

Identifikation von Automationsrisiken hochautomatisierter Fahrfunktionen in PEGASUS

Matthias Bükler, Birte Kramer¹, Eckard Böde, Sebastian
Vander Maelen, Martin Fränzle

OFFIS - Institut für Informatik

Escherweg 2, 26121 Oldenburg

¹birte.kramer@offis.de

Abstrakt

Die Identifikation von Automationsrisiken bedient sich einer iterativen erweiterten Sicherheitsanalyse. Sie dient dem Ziel, Ursachen für Risiken, die durch die Einführung hochautomatisierter Fahrfunktionen (HAF-Funktionen) entstehen frühzeitig zu identifizieren. Im Rahmen der Analyse werden insbesondere die für eine Gefährdung ursächlichen Umgebungsbedingungen identifiziert und in Form von Parameterbereichen logischer Szenarien (Bagschik, et al. 2017) beschrieben. Dies ermöglicht im späteren Testprozess ein zielgerichtetes Testen und belegt eine hinreichende Testfallabdeckung.

Das PEGASUS Projekt

Die hier beschriebene Methode zur Identifikation von Automationsrisiken wurde im Kontext des Projekts PEGASUS erstellt. PEGASUS ist das vom Bundesministerium für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestages geförderte **P**rojekt zur **E**tablierung von **g**enerell **a**kzeptierten Gütekriterien, Werkzeugen und Methoden sowie **S**zenarien und **S**ituationen zur Freigabe hochautomatisierter Fahrfunktionen. Ziel ist es, ein Vorgehen für das Testen automatisierter Fahrfunktionen zu

entwickeln, um so die rasche und sichere Einführung des automatisierten Fahrens in der Praxis zu ermöglichen.

Die deutsche Automobilindustrie ist sich einig, dass für die Absicherung und Freigabe von hochautomatisierten Fahrfunktionen ein einheitliches Vorgehen für das Testen und die Erprobung nötig ist. Daher haben sich im Projekt PEGASUS Automobilunternehmen, Zulieferer, kleinere und mittelständische Unternehmen sowie Forschungseinrichtungen zusammengeschlossen um bis Mitte 2019 für das Erproben und die Freigabe automatisierter Fahrfunktionen ein allgemein akzeptiertes und einheitliches Vorgehen zu entwickeln. Die 17 Projektpartner aus Wissenschaft und Wirtschaft definieren hierfür einen Stand der Technik zur Absicherung des hochautomatisierten Fahrens und demonstrieren die Entwicklung praxisnah anhand der Beispielanwendung Autobahn-Chauffeur, der das Fahren auf der Autobahn hochautomatisiert übernimmt. Dieses System übernimmt die Längs- und Querführung auf der Autobahn und ist in der Lage, die Grenzen seines Wirkbereichs zu erkennen und kann mit ausreichender Zeitreserve die Führungsaufgabe an den Fahrer übergeben. Weitere Informationen sind auch auf der Homepage <https://www.pegasusprojekt.de/de/> zu finden.

Motivation

Unter einem Risiko wird laut ISO 26262 die Kombination aus Schadensschwere und Eintrittswahrscheinlichkeit des Schadens verstanden. Basierend auf dieser Definition wurde in PEGASUS (PEGASUS, 2017) ein Automationsrisiko als ein solches Risiko definiert, bei dem der Schaden ursächlich durch die Automation, deren Verhalten oder deren Interaktion mit dem Fahrer entstanden ist. Um die unterliegenden Ursachen möglichst vollständig zu erfassen und zu bewerten, wurden die folgenden drei Klassen von Automationsrisiken identifiziert:

1. Auswirkungen des Umfelds auf die Automation.
2. Auswirkungen der Automation auf andere Verkehrsteilnehmer.
3. Interaktion des Fahrers mit der Automation und Einfluss des Verkehrsumfelds.

Klasse 1: Auswirkungen des Umfelds auf die Automation

In dieser Klasse werden Automationsrisiken untersucht, die auf Grund von äußeren Einflüssen auf die Automation entstehen. Dies sind also

Situationen, in denen die Automation Schwierigkeiten hat, mit der Umwelt zurechtzukommen. Hierzu gehört unter anderem die Nicht-Erkennung oder Fehlklassifikation von Objekten, die sich im Umfeld der Automation befinden, die Fehlerkennung von Objekten, die in der Realität gar nicht vorhanden sind, sowie die Fehlprädiktion der zukünftigen Entwicklung einer Situation.

Klasse 2: Auswirkungen der Automation auf andere Verkehrsteilnehmer

In dieser Klasse werden Risiken betrachtet, die aus der fehlerhaften Antizipation des Verhaltens der Automation durch andere Verkehrsteilnehmer entstehen. Eine hochautomatisierte Fahrfunktion würde beispielsweise eher konservativ und defensiv fahren, was jedoch andere menschliche Verkehrsteilnehmer irritieren und durchaus zu riskantem Verhalten verleiten könnte.

Klasse 3: Interaktion des Fahrers mit der Automation und Einfluss des Verkehrsumfelds

Automationsrisiken der Klasse 3 resultieren aus Szenarien in denen der Mensch mit der Automation interagiert. Dazu gehören klassischerweise Situationen, in denen eine *Mode Confusion* vorliegt oder in der die Funktion vorsätzlich missbraucht wird.

In diesem Dokument stehen die Automationsrisiken der Klasse 1 im Fokus, da die hier beschriebene Methode im Hinblick auf diese Klasse entwickelt wurde. Fehlklassifikationen oder Fehlerkennung von Objekten, die nach obiger Definition zu den Automationsrisiken der Klasse 1 gehören, spielen auch bei sich heute bereits im Einsatz befindlichen Fahrassistenzsystemen mit SAE Level 2 (Society of Automotive Engineers (SAE), 2010) eine Rolle. Es ist daher naheliegend sich mit den bisher für diese Probleme verwendeten Methoden zu beschäftigen und ihre weitere Anwendbarkeit zu prüfen.

Im derzeitigen Entwicklungsprozess nach ISO 26262 (International Organization for Standardization (ISO), 2009) sind die Sicherheitsanalyse und das Testen von Systemen fest verankerte Schritte. Insbesondere ist die Sicherheitsanalyse eine notwendige Voraussetzung für das Testen, da durch die Gefahren- und Risikoanalyse basierend auf der *Item Definition* und den so genannten *Operational Situations* die Sicherheitsziele abgeleitet werden. Die bisher eingesetzten Methoden um eine Gefahren- und Risikoanalyse

durchzuführen beziehen jedoch einige Herausforderungen, die hochautomatisierte Fahrfunktionen mit sich bringen, nicht mit ein.

Derzeitig auf dem Markt befindliche Fahrassistenzsysteme übernehmen lediglich vorübergehend die Kontrolle (SAE Level 2). Es ist daher wichtig, dass das System keine zusätzlichen Risiken, wie Eingreifen in Situationen, in denen dies nicht nötig ist, erzeugt (*false positives*). Der Fahrer ist als redundantes Kontrollsystem jederzeit verfügbar. Daher ist ein *Fail-safe* Sicherheitskonzept ausreichend. Hochautomatisierte Fahrfunktionen (SAE Level 3 und höher) entbinden den Fahrer jedoch temporär von der Verantwortung das Fahrzeug zu führen. Der Fahrer als jederzeit verfügbare Rückfallebene fällt somit weg. Daher ergibt sich hier neben der nötigen Zusicherung, dass das System nur eingreift, wenn dies nötig ist auch, dass ein solches System in allen Situationen korrekt und zum richtigen Zeitpunkt eingreift. Aus diesen Gründen spielen hier auch fehlende Reaktionen (*false negatives*) eine entscheidende Rolle für die Betriebssicherheit und ein *Fail-operational* Sicherheitskonzept wird benötigt.

Eine HAF-Funktion muss komplexe Umgebungen wahrnehmen und in diesen operieren können. Die Umwelt kann also nicht statisch (wie zuvor mit *operational situations*) beschrieben werden, da für das Verhalten die zeitlichen und dynamischen (reaktive) Aspekte eine zentrale Rolle spielen. Daher ist ein dynamischer Ansatz nötig um die komplexe Umwelt und deren zeitliche Abfolge modellieren zu können. Hierzu eignet sich ein Szenarien-basierter Ansatz, der auch in PEGASUS verfolgt wird.

Eine besondere Herausforderung für das Testen von hochautomatisierten Fahrfunktionen ist, dass Unfälle zwar relativ gut aus vorhandenen Daten rekonstruiert werden können, sich allerdings die Frage stellt, ob die Ursachen für kritische Szenarien für die Automation die gleichen sind, wie für den Menschen. So könnten beispielsweise Reflexionen von metallischen Objekten zu einer fehlerhaften Erkennung von Objekten durch einen RADAR-Sensor führen. Solche Ursachen für plötzliches Bremsen oder Ausweichen sind für Menschen nicht als Unfallursachen hinterlegt. Für hochautomatisierte Fahrfunktionen existieren derzeit noch keine umfangreichen Datenbanken zum beobachteten Unfallgeschehen, also sind mögliche kritische Szenarien nicht a priori bekannt.

Zusätzlich verhält sich die Kritikalität von Fahrfunktionen gegebenenfalls hochgradig unstetig in den Umgebungsparametern, weil Algorithmen diskrete Entscheidungen treffen, deren Schwellen insbesondere durch die Nutzung von *Machine Learning* Algorithmen nur teilweise bekannt sind. Beispielsweise können nur wenige, mit dem menschlichen Auge kaum sichtbare, Unterschiede in Bildern zu einer unterschiedlichen Objektklassifikation führen (siehe z.B. (Mian, 2017)). Daher müssen kritische Szenarien frühzeitig mit Hilfe von Expertenwissen identifiziert werden.

Prinzipiell könnte dieser Problematik mit einer vollständigen Charakterisierung aller Einsatzszenarien begegnet werden, was aber aufgrund der Größe und unscharfen Abgrenzung des Einsatzspektrums nicht möglich ist. Alle Situationen und Szenarien, die der HAF-Funktion in ihrem Leben begegnen werden, konkret zu beschreiben ist aussichtslos. Folglich ist es schwierig, das für die jeweilige Situation angemessene Verhalten vollumfänglich zu beschreiben. Daher kann die intendierte Funktionalität nicht als sicher und vollständig spezifiziert angenommen werden (SOTIF). Diese Problematik wird bis Level 2 auch schon vom ISO/PAS 21448 (International Organization for Standardization, 2019) behandelt.

Insgesamt ergibt sich somit die Notwendigkeit die bestehende Gefahren- und Risikoanalyse anzupassen, um die oben genannten Punkte möglichst frühzeitig zu adressieren. Mit diesem Ziel wurde die Methode zur Identifikation von Automationsrisiken entwickelt.

Methode zur Identifikation von Automationsrisiken

Die Automationsrisikenmethode ist eine iterative Methode um risikoauslösende Eigenschaften logischer Szenarien ausgehend von der Funktionsmodellierung und mit Hilfe einer Umgebungsmodellierung zu identifizieren. Wir werden hier einen beispielhaften Durchlauf der Methode vorstellen. Eine detailliertere Darstellung der Prozessschritte ist Gegenstand einer Folgepublikation.

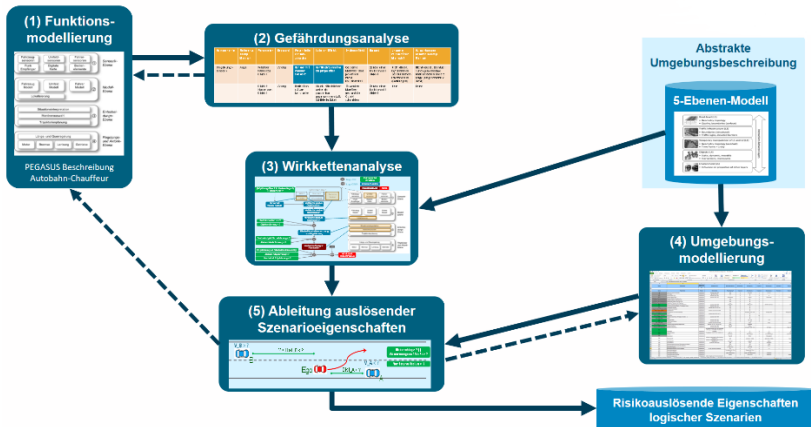


Abbildung 1: Überblick der Methode

In PEGASUS dient die Beschreibung des Autobahn-Chauffeurs als Fallbeispiel um die Anwendbarkeit der Methodik zu zeigen. Diese Beschreibung wurde innerhalb von PEGASUS erstellt und wird im ersten Schritt unserer Methode für einzelne Funktionskomponenten konkreter ausgestaltet. Abbildung 2 zeigt die verfeinerte Architektur des Autobahn-Chauffeurs. Der linke Teil stellt im Wesentlichen die in der Beschreibung des Autobahn-Chauffeurs enthaltene Grobarchitektur dar (ergänzt um Datenfluss zwischen den einzelnen Komponenten, der hier nicht dargestellt ist). Auf der rechten Seite sind die von uns vorgenommenen Verfeinerungen der Umfeldsensoren, der Heckkamera und der Situationsinterpretation dargestellt. Bei den Umfeldsensoren wurde also die Existenz verschiedener Sensoren angenommen (wie z.B. ein Front-LIDAR und eine Frontkamera). Weiterhin nehmen wir eine Sensorfusion an, wobei diese nicht näher ausgestaltet wurde. Wir gehen daher zunächst davon aus, dass Probleme einzelner Sensoren zu einer fehlerhaften Umweltwahrnehmung des Gesamtsystems führen können.

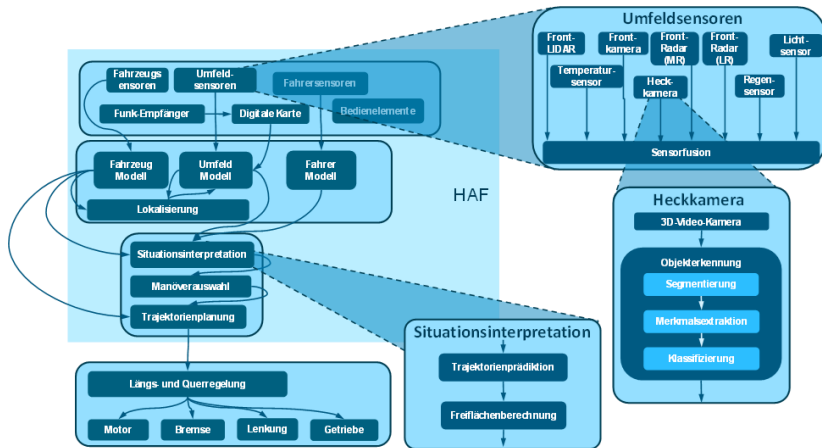


Abbildung 2: Verfeinerung der Architektur

Diese verfeinerte Architektur dient als Einstieg in den Schritt (2) Gefährdungsanalyse. Dieser Schritt hat das Ziel, die potentiell durch die Funktion entstehenden Gefährdungen, deren Ursachen im System sowie auch mögliche Auslöser aus der Umgebung des Fahrzeugs zu identifizieren. Hier liegt der Fokus insbesondere auf Gefährdungen, die nicht durch zufällige Hardwarefehler (*random hardware faults*) sondern durch Leistungsgrenzen (*performance limitations*) oder funktionale Unzulänglichkeiten (*functional insufficiencies*) in der Wahrnehmung, Modellierung und Interpretation der Umgebung sowie in der Planung von Manövern und Trajektorien entstehen. Die Gefährdungsanalyse geschieht in zwei Schritten und ist aus der klassischen HAZOP-Methode (siehe beispielweise (Ericson 2005)) abgeleitet.

Zunächst werden auf Fahrzeugebene die möglichen Gefährdungen, die aus einem falschen Verhalten der HAF-Funktion entstehen können, mit einem Keyword-basierten Ansatz identifiziert. Dafür starten wir mit so genannten Basisszenarien (in unserem Beispiel ein langsamer Vorausfahrer) und atomaren Manövern (hier: Spurwechsel nach links) mit denen die HAF-Funktion auf das Basisszenario reagieren könnte. Auf jedes dieser Manöver in dem vorgegebenen Basisszenario werden nun Keywords angewendet (hier: unangemessen), die das daraus beobachtbare Verhalten in dem Szenario ergeben (hier: Wechsel auf linken Fahrstreifen). Nun werden zusätzliche Umgebungsbedingungen ergänzt, die dazu führen

würden, dass dies zu einem kritischen Szenario wird (hier: schnell überholendes Fahrzeug auf Fahrstreifen links vom Ego). Dieses Vorgehen ist unabhängig von einer konkreten HAF-Funktion und das Ergebnis daher wiederverwendbar.

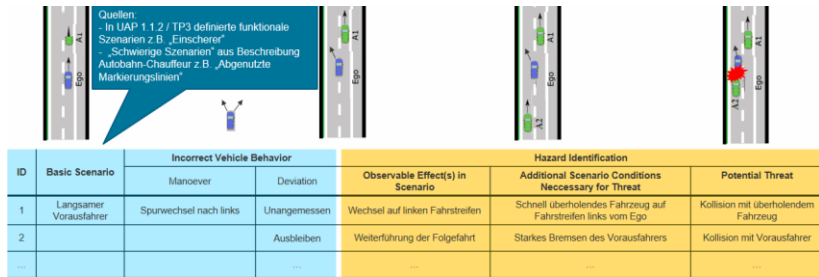


Abbildung 3: Identifikation von Gefährdungen 1

Anschließend wird auf dieses Verhalten in einem weiteren Keyword-basierten Ansatz referenziert. In diesem zweiten Schritt unserer Gefährdungsanalyse werden die Keywords nun auf die einzelnen Teilfunktionen (*Components*) der HAF-Funktion angewendet und zunächst untersucht welche Effekte sich lokal, für das System sowie für das Gesamtfahrzeug ergeben. Die daraus entstehende Tabelle ist in Abbildung 3 dargestellt. Falls Keywords nicht eindeutig anwendbar sind, so ist dies ein Indiz dafür, dass es nötig ist die Funktionsarchitektur weiter zu verfeinern. Natürlich wird es aber Keywords geben, die für diese bestimmte Komponente nicht sinnvoll anwendbar sind. Dies sollte an der entsprechenden Stelle vermerkt werden.

Auf der linken Seite im hellgelben Bereich werden systematisch alle Komponenten der Funktionsarchitektur sowie deren Teilfunktionen betrachtet. Die Teilfunktionen (in unserem Beispiel die Objekterkennung der Heckkamera) werden als Tripel aus Input (hier: Kamerabild), Computation (hier: Segmentierung) und Output (hier: Segmentiertes Kamerabild) dargestellt. Auf jedes Element dieses Tripels werden die Keywords aus einer Keywordliste systematisch angewendet (hier: *too large*). Dabei wird jeweils geprüft, ob das daraus entstehende Verhalten inkorrekt und somit ein lokaler Fehler (*Local Failure*) ist. Dies kann in unserem Beispiel bedeuten, dass ein Objekt im segmentierten Kamerabild zu groß ist. In diesem Fall werden im hellblauen Bereich der Tabelle die potentiell schlimmsten Folgen dieses lokalen Fehlers ermittelt (*Bottom-Up*, induktiv). Dies

wird für jedes Basisszenario (hier: langsamer Vorfahrer) separat durchgeführt. Darauf basierend wird zunächst ein möglicher negativer Systemeffekt innerhalb der HAF-Funktion (hier: Objektklasse wird falsch erkannt, dies führt zu fehlerhafter Trajektorienprädiktion des Objektes und dies wiederum führt dazu, dass die geplanten Trajektorien des Ego Fahrzeuges unsicher sind) bestimmt und das daraus entstehende inkorrekte Verhalten auf Fahrzeugebene (hier: unangemessener Spurwechsel nach links) abgeleitet. Die weiteren Auswirkungen dieses Verhaltens wurden entweder bereits im ersten Teilschritt in der Tabelle vorher (Schritt 2a) identifiziert (wie es im Beispiel der Fall ist) oder es wird, falls dies nicht der Fall ist, eine neue Zeile in die dortige Tabelle hinzugefügt mit dem neu identifizierten Fahrzeugverhalten.

Zusätzlich werden bereits einzelne lokale Wirkketten identifiziert. Dazu werden mögliche Ursachen innerhalb des Systems (hier: verzerrtes Kamerabild) für die betrachteten Abweichungen genannt (sofern vorhanden) sowie jeweils auslösende Umwelteigenschaften (in unserem Beispiel könnte das verzerrte Kamerabild durch einzelne Wassertropfen verursacht werden). Als letzte Spalte in der Tabelle wird mit Hilfe einer Ordinalskala nach der geschätzten Relevanz der auslösenden Umwelteigenschaft für den Fall gefragt, dass ein Mensch das Fahrzeug führt. Falls auch ein Mensch mit der Umwelteigenschaft starke Schwierigkeiten hat, so kann argumentiert werden, dass diese kein neues Automationsrisiko darstellt und die dadurch induzierten Szenarien bereits in vorhandenen Messdaten zu finden sein sollten. In unserem Beispiel schätzen wir die Wahrscheinlichkeit, dass in Mensch mit einzelnen Wassertropfen Probleme hat als gering ein.

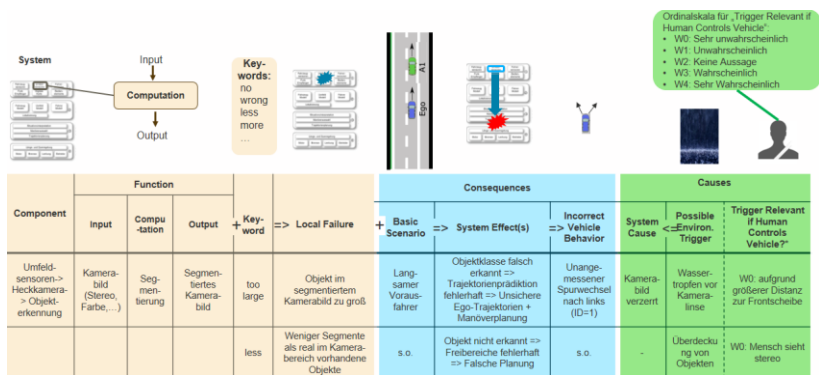


Abbildung 4: Identifikation von Gefährdungen 2

Daraufhin wird eine Wirkkette abgeleitet (siehe Abbildung 5:). Hier starten wir mit einem in der Gefährdungsanalyse identifizierten Top Level Event in einem Basisszenario (hier: Kollision mit überholendem Fahrzeug im Basisszenario langsamer Vorfahrer). Der Schritt der detaillierten Wirkkettenanalyse geschieht mittels einer erweiterten Fehlerbaumanalyse (für eine Einführung in Fehlerbäume, siehe (Ericson 2005)), die insbesondere das Ziel hat, Design- und Spezifikationsfehler (*Systematic Faults*) zu identifizieren, die verantwortlich für Gefährdungen sind. Diese Fehler sind inhärent im System enthalten, führen aber üblicherweise nur unter bestimmten Bedingungen tatsächlich zu einer Gefährdung. Dies können andere Fehler im System sein, aber insbesondere auch Bedingungen der Umgebung. Um diese identifizieren und modellieren zu können wurde die Fehlerbaumanalyse um ein weiteres logisches Gatter erweitert (eine Variation des Inhibit-Gates), welches es ermöglicht solche Umgebungsbedingungen, die für eine Fehlerpropagation notwendig sind, zu spezifizieren. Des Weiteren können Basisereignisse nicht nur systemseitige Fehler sein, sondern ebenfalls Umgebungsbedingungen, die den darüber liegenden Fehler auslösen. Hier wird, wie im gesamten Prozess, das jeweilige Basisszenario wieder als geltend angenommen. Ausgehend vom Top Level Event wird für jede mögliche Abweichung eines systemweiten Ausgangssignals vom korrekten Verhalten, die unmittelbar zum Auftreten des Top Level Events führen kann, ein neuer Knoten erstellt und logisch verknüpft. Die möglichen Ursachen für diese Auswirkungen werden unterschieden in

- (Zufällige) Hardwarefehler (E/E-Fehler abgedeckt durch ISO 26262): z.B. HW-Fehler in ECU für Trajektorienberechnung
- Designfehler in HW oder SW (Abweichung der Implementierung von der Spezifikation): z.B. Fehler in Trajektorienberechnungsalgorithmus
- Spezifikationsfehler (Fehler in der Spezifikation oder Unterspezifikation): z.B. Trajektorienberechnung ohne Dynamikmodell konzipiert
- Propagierte Fehler (der Input, den die Komponente bekommt ist bereits fehlerhaft): z.B. das gewählte Manöver und die Referenztrajektorie sind bereits fehlerhaft.

Um systematische Fehler zu identifizieren stellen sich insbesondere die Fragen, wo der Unterschied zum Menschen liegt und welche Einflüsse Position, Fehlerbilder und Funktionsweise haben. Zusätzlich werden alle Kombinationen von Umgebungsbedingungen unter denen

dieser Fehler zum nächsten Knoten propagiert durch ein modifiziertes *Inhibit-Gate* modelliert. Anschließend wird für jeden propagierten Fehler dieses Verfahren solange iterativ angewendet bis man auf der Sensorik-Ebene angelangt ist oder die betrachteten Komponenten keinen weiteren Input bekommen und es somit keine propagierten Fehler geben kann.

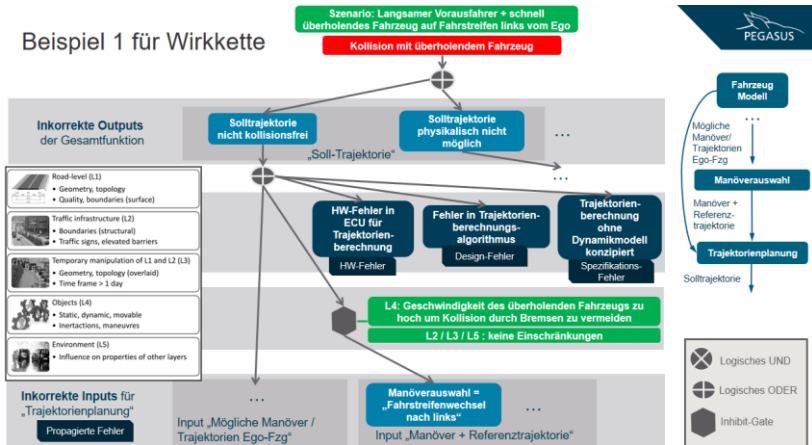


Abbildung 5: Beginn der Wirkkette

Nachdem diese Wirkkette vollständig durchlaufen wurde, können nun die risikoauslösenden Eigenschaften abgeleitet werden (siehe Abbildung 6:). Dazu wird zunächst die obige Wirkkette auf jene Knoten reduziert, die Umgebungsbedingungen darstellen unter Erhaltung der logischen Struktur. Basierend auf diesem Baum werden alle Umgebungsbedingungen schrittweise formalisiert und konkretisiert mit dem Ziel, dass sie nach diesem Schritt

- eindeutig formalisiert,
- präzise genug und damit prinzipiell quantifizierbar,
- in der Sprache der Umgebungsmodellierung beschreibbar und
- wo notwendig zeitlich relativ zueinander angeordnet sind.

Falls die Umgebungsbeschreibung nicht konkret genug ist um ihre Auftretenswahrscheinlichkeit prinzipiell zu schätzen, so ist dies an dieser Stelle ein weiteres Indiz dafür, dass die Funktionsmodellierung oder die Umgebungsmodellierung nicht ausreichend konkret sind.

Daher wird auch hier iterativ, wie in Abbildung 1 dargestellt, an der notwendigen Stelle konkretisiert und der Prozess entsprechend der Konkretisierung für die betreffenden Stellen erneut durchgeführt. Die von uns verwendete Umgebungsbeschreibung verwendet das 5-Ebenen Modell basierend auf (Schuldt 2017).

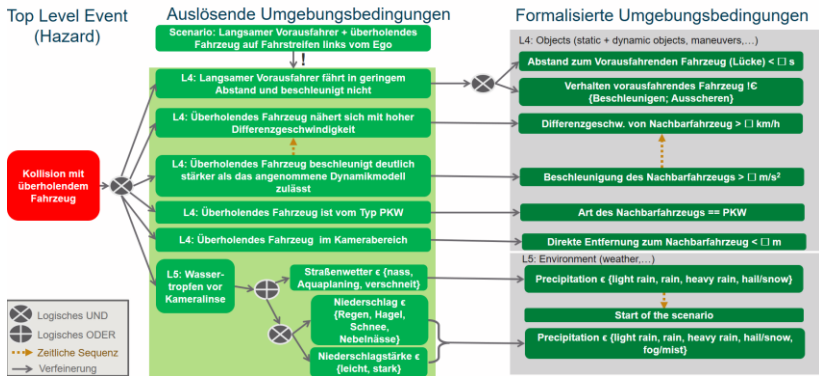


Abbildung 6: Ableitung auslösender Eigenschaften

Diese Umgebungsbedingungen können nun in zeitliche Sequenzen eingeteilt werden, sodass daraus ein Szenario entsteht, in welchem nur die relevanten Umweltbedingungen spezifiziert wurden. Dieses Szenario kann beispielsweise mit Hilfe einer Traffic Sequence Chart (Damm et al., 2018) beschrieben werden.

Literaturverzeichnis

- Akhtar, Naveed, und Ajmal Mian. „Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey.“ In *IEEE Access*, 14410-14430. 2018.
- Bagschik, Gerrit, Till Menzel, Andreas Reschka, und Markus Maurer. „Szenarien für Entwicklung, Absicherung und Test vom automatisierten Fahrzeugen.“ Walting, 2017. 125-135.
- Damm, Werner, Eike Möhlmann, Thomas Peikenkamp, und Astrid Rakow. „A Formal Semantics for Traffic Sequence Charts.“ In *Proc. Principles of Modeling - Essays Dedicated to Edward A. Lee on the Occasion of His 60th Birthday*, 182-205. Cham: Springer, 2018.
- Ericson, Clifton A. *Hazard Analysis Techniques for System Safety*. John Wiley & Sons, Inc., 2005.
- International Organization for Standardization. *ISO/DIS 26262-1 - Road vehicles — Functional safety*. International Organization for Standardization / Technical Committee 22 (ISO/TC 22), 2009.
- International Organization for Standardization. *ISO/PAS 21448 - Road vehicles -- Safety of the intended functionality*. International Organisation for Standardization / Technical Committee 22/ Sub committee 32, 2019.
- PEGASUS. „PEGASUS Projekt.“ November 2017.
https://www.pegasusprojekt.de/files/tmpl/PDF-HZE/06_Kritische-Szenarien-fuer-und-durch-die-HAF.pdf.
- Pütz, Andreas, Adrian Zlocki, und Lutz Eckstein. „Absicherung hochautomatisierter Fahrfunktionen mithilfe einer Datenbank relevanter Szenarien.“ *11. Workshop Fahrassistenzsysteme und automatisiertes Fahren*. Walting, 2017. 161-168.
- Schuldt, Fabian. „Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen.“ *Ph.D. Dissertation*. Braunschweig: Technische Universität Braunschweig, 2017.
- Society of Automotive Engineers (SAE). *SAE ARP4754A Guidelines for Development of Civil Aircraft and Systems*. Warrendale, USA: Society of Automotive Engineers (SAE), 2010.