbar über: <https://web.stanford.edu/~jurafsky/slp3/>. Abgerufen am 21.10.2024
- [7] Hovemann, Aschot; Bitá, Isaac Mpid ; Aldade, Abed Alrahman; Heißen, Oliver von; Dumitrescu, Roman: Prompt Engineering im Systems Engineering: Potenziale und Grenzen moderner großer Sprachmodelle im V-Modell. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb Bd. 120, De Gruyter (2025), Nr. s1, S. 101–106 — DOI 10.1515/zwf-2024-0139
- [8] Brown, Tom; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared D; Dhariwal, Prafulla; Neelakantan, Arvind; Shyam, Pranav; u.a.: Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems. Bd. 33: Curran Associates, Inc., 2020, S. 1877–1901
- [9] Wu, Tongshuang; Terry, Michael; Cai, Carrie Jun: AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI '22. New York, NY, USA : Association for Computing Machinery, 2022 — ISBN 978-1-4503-9157-3, S. 1–22
- [10] Kochbati, Takwa; Li, Shuai; Gérard, Sébastien; Mraidha, Chokri: From User Stories to Models: A Machine Learning Empowered Automation. In: Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development. Bd. 10: SCITEPRESS – Science and Technology Publications, 2021 — ISBN 978-989-758-487-9, S. 28–40
- [11] Zhu, R.; Li, W.; Jin, C.: TAG: UML Activity Diagram Deeply Supervised Generation from Business Textual Specification. In: IEEE International Conference on Software Analysis, Evolution and Reengineering, 2023, S. 956–961 — DOI 10.1109/SANER56733.2023.00116
- [12] Gupta, A.; Poels, G.; Bera, P.: Generating multiple conceptual models from behavior-driven development scenarios. In: Data and Knowledge Engineering Bd. 145 (2023) — DOI 10.1016/j.datak.2023.102141
- [13] Jahan, M.; Hassan, M.M.; Golpayegani, R.; Ranjbaran, G.; Roy, C.; Roy, B.; Schneider, K.: Automated Derivation of UML Sequence Diagrams from User Stories: Unleashing the Power of Generative AI vs. a Rule-Based Approach. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems: ACM, 2024, S. 138–148 — DOI 10.1145/3640310.3674081
- [14] Schleifer, Simon; Lungu, Adriana; Kruse, Benjamin; van Putten, Sebastiaan; Goetz, Stefan; Wartack, Sandro: Minimal Data, Maximal Impact: Language Model-based Pipelines for the Automatic Generation of Use Case Diagrams from Requirements. In: DS 133: Proceedings of the 35th Symposium Design for X (DFX2024): The Design Society, 2024, S. 240–249 — DOI 10.35199/dfx2024.25
- [15] Apvrille, L.; Sultan, B.: System Architects Are not Alone Anymore: Automatic System Modeling with AI. In: Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering. Bd. 1, 2024, S.27–38 — DOI 10.5220/0012320100003645
- [16] Schleifer, Simon; Lungu, Adriana; Kruse, Benjamin; Goetz, Stefan; Wartack, Sandro: Large Language Model-based Generation of Use Case Diagrams from Requirement Specifications. In: INCOSE International Symposium Bd. 35 (2025), Nr. 1, S. 261–275 — DOI 10.1002/iis2.70015
- [17] Cameo Systems Modeler. URL <https://www.3ds.com/products/catia/no-magic/cameo-systems-modeler>. — abgerufen am 15.10.2025. — Dassault Systèmes
- [18] Hello GPT-4o. URL <https://openai.com/index/hello-gpt-4o>. — aufgerufen am 03.09.2024 — OpenAI
- [19] Pohl, Klaus; Rupp, Chris: Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level. Heidelberg, Germany : dpunkt.verlag, 2021 — ISBN 978-3-96910-247-3
- [20] Schleifer, Simon; Lungu, Adriana; Kruse, Benjamin; Putten, Sebastiaan van; Goetz, Stefan; Wartack, Sandro: Use case identification of natural language system requirements with graph-based clustering. In: Design Science Bd. 11 (2025), S. e30 — DOI 10.1017/dsj.2025.10019
- [21] Schleifer, Simon; Lungu, Adriana; Kruse, Benjamin; Schwarz, Lea; Goetz, Stefan; Wartack, Sandro: Leveraging large language models to model use case behaviors via activity diagrams. Eingereicht in: Systems Engineering (derzeit im Review befindlich) (2025)

Simon Schleifer
schleifer@mfk.fau.de

Stefan Goetz
goetz@mfk.fau.de

Sandro Wartack
wartack@mfk.fau.de

Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Konstruktionstechnik, Erlangen

Adriana Lungu
adriana.lungu@audi.de

Benjamin Kruse
benjamin.kruse@audi.de

Lea Schwarz
lea.schwarz@audi.de

Audi AG, Ingolstadt

LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons Namensnennung 4.0 International (CC BY 4.0)

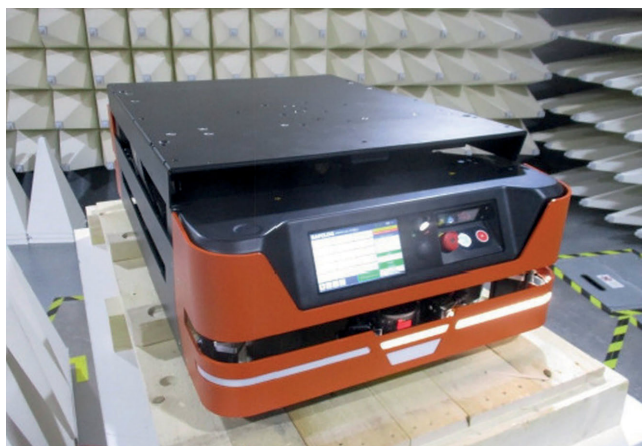
Safelog erhält UL 3100 Zertifizierung für mobile Roboter

Mehrere Geräteserien der Safelog GmbH erfüllen die strengen Vorgaben von Underwriters Laboratories (UL). Die Zertifizierung nach UL 3100 und CAN/UL 3100 eröffnet, wie das Unternehmen ausführt nicht nur neue Marktpotenziale in den USA und Kanada, sondern festigt auch die strategische Zusammenarbeit mit Partnerunternehmen wie TGW Logistics und SEW-Eurodrive.

Die UL 3100 Zertifizierung zählt zu den maßgeblichen nord-amerikanischen Standards für automatisierte mobile Plattformen. Sie bewertet die elektrische und mechanische Sicherheit eines Geräts umfassend. Dazu gehören unter anderem die sichere Auslegung von Batterie- und Ladesystemen sowie ein optimales Bremsverhalten. Zusätzlich müssen detaillierte Anforderungen an Gehäusekonstruktion und Materialstärken erfüllt werden sowie eine Beurteilung der Entflammbarkeit nach UL 94 erfolgen.

Ebenso werden sicherheitsrelevante Funktionen wie der Notstopp, die Hinderniserkennung und das Verhalten im Fall technischer Fehlfunktionen geprüft.

Nach den schon früher zertifizierten Modellen S3 und M4 absolvierte nun auch das Safelog L2 erfolgreich die Prüfungen und Tests. Mit der Zertifizierung erfüllen die drei Modelle nun sämtliche sicherheitsrelevanten Vorgaben für den Einsatz in den USA und Kanada. „Die Zertifizierung nach UL 3100 bietet Anwendern vor allem Rechtssicherheit und Risikominimierung, da die sicherheitstechnische Auslegung durch eine unabhängige Stelle geprüft wurde“, sagt Mathias Behounek, Geschäftsführer von Safelog. Für den Hersteller mobiler Roboter ist die Zertifizierung



Das Safelog L2 lift wurde vom TÜV Süd nach UL 3100 zertifiziert. Bild: Safelog GmbH

weit mehr als ein formaler Schritt. Sie schafft Transparenz hinsichtlich geprüfter Sicherheitsfunktionen, Materialanforderungen sowie Betriebsgrenzen und vereinfacht die Integration in industrielle Umgebungen. Vor allem auch für die Zusammenarbeit mit Partnern wie TGW Logistics und SEW-Eurodrive ergibt sich so eine klar definierte und normativ abgesicherte Sicherheitsbasis. Die UL 3100 reduziert den regulatorischen Abstimmungsaufwand und erleichtert die Inbetriebnahme auf dem Shopfloor.

www.safelog.de

Vorschau 11/12-2026



Damit ein Schaltschrank beim Kunden tatsächlich auf Antrieb funktioniert, muss er bereits beim Hersteller vollständig geprüft und für den Einsatz vorbereitet sein. Bild: Unitechnik

Schaltschrankbau

Wenn eine Maschine auf der Baustelle steht und der Schaltschrank nicht funktioniert, wird jede Stunde teuer. Dabei lässt sich ein Großteil dieses Risikos bereits beim Schaltschrankbauer eliminieren, lange vor der Auslieferung.

Ringsystem

Moderne Industrieanwendungen stellen immer höhere Anforderungen an die Verbindungstechnik. Der Beitrag zeigt, wie Schnellkupplungs-, Multikupplungs- und Dockingsysteme entwickelt, geprüft und validiert werden, damit sie auch unter extremen Bedingungen zuverlässig funktionieren.

Fachteil Ingenieur-Werkstoffe



Beim Extremen Hochgeschwindigkeits-Laserauftragschweißen EHLA wird Metallpulver im Laserstrahl aufgeschmolzen und als dünne Funktionsschicht auf das Bauteil aufgetragen. Foto: Fraunhofer ILT

WERKSTOFFFORSCHUNG

Warum metallisches Glas auf der ISS schweben muss

POLYMERE

Neue NMR-Analytik macht Materialveränderungen sichtbar

METALLKORROSION

Rost kostet Milliarden: Forscher entschlüsseln die Korrosion