VANASSIST - INTEGRIERTES SICHERHEITSKONZEPT FÜR AUTOMATISIERTE KLEINTRANSPORTER IN DER PAKETLOGISTIK

T. Hegerhorst[1], A. Vorwald[2], M. Flormann[1], M. Zhang[2], R. Henze[1] und A. Rausch[2]

[1] Institut für Fahrzeugtechnik, TU Braunschweig, Hans-Sommer-Str. 4, 38106 Braunschweig, t.hegerhorst@tu-braunschweig.de
[2] Institute for Software and Systems Engineering, TU Clausthal, Arnold-Sommerfeld-Straße 1, 38678 Clausthal-Zellerfeld, andreas.vorwald@tu-clausthal.de

Keywords: Automated Vehicle, Connected Vehicle, Highly Automated Driving Functions, Safety, Dependability, Laufzeitüberwachung, Human Intervention

ABSTRACT

Innerhalb des Projekts „VanAssist“ ist mit dem Ziel der Automatisierung der Paketzustellung ein autonomes Fahrzeug zu entwickeln. Die Erbringung eines Korrektheitsbeweises zur Umsetzung des automatisierten Fahrsystems ist dabei eine große Herausforderung und wird innerhalb dieses Beitrages durch die Anwendung einer Laufzeitüberwachung gelöst. Hierfür wird ein Sicherheitskonzept entwickelt, welches aus den drei Schichten „rekonfigurierbares Fahrsystem“, „Dependability Cage (Onboard-Laufzeitüberwachung)“ und „Leitstand (Offboard-Laufzeitüberwachung)“ besteht und es ermöglicht, kritische und für das Fahrsystem nicht lösbare Aufgaben zu detektieren und durch eine Rekonfiguration des Fahrsystems oder durch Anfordern externer Hilfe zu lösen. Die drei Teilsysteme und auch der aufgebaute Fahrzeugversuchsträger werden dabei sowohl konzeptionell als auch in der Realimplementierung vorgestellt und evaluiert. Dabei wird deutlich, dass durch das Zusammenspiel der verschiedenen Systeme mit der Möglichkeit zur Rekonfigurierbarkeit des Fahrsystems (Wechsel in Fail-Operational Mode) durch das Sicherheitssystem kritische Situationen wie Engstellen erfolgreich erkannt und gelöst werden, wodurch der Wirkradius des automatisierten Fahrzeuges zur Paketzustellung deutlich vergrößert werden kann. Durch den Ansatz einer fahrzeugungebundenen Instanz in Form eines Leitstandes ist zudem die Anwendung auf größere Flottenverbunde möglich.

1 ZIELE DES PROJEKTS VANASSIST

Die Paketverteilung in urbanen Räumen wird derzeit vorwiegend durch Zusteller:innen in konventionellen Fahrzeugen der „Sprinterklasse“ durchgeführt. Dieser Prozess ist personal- und kostenintensiv, da beispielsweise bis zu einem Viertel der reinen Fahrzeit auf die An- und Abfahrt vom und zum Depot entfallen. Weiterhin führt der Einsatz der genannten Fahrzeuge im urbanen Raum zu einer Vielzahl an Anfahr- und Bremsvorgängen, wodurch die Umwelt mit hohen Schadstoffemissionen, Verschleiß und Lärmbelästigungen belastet wird.

Das vom BMVI geförderte Verbundprojekt “VanAssist” adressiert zur Lösung dieser Probleme die Entwicklung einer integrierten Fahrzeug- und Systemtechnologie, die eine weitgehend emissionsfreie und automatisierte Zustellung von Gütern in urbanen Zentren ermöglicht. Konkret war es Aufgabe ein automatisiertes und durch einen Leitstand fernüberwachtes Fahren sowohl auf Betriebsgeländen (Rangiervorgänge) als auch in urbanen Umgebungen durch den Einsatz eines automatisierten Fahrsystems in elektrifizierten Transportfahrzeugen als intelligente Assistenz des Zustellers zu realisieren und durch Einsatz dieser Technologie eine Steigerung der Effizienz der Zustellung und Entlastung des Zustellers zu erreichen. Indem das Fahrzeug sich bei der gemeinsamen Zustellung mehrerer Stopps automatisiert zum nächsten vom Zusteller vorgegebenen Haltepunkt bewegt und somit der mitunter mehrfache Rückweg zum Fahrzeug entfällt, können die zu Fuß zurückzulegenden Wegstrecke reduziert werden. Weiterhin kann der Zusteller das Fahrzeug aus der Distanz zu bestimmten Haltepunkten senden und so die unproduktiven Wegstrecken zurück zum Fahrzeug vermeiden. Hierdurch wird eine kontinuierliche Bewegung entlang der von ihm geplanten Route ermöglicht.

Dieser Beitrag adressiert dabei die Arbeiten des Niedersächsischen Forschungszentrum Fahrzeugtechnik vertreten durch das Institut für Fahrzeugtechnik (TU Braunschweig) und das Institute for Software and Systems Engineering (TU Clausthal). Dies umfasst zum einen die Entwicklung des automatisierten Versuchsträgers sowie des automatisierten Fahrsystems (IfF) als auch des Laufzeitüberwachungssystems (sogenannter „Dependability Cage“) sowie des Leitstands (ISSE). Besonders im Fokus soll dabei das Zusammenspiel der beiden Systeme im Rahmen des entwickelten Sicherheitskonzepts stehen, wobei das Ziel darin besteht ein zum Fahrsystem redundantes Sicherheitssystem zu entwickeln, welches in der Lage ist kritische Situationen oder Fehl und so einen sicheren Betrieb des automatisierten Fahrzeugs sicherzustellen. Folgend wird dazu das erarbeitete Gesamtkonzept und daran anschließend die Teilaspekte Fahrzeug, Fahrsystem, Dependability Cage und Leitstand vorgestellt.

2 GESAMTKONZEPT FÜR ZUVERLÄSSIGES AUTOMATISIERTES FAHREN

Automatisierte Fahrzeuge sind in aller Munde und bieten, wie bereits erwähnt, großes Potential die Logistikprozesse des Güterverkehrs zu optimieren. Das Ziel des automatisierten Fahrens ist es, den Menschen möglichst von der Fahraufgabe zu entlasten. Dabei wird die Höhe der Entlastung durch den Automatisierungsgrad des Fahrsystems bestimmt. Um den Automatisierungsgrad von Fahrzeugen zu bestimmen hat SAE J3016 [1] sechs Stufen definiert (SAE Stufe 0-5). SAE Stufe 0 definiert, dass keine Automatisierung auf dem Fahrzeug vorherrscht, wodurch die Fahraufgabe und die Verantwortung über die Sicherheit vollständig bei dem Menschen liegt. Dahingehend werden Fahrzeuge der SAE Stufe 5 als vollautomatisiert angesehen, wodurch der Mensch vollständig von der Fahraufgabe und der Sicherheitsverantwortung befreit werden kann. In den Stufen 1-4 teilen sich Mensch und Fahrzeug die Fahraufgabe, sowie die Verantwortung über die Sicherheit.

Der klassische Engineering Prozess von automatisierten Fahrsystemen basiert auf einer Closed-World-Assumption, bei der davon ausgegangen wird, dass alle Systemartefakte vollständig und valide sind. Unter dieser Vollständigkeitsannahme kann somit ein automatisiertes Fahrzeug entwickelt werden, bei dem, zumindest theoretisch, während der Entwicklungszeit eine Zusicherung über die Sicherheit zur Laufzeit gegeben werden kann. Natürlich ist die Erbringung eines Korrektheitsbeweises zur Entwicklungszeit keine triviale Aufgabe [7]. Dennoch existieren Methoden, welche die Sicherheit des Systems während der Laufzeit gewährleisten, indem sie sicherheitskritische Funktionen überwachen und bei einer Verletzung gegenüber Sicherheitsspezifikationen eine Reaktion auslösen, mit der gefährliche Situationen vermieden werden (Laufzeitüberwachung).

Bei den SAE Stufen 0-2 manifestiert sich die Reaktion durch eine sog. Fail-Safe Reaktion. Der permanent anwesende Sicherheitsfahrer muss hierbei die Aufgabe ausgefallener Funktionen übernehmen. In den SAE Stufen 3-5 ist dies allerdings nicht mehr ausreichend. Neben Sicherheitsanforderungen gilt es auch Lebendigkeitsanforderungen nachzuweisen, die sicherstellen, dass das Fahrzeug das Ziel in endlicher Zeit erreichen kann. Um dies sicherzustellen, kann ein sog. Fail-Operational Ansatz verfolgt werden, der z. B. die Funktionalität des Fahrzeugs einschränkt (z. B. durch Limitierung der Maximalgeschwindigkeit), sodass das Fahrzeug weiter im Betrieb bleibt.

Eine Closed-World-Assumption ist allerdings eine Traumvorstellung, welche, durch den derzeitigen bzw. zukünftigen Anforderungen an automatisierten Fahrsystem, nicht mehr anwendbar ist. Aniculeasei et al. [2] definieren Herausforderungen beim Engineering von modernen zuverlässigen autonomen Systemen[1]. Ungenaue und unbekannte Systemumgebungen erzeugen zwangsläufig unvollständige Systemartefakte. Zudem steigen die Anforderungen in Richtung Selbstadaptivität, um Probleme effizienter lösen bzw. um neue Fähigkeiten dynamisch zu erlernen, wodurch das vollständige Verhalten des Systems a priori nicht bekannt sein kann. Die Argumente von Aniculeasei et al. [2] zeigen, dass eine Closed-World-Assumption nicht mehr angemessen ist, um zuverlässige automatisierte bzw. autonome Fahrzeuge zu entwickeln. Vielmehr sollte der Engineering Prozess die Unvollständigkeit der Systemartefakte akzeptieren und berücksichtigen (sog. Open-World-Assumption), indem ein

[1] Nach SAE J3016 [1] dürfen autonome Fahrzeuge keine Kommunikation und/ oder Kooperation mit außenstehenden Entities besitzen. In diesem Fall sind automatisierte Fahrzeuge die Basis für autonome Fahrzeuge. Konzeptionell beinhaltet die Absicherung autonomer Fahrzeuge auch die Absicherung von automatisierten Fahrzeugen, wodurch diese Herausforderungen auch für automatisierte Fahrzeuge bzw. Fahrsysteme gelten.

automatisiertes bzw. autonomes Fahrzeug, durch ein abwechselndes Zusammenspiel aus Entwicklungs- und Laufzeit, entwickelt wird, sodass Systemartefakte iterativ vervollständigt werden.

Im Rahmen des Projektes VanAssist wird aus Absicherungsperspektive nur das Fahrsystem betrachtet. Die o. g. Herausforderungen stellen Anforderungen an eine übergreifende Architektur für zuverlässiges (inklusive sicheres) automatisiertes Fahren. Prinzipiell lassen sich zwei Bereiche identifizieren: (a) ein Fahrsystem, welches neben dem Nominal Mode auch Fail-Operational Modes realisieren kann und (b) eine Laufzeitüberwachung, welche die Notwendigkeit von Fail-Operational Modes erkennt und diese dann auslöst. Basierend auf diesen Anforderungen wurde im Rahmen von VanAssist eine übergreifende 3-Schichten Architektur entwickelt (vgl. Abbildung 1). Die unterteste Schicht definiert ein rekonfigurierbares modulares automatisiertes Fahrsystem, welches neben dem Nominal Mode Fail-Operational Modes, z. B. durch die Rekonfiguration auf Komponentenebene bzw. Systemebene realisiert. Als Laufzeitüberwachung kommt ein redundantes kooperatives System aus einer Onboard-Laufzeitüberwachung (mittlere Schicht) und einer Offboard-Laufzeitüberwachung (obere Schicht) zum Einsatz.

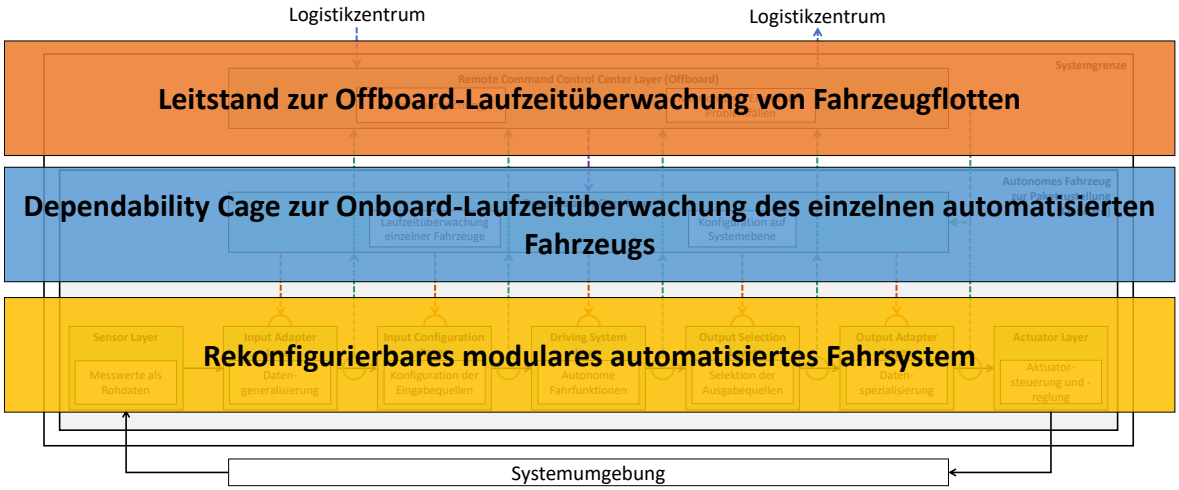

Abbildung 1: Gesamtarchitektur für sicheres und zuverlässiges automatisiertes Fahren

Die Onboard-Laufzeitüberwachung ist eine Instanz eines sog. Dependability Cages [2], der auf Systemebene die Sicherheitsspezifikationen des Fahrsystems automatisch überwacht und im Falle einer Verletzung einen Fail-Operational Mode auslöst. Um eine Entscheidung über die Maßnahmen zur Realisierung des Fail-Operational Modes zu treffen wird eine entsprechende Instanz benötigt. Momentan ist diese Instanz der Remote Operator, der basierend auf menschlicher Wahrnehmung und Inferenz die Entscheidung trifft. Dieser kann durch einen Leitstand das Fahrzeug aus der Ferne mit Hilfe des Sensorstreams überwachen und bei Problemfällen mittels Fernzugriff den menschlichen Eingriff vornehmen. Es kann sozusagen von einem Offboard-Sicherheitsfahrer gesprochen werden, der zudem, anders als ein Onboard-Sicherheitsfahrer, in der Lage ist, eine ganze Fahrzeugflotte zu überwachen. In diesem Sinne kann der Leitstand zusammen mit dem Remote Operator als eine Art menschlicher Dependability Cage interpretiert werden. Um den Prozess zukünftig weiter zu optimieren, kann durch Steigerung des Automatisierungsgrades (SAE Stufe) auch der remote Leitstand entlastet werden, indem der Dependability Cage, durch Ausnutzung des Potentials für Fail-Operational Modes, mehr Sicherheitsverantwortung vom remote Leitstand abnimmt.

3 MODULARES AUTOMATISIERTES FAHRSYSTEM

Aus den obigen Ausführungen wird deutlich, dass es eines automatisierten Fahrsystems bedarf, welches die unterste Schicht des in Abbildung 1 dargestellten Gesamtkonzepts realisiert. Hierzu soll folgend sowohl der aufgebaute Fahrzeugversuchsträger zusammen mit der Hardware-Architektur des Automatisierungssystems (ADS) als auch die entwickelten Fahrfunktionen vorgestellt werden.

3.1 Fahrzeugversuchsträger PLUTO

Innerhalb des Projekts wurde der Versuchsträger PLUTO (Plattform for future urban mobility and transport) aufgebaut, welcher zur Demonstration der entwickelten Funktionalitäten dient. Bei PLUTO handelt es sich um ein auf einem HFM Motionboard (vgl. Abbildung 2) basierendem rolling chassis. Diese adaptive und elektrifizierte Fahrzeugplattform mit einer Größe von ca. 5 x 2m nimmt alle notwendigen Elemente des Fahrwerks und Antriebs in sich auf. Dies umfasst antriebsseitig zwei leistungsstarke Radnabenelektromotoren der Bauart Elaphe L1500 mit einer Leistung von je 66 kW an den Rädern der Hinterachse, welche zusätzlich eine Rekuperationsbremsung erlauben. Fahrwerksseitig verfügt das Fahrzeug über eine Luftfederung sowie eine zusätzliche hydraulische Bremsanlage. Ein Drive-by-wire-System „SpaceX“ der Firma Paravan dient zur vollelektrifizierten Ansteuerung der Aktorik und stellt dabei einen Joystick als Bedienelement zur Verfügung. Das verbaute drive-by-wire-System ist herstellerseitig zusätzlich über ein CAN-Interface ausgestattet, welches es erlaubt ein ADS-System zur Fahrzeugführung zu integrieren. Als Energiespeicher dient eine Hochvoltbatterie mit einer Kapazität von 20 kWh.

Die Fahrplattform kann mit diversen Aufbauten zur Darstellung verschiedener Anwendungsfälle ausgestattet werden, wobei es ermöglicht wird den vorliegenden Use-Case des Lieferfahrzeuges umzusetzen.

Abbildung 2: HFM Motionboard als Grundlage für den Versuchsträger PLUTO

Der Versuchsträger wurde dann zum automatisierten Fahren entsprechend einer am IfF entwickelten Hardwarearchitektur (vgl. Abbildung 3) ertüchtigt. Im Kern besteht diese aus einer Rapid Control Prototyping (RCP) Plattform, wobei eine dSpace SCALEXIO AutoBox als zentrale CPU in Form eines Echtzeitsystems zum Einsatz kommt. Diese besitzt einen Intel i7-6820EQ, quad-core Prozessor mit 2.8 GHz Taktfrequenz und 4GB Arbeitsspeicher. Ergänzt wird dieses System durch weitere Kontroll- und Messrechner. Darüber hinaus ist das Fahrzeug mit umfangreicher umfelderfassender Sensorik ausgestattet worden. Maßgeblich wird dabei ein Sensorsetup aus 8 Ibeo NEXT Solid State Laserscannern verwendet, welche einen horizontalen Öffnungswinkel von je 60° aufweisen. Das verwendete und in Abbildung 3 dargestellte Setup führt somit zu einer 360°-Abdeckung bei der Sensierung der Umgebung. Weiterhin sind rund um das Fahrzeug vier Fish-Eye-Kameras mit einem Öffnungswinkel von mehr als 180° verbaut, wodurch auch eine kamerabasierte redundante Erfassung der Umgebung möglich ist. Eine GeneSys ADMA Slim dient weiterhin als Referenz-Lokalisierungssystem und komplettiert das verwendete Sensorsetup. Über ein CAN-basiertes Interface wird die Kommunikation zwischen automatisiertem Fahrsystem (ADS) und dem Fahrzeug bzw. der Vehicle Control Unit (VCU) umgesetzt.

Ergänzend ist ein weiterer Rechner verbaut, auf welchem das Onboard-Sicherheitssystem (Dependability Cage) implementiert ist. Zusätzlich ist dieses System kabellos mit dem Leitstand vernetzt. Die Kommunikation der verschiedenen Systeme im Fahrzeug wird über das Framework *ROS2* umgesetzt, vgl. auch Kapitel 4.1.

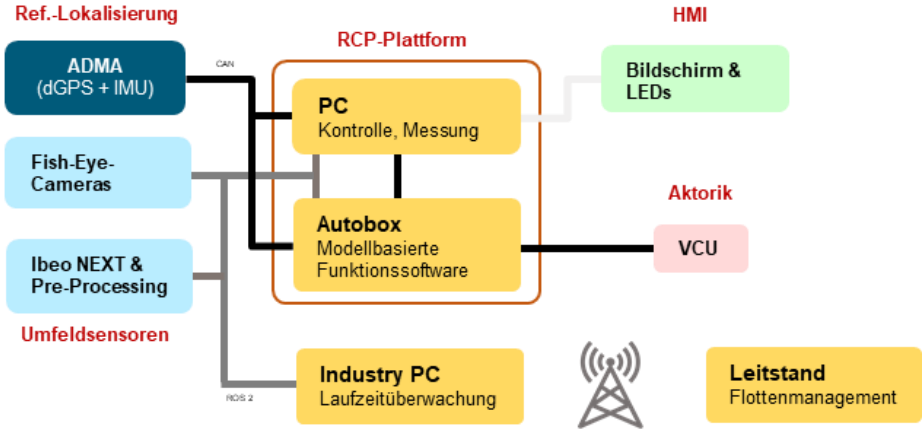

Abbildung 3: Hardware-Architektur des Versuchsträgers PLUTO zusammen mit dem Leitstand

Nachdem das Fahrzeug so hardwareseitig zum automatisierten Fahren ertüchtigt werden konnte, wurde das Fahrzeug abschließend mit einem Aufbau entsprechend des Use-Cases ausgestattet. Hierbei wird als weiteres Element des ADS ein externales HMI im Fahrzeug entwickelt und verbaut, welches zur Kommunikation mit der Umwelt dient. Dieses umfasst ein in der Front verbautes Display sowie frei programmierbare LED-Streifen an der Front sowie Heck des Fahrzeugs. Aufgabe des Systems ist es die Umwelt über den Fahrzeugstatus zu informieren, was sowohl der aktuelle Missionsstatus – wie bspw. "Parken" oder "automatisierte Fahrt" aber auch die Information über ein vom Sicherheitssystem detektierten technischem Problem oder der Wechsel zwischen dem Nominal Mode und dem Fail-Operational Mode sein kann. Abbildung 4 zeigt den Versuchsträger mit den Laserscannern wie es im Projekt VanAssist eingesetzt wurde.

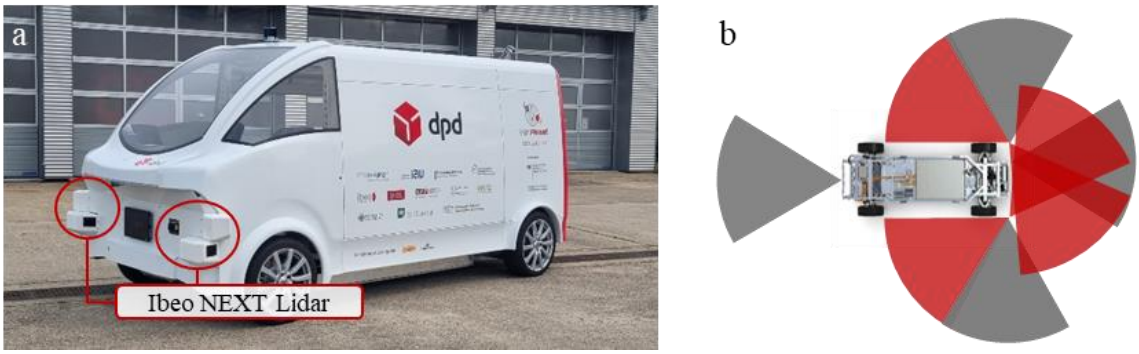

Abbildung 4: Versuchsträgers PLUTO (a) für den Anwendungsfall der Paketlogistik und (b) verwendetes Laserscannersetup.

3.2 Automatisiertes Fahrsystem in VanAssist

Auf der RCP-Plattform ist das automatisierte Fahrsystem implementiert, wobei sich auf eine am IfF entwickelten und in Abbildung 4 dargestellten Funktionsarchitektur gestützt wird. In einem ersten Schritt, dem sogenannten *Pre-Processing*, werden sämtliche Informationen aus den verschiedenen vorgestellten Sensoren, vom Fahrzeug sowie externer Informationsquellen wie digitalen Karten oder anderer externer Systeme in einem *Umfeldmodell* verarbeitetet, fusioniert und abstrahiert um eine ganzheitliche Repräsentation der Szene zu erreichen. Weiterhin werden diese Informationen auch zur Ego-Lokalisierung genutzt und abschließend wird eine Analyse der aktuellen Fahrsituation durchgeführt. Diese Informationen dienen als Grundlage für die nachgelagerten Algorithmen der *Handlungs- und Bewegungsplanung*. Maßgeblich wird hierin ein initialer Pfad auf Basis der bekannten Route (Start- und Zielpose) sowie der vorliegenden digitalen Karte geplant. Um einen fahrbaren Pfad – auch für das vorgestellte und durch einen Radstand von 3,25m relativ unwendigen Fahrzeug im urbanen Raum – zu erhalten, wird ein Ansatz unter Verwendung von B-Splines genutzt, welcher einen krümmungsoptimierten und letztendlich kinematisch fahrbaren Pfad berechnet. Die Pfadinformationen

werden nachgelagert mit einem Geschwindigkeitsprofil überlagert, wodurch eine Trajektorie entsteht. Über die *Handlungsplanung* können Neu- oder Umplanungen sowie Neuparametrierungen im Hinblick auf die Trajektorie vorgenommen werden. Zusätzlich wird hierüber auch das HMI maßgeblich gesteuert. Anschließend erfolgt im Bereich der *Fahrzeugregelung* die Transformation der Informationen der Soll-Trajektorie unter Abgleich des Bewegungszustands des Fahrzeuges in Stellgrößen für die Fahrzeugaktorik. Dies umfasst die Berechnung eines Lenkwinkels sowie einer Beschleunigung oder Verzögerung und auch die Wahl der Fahrstufe. Abschließend werden diese Größen plausibilisiert und an die Fahrzeugaktorik übergeben. Hier erfolgt dann die Umsetzung der Stellbefehle.

Um dem Dependability Cage bzw. dem Leitstand Möglichkeiten zur Beeinflussung des Fahrzeugverhaltens zu geben, wird das Fahrsystem entsprechend der in Kapitel 2 dargestellten Anforderungen rekonfigurierbar designt. Dies bedeutet, dass diverse Parameter aber auch konkrete Stellbefehle über eine entsprechende Schnittstelle rekonfiguriert werden können. Softwareseitig werden beide Systeme dazu als weitere Eingangsgröße betrachtet, wodurch auf alle nachgelagerten Module Einfluss genommen werden kann, vgl. Abbildung 5. So können sowohl übergeordnete Parameter wie bspw. die maximal zulässige Geschwindigkeit reduziert oder aber auch konkrete Handlungen wie das Ausführen eines Notstopps angefordert werden.

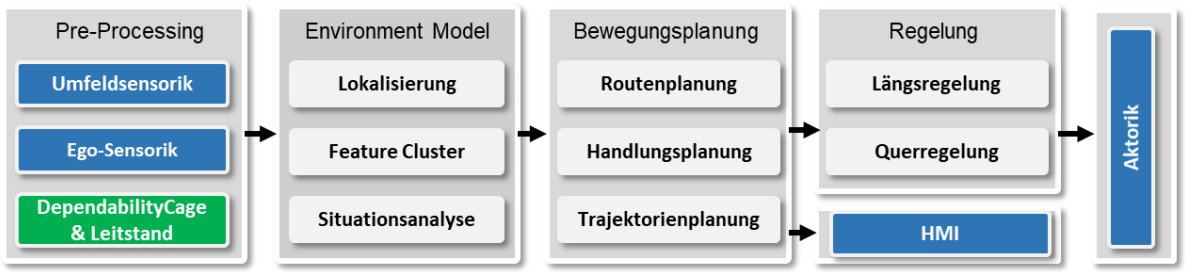

Abbildung 5: TOP-Layer der am IfF entwickelten und im Versuchsträger PLUTO implementierten Funktionsarchitektur des automatisierten Fahrsystems, wobei die blau markierten Komponenten Schnittstellen zu anderen Systemen darstellen und das Sicherheitssystem grün markiert ist

4 KOOPERATIVES SICHERHEITSKONZEPT MIT REDUNDANTER LAUFZEITÜBERWACHUNG

Wie bereits erwähnt ist die Absicherung automatisierter Fahrsysteme keine triviale Aufgabe. Neben Absicherungsmaßnahmen zur Entwicklungszeit, müssen auch Absicherungsmaßnahmen zur Laufzeit vorgenommen werden. Heutzutage ist auf öffentlichen Straßen stets ein Sicherheitsfahrer während des automatisierten Fahrens notwendig, der im Falle eines Problems eingreifen muss. Insbesondere in den SAE Stufen 0 – 2 muss der Mensch immer bereit sein einzugreifen. Höhere SAE Stufen (3 - 5) sind jedoch notwendig, wenn automatisierte Fahrzeuge Logistikprozesse optimieren sollen, da nur die Entfernung des Sicherheitsfahrers den Nutzen maximiert. Prinzipiell ist erkennbar, dass mit steigender Automatisierung der Mensch immer mehr von der Sicherheitsverantwortung entfernt werden kann, bis hin zu völliger Befreiung von dieser. Allerdings ist der Stand der Technik noch weit davon auf einen menschlichen Sicherheitsfahrer zu verzichten.

Ein Ansatz den Nutzen von automatisierten Fahrsystem im Rahmen des Projektes zu gewährleisten ist der Einsatz von Offboard-Sicherheitsfahrern, die durch ein Remote Leitstand mittels Fernzugriff automatisierte Fahrsysteme aus problematischen Situationen manövrieren können. Die Arbeit von Aniculeasei et al. [3] zeigt konzeptionell das Potential dieses Ansatzes, indem Onboard-Laufzeitüberwachung in Kombination mit einem Remote Leitstand vorgestellt wurde. Das Konzept beinhaltet vier Interventionsniveaus (*IN*) des Menschen („Classification“, „Decision“, „Planning“ und „Action“), indem dem Menschen in Abhängigkeit zum *IN* eine Funktion zugeordnet wird (sog. „Graceful Degradation“). So entspricht z. B. das *IN* „Classification“, dass der Mensch dem automatisierten Fahrsystem nur bei der Klassifikation von Objekten und Situationen hilft und *IN* „Action“, das lokale Hilfe benötigt wird, um das Problem zu lösen, da das Problem nicht aus der Ferne gelöst werden kann.

Inspiriert durch diesen Ansatz, wird im Projekt VanAssist für das allgemeine Sicherheitskonzept ein

kooperatives Sicherheitssystem vorgesehen. Dabei wird konkret zwischen On- und Offboard Laufzeitüberwachung unterschieden. Die Onboard-Laufzeitüberwachung überprüft Sicherheitsspezifikationen maschinell und löst im Falle eines Problems grundlegende Fail-Operational Maßnahmen, wie z. B. einen Notstopp, aus. Durch menschliche Wahrnehmung und Inferenz können nun komplexere Fail-Operational Maßnahmen durch die Offboard-Laufzeitüberwachung durchgeführt werden, um so das Fahrsystem aus dem Problemfall zu befreien.

Um das Konzept zu demonstrieren wurden zwei Szenarios im Projekt erstellt:

a) Das automatisierte Fahrsystem nähert sich einer Fahrbahnverengung. Der Remote Leitstand Operator wird informiert, erkennt die Situation und löst das Problem aus der Ferne durch die Rekonfiguration des automatisierten Fahrsystems. Nachdem die problematische Situation bewältigt wurde, erhält das automatisierte Fahrsystem wieder volle Verantwortung (für IN „Decision“ [3]).
b) Eine Kamera des automatisierten Fahrsystems wird durch herunterfallendes Laub bedeckt. Der Remote Leitstand Operator wird informiert, erkennt die Situation und fordert lokale Unterstützung aus der Ferne an. Die lokale Unterstützung löst das Problem und informiert den Remote Leitstand Operator, sodass dieser das automatisierte Fahren wieder freigeben kann (für IN „Action“ [3]).

4.1 Dependability Cages zur Onboard-Laufzeitüberwachung in VanAssist

Um die o. g. Szenarios zu berücksichtigen, wurden zunächst Sicherheitsanforderungen abgeleitet. Eine Fahrbahnverengung lässt sich i. d. R. auf Verkehrsteilnehmer oder sonstige Objekte auf der Straße (Systemumgebung) zurückführen, die weder angefahren noch überfahren werden dürfen. Zum Messen der Systemumgebung nutzen automatisierte Fahrsysteme Sensoren (Umgebungswahrnehmung). Somit ist die Validität der Sensordaten zwingend notwendig, damit sich ein automatisiertes Fahrsystem sicher bewegen kann. Falls diese nicht mehr gewährleistet ist, z. B. durch Verdeckung der Sensoren durch Laub, kann eine sichere Bewegung des automatisierten Fahrsystems nicht mehr garantiert werden. Dabei Validität der Sensordaten auf Datenebene ist dabei die Basis für die Validität auf semantischer Ebene. Im Projekt VanAssist wurden hierzu Kameradaten ausgewählt, da moderne automatisierte Fahrsysteme häufig Kameradaten verwenden und der Offboard-Sicherheitsfahrer Kameradaten benötigt, um genügend Informationen über die aktuelle Situation zu erhalten. Daraus ergibt sich, dass im Rahmen vom Projekt VanAssist folgende Sicherheitsspezifikationen überwacht werden müssen, wobei Szenario (a) durch Sicherheitsspezifikation (1) und Szenario (b) durch Sicherheitsspezifikation (2) validiert wird:

(1) Das Fahrsystem PLUTO darf nicht gegen bzw. über Hindernisse fahren
(2) Die Kameradaten des Fahrsystems PLUTO müssen auf Datenebene valide sein

Das Monitoring Framework Dependability Cage (vgl. Abbildung 6) adressiert zwei Bereiche der Laufzeitabsicherung: (a) Absicherung der Korrektheit des Systemverhaltens hinsichtlich Sicherheitsanforderungen (*Qualitative Monitor*) und (b) Absicherung des Systems hinsichtlich der zur Entwicklungszeit berücksichtigten und getesteten Situationen und Systemumgebungen (*Quantitative Monitor*).

Beide Monitore erfordern einen Zugriff auf konsistente und abstrakte Daten des betrachteten Systems mit Hilfe der Ein- und Ausgaben-Abstraktion. Sie stellen die Schnittstelle zwischen dem automatisierten Fahrsystem und den beiden Monitoren dar. Sowohl die Ein- als auch die Ausgabe-Abstraktionskomponenten verwenden definierte Schnittstellen, um auf die Daten des betrachteten Systems zuzugreifen und sie in abstrakte Repräsentationen zu transformieren. Abstrakte Darstellungstypen und -werte werden aus der Anforderungsspezifikation und den Zuverlässigkeitskriterien des betrachteten Systems abgeleitet.

Der *Qualitative Monitor* bewertet die Korrektheit und Sicherheit des Systemverhaltens unter der Annahme, dass das System in einer Situation und einer Systemumgebung arbeitet, die der Anforderungsspezifikation entsprechen. Er besteht aus einer abstrakten Verhaltensfunktion und einem Konformitätsoracle. Die abstrakte Verhaltensfunktion berechnet in der aktuellen abstrakten Situation in Echtzeit eine Reihe von korrekten und sicheren abstrakten Aktionen für das System. Das Konformitätsoracle vergleicht die abstrakte Ausgabe mit der Menge der korrekten und sicheren abstrakten Aktionen der abstrakten Verhaltensfunktion. Der *Quantitative Monitor* beobachtet die

angetroffenen abstrakten Situationen. Für jede Situation bewertet dieser Monitor in Echtzeit, ob die angetroffene abstrakte Situation bereits während der Entwicklungszeit bekannt war und getestet wurde. Eine Wissensbasis liefert Informationen über getestete Situationen auf abstrakter Ebene. Wenn einer der Monitore ein fehlerhaftes und unsicheres Systemverhalten feststellt, müssen Sicherheitsmaßnahmen eingeleitet werden, um die Zuverlässigkeitsanforderungen zu gewährleisten. Die Komponente *Fail-Operational Reaction* muss das gestörte System in einen sicheren Zustand mit akzeptablem Risiko überführen. Eine Fail-Operational-Reaktion kann z. B. eine "Graceful Degradation" des automatisierten bzw. autonomen Systems sein, wie in [3] beschrieben.

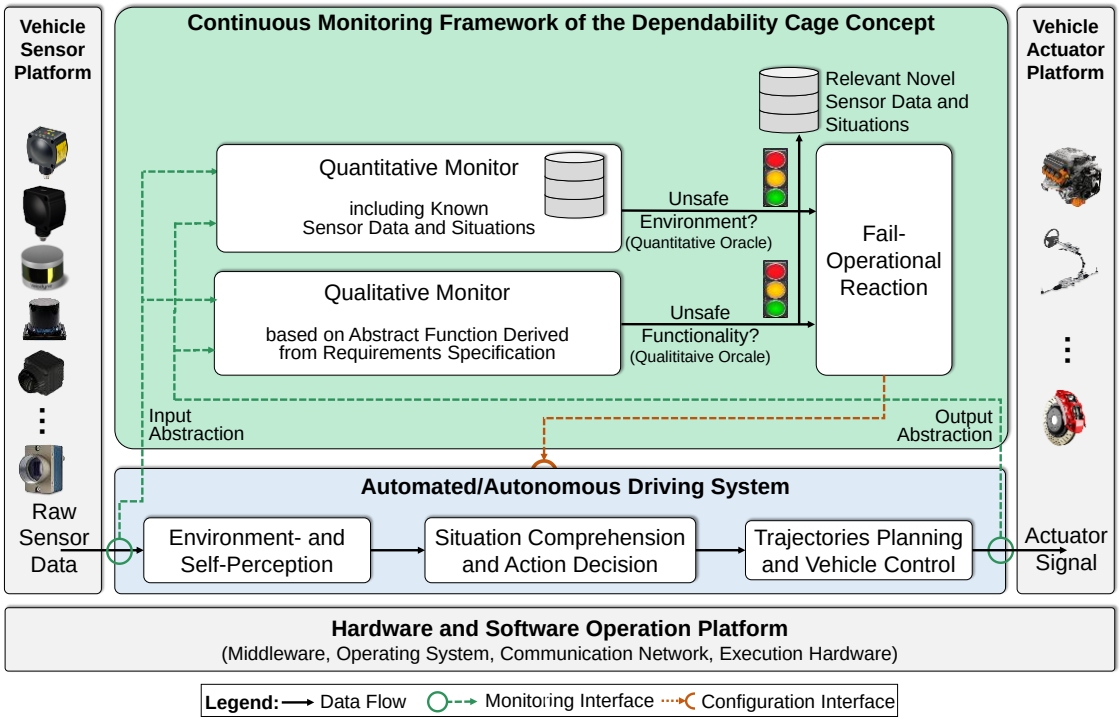

Abbildung 6: Übergreifende Architektur des Dependability Cages nach [2]

Basierend auf den Sicherheitsspezifikationen wurde die Sicherheitsarchitektur verfeinert. Für das Projekt VanAssist wird der *Quantitative Monitor* ausgeschlossen. In Abbildung 7 wird ein detaillierterer Einblick der Sicherheitsarchitektur bestehend aus dem rekonfigurierbaren modularen automatisierten Fahrsystem, der Onboard-Laufzeitüberwachung (Dependability Cage) und der Offboard-Laufzeitüberwachung (Remote Leitstand) dargestellt. Der Fokus liegt hierbei auf den Interna des Dependability Cages. Mittels Überwachungsschnittstellen sind die Laufzeitüberwachungssysteme in der Lage die aktuelle Situation zu beurteilen und in den Problemfällen Fail-Operational Modes auszulösen. Um dies zu erreichen wurden folgende Komponenten des *Qualitative Monitors* abgeleitet: *Safe Zone*, *Lidar Detector*, *Camera Validitor*. Als Komponente der *Fail-Operational Reaction* wurde die *Mode Control* abgeleitet.

Die Komponente *Safe Zone* ermittelt basierend auf der aktuellen Geschwindigkeit und dem Lenkwinkel des PLUTO Gefahrenbereiche (sog. „Safe Zone“), die im Nominalbetrieb komplett frei von Hindernissen sein muss. Die „Safe Zone“ ist eine Erweiterung der Monitoring Area von Grieser et al. [4] und berücksichtigt ein vollständiges Kreissegment bei Kurvenfahrten in beide Fahrtrichtungen und rechteckige Ausbreitungen ebenfalls in beide Fahrtrichtungen. Die „Safe Zone“ unterteilt sich in „Clear Zone“ und „Focus Zone“, wobei folgendes gilt: Wenn die „Focus Zone“ frei von Hindernissen ist, dann ist auch die „Clear Zone“ frei von Hindernissen. Um zu ermitteln, ob sich Hindernisse in der „Safe Zone“ befinden, wurde die Komponente *Lidar Detector* ermittelt. Basierend auf der Punktwolke aus den achten Ibeo NEXT Solid State Laserscannern (vgl. Abbildung 4) ermittelt diese Komponente, ob sich Hindernisse in der „Clear Zone“ bzw. „Focus Zone“ befinden. Hierzu wurde das Verfahren von Grieser et al. [4] zur Bestimmung, ob sich Lidar Punkte in der Monitoring Area befinden, durch ein

Clustering Verfahren erweitert, sodass Ghosting Points bzw. unkritische Lidar Punkte nicht berücksichtigt werden. Die Komponente *Camera Validator* ermittelt die Validität der Rohdaten der Kamera durch Überprüfung, indem Überprüft wird, ob die Kamera verdeckt ist. Hierbei wird ein Algorithmus zur Ermittlung der Schärfe von den Kamerabildern eingesetzt, dass beim Unterschreiten eines empirisch ermittelten Wertes, eine Bedeckung der Kamera erkennt. Die Komponente *Mode Control* entscheidet den aktuell gültigen Fahrmodus, basierend auf den Ergebnissen des *Lidar Detectors*, *Kamera Detectors*, sowie der aktuellen Geschwindigkeit des Fahrzeugs und der Anfrage zur Änderung des Fahrmodus vom Remote Leitstand. Der Fahrmodus bestimmt entspricht dabei der Konfiguration des rekonfigurierbaren modularen automatisierten Fahrsystems. Sensordaten, sowie Teilergebnisse der Komponenten des Dependability Cages können vom Remote Leitstand für die menschliche Wahrnehmung und Inferenz benutzt werden. Danach kann der menschliche Eingriff durch die Anfrage über den Fahr- bzw. Cagemodus realisiert werden. Die Kommunikation läuft hierbei stets über die *Mode Control*, sodass ungültige Modi im aktuellen Zustand ausgeschlossen werden können.

Komponenten in der Architektur werden durch die dezentrale Middleware *ROS2*, die auf dem Publish-Subscribe Kommunikationsparadigma basiert, integriert [5]. Die Entscheidung für diese Technologie wurde getrieben durch eine ausführliche Literaturrecherche von Warnecke et. al [6] und der Konzipierung von Fahrsystemen als verteiltes System. Zudem besitzt *ROS2* einen besonders hohen Fokus auf echtzeitkritische Systeme.

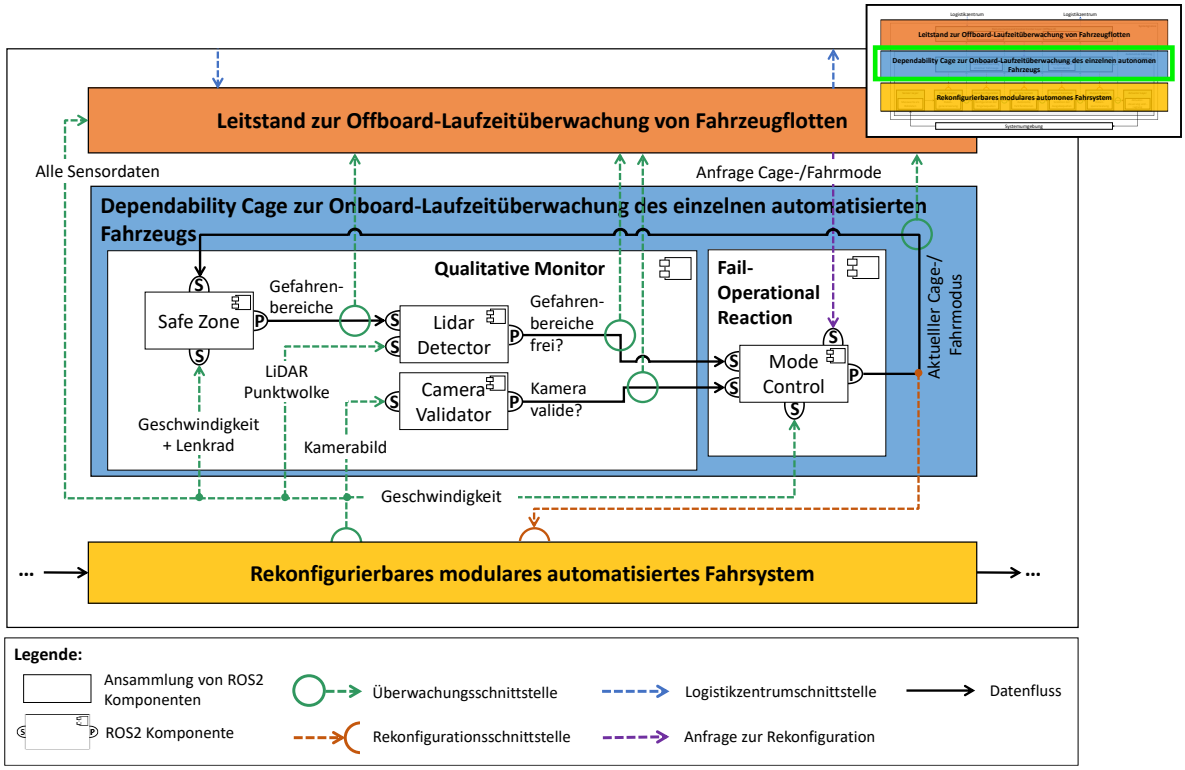

Abbildung 7: Abgeleitete Architektur-Instanz des Qualitativen Monitors in VanAssist

4.2 Remote Leitstand zur Offboard-Laufzeitüberwachung in VanAssist

Der Remote Leitstand dient als Schnittstelle zwischen Remote Operator und Fahrsystem. Der Remote Leitstand muss hierzu folgende Funktionalitäten berücksichtigen: (a) die Überwachung der Fahrzeugflotte, (b) die detaillierte Überwachung eines einzelnen automatisierten Fahrsystems und (c) die Realisierung der o. g. INs („Decision“ und „Action“). Die Besonderheit ist hierbei, dass es sich um ein verteiltes System handelt, welches mittels *ROS2* Einsicht und Bewertung der aktuellen Situation des Fahrsystems, sowie den menschlichen Eingriff bei Bedarf ermöglicht. In Abbildung 8 ist ein Screenshot der, im Rahmen von VanAssist, entwickelten grafischen Benutzeroberfläche abgebildet, welche die drei genannten Punkte berücksichtigt. Hierzu teilt sich die grafische Benutzeroberfläche in mehrere Teilbereiche mit unterschiedlichen Funktionalitäten auf. Drei wesentliche Teilbereiche sind „Flotten

Überwachung“, „Fahrzeug Überwachung“ und „Fahrzeug Rekonfiguration“.

Die „Flotten Überwachung“ stellt einen reduzierten Überblick über die zu überwachende Flotte, durch von Menschen bekannte Farbschemas, dar. Dazu werden nur relevante Informationen dargestellt. So repräsentiert z. B. ein roter Kreis beim „Driving Mode“ den Fahrmodus „Emergency Stop“. In einem Problemfall ist der Remote Operator in der Lage, das entsprechende Fahrzeug auszuwählen und einen detaillierten Einblick durch die „Fahrzeug Überwachung“ anzufordern. Diese umfasst den Kamerastream der vorderen und hinteren Kamera, eine Anzeige der Lidar-Punktwolke in Kombination mit der aktuellen „Safe Zone“ (Grün = „Clear Zone“, Orange = „Focus Zone“) und eine Anzeige der Karte mit aktueller Position des Fahrsystems und allen Missionszielen. Der Remote Operator hat, nachdem die Situation verstanden wurde, die Möglichkeit angemessen auf die Situation zu reagieren. Die grafische Benutzeroberfläche ermöglicht dies über die „Fahrzeug Rekonfiguration“. So besitzt z. B. der Remote Operator die Möglichkeit das Fahrsystem auf „Limited Autonomous Driving“ (bzw. „Limited Automated Driving“) zu rekonfigurieren, sodass die „Focus Zone“ bei der Laufzeitabsicherung nicht mehr berücksichtigt wird. Allerdings findet eine Limitierung der maximal zulässigen Geschwindigkeit des Fahrsystems statt. Das Fahrzeug (Fahrsystem und Dependability Cage) kann nun zusammen mit dem Remote Operator, sowie Leitstandsystem Situationen als kooperatives System bewältigen (IN „Decision“). Falls das Problem nicht aus der Ferne gelöst werden kann, kann lokale Unterstützung, z. B. durch telefonische Kontaktierung des Zustellers vor Ort, angefordert werden (IN „Action“).

Abbildung 8: Grafische Benutzeroberfläche des Remote Leitstands in VanAssist

5 EVALUATION UND ZUSAMMENFASSUNG

5.2 Evaluation des Gesamtsystems in der Testumgebung

Als Demonstrationsumgebung für die automatisierte Zustellfahrt dient der Campus Nord der TU Braunschweig. Fig. 5 zeigt den relevanten Bereich zusammen mit der weiß hinterlegten digitalen Karte und den für die Demonstration festgelegten Haltepunkten (H) sowie einem Parkbereich (P) und dem stilisierten Depot (D). Der beispielhafte Use-Case sieht dabei sowohl die Fahrt im Depot als auch im Zustellgebiet vor, wobei sich das Fahrzeug vollständig automatisiert zwischen den verschiedenen Haltestopps bewegt.

Der Use-Case des An- und Abfahrens von den dargestellten aber auch von frei wählbaren Haltepunkten innerhalb dieses Gebiets lässt sich mit dem vorgestellten ADS-System erfolgreich bewältigen. Zum Wenden im unkartierten Bereich des Depots wurde ein separater Algorithmus

entwickelt.

Zusätzlich ist auch eine Evaluierung der in Kapitel 4.1 dargestellten Szenarien erforderlich. Das Szenario der Engstelle wird, wie bereits aus Abbildung 8 deutlich wird an dem in Abbildung 9 dargestellten Bereich demonstriert. Das Sicherheitssystem erkennt dabei entsprechend der Ausführungen in Kapitel 4.2 die Gefahrensituation korrekt und leitet über die gewählte Rekonfigurationsanforderung die Lösung des Problems ein. Das automatisierte Fahrsystem setzt die erhaltene Anforderung des Not-Stopps erfolgreich um und durchfährt die Engstelle entsprechend der angeforderten Rekonfiguration mit reduzierter Geschwindigkeit sicher. Auch beim zweiten Szenario mit Einbeziehung einer externen Hilfe wird das Problem korrekt erkannt und der angeforderte Wechsel vom Nominal in den Fail-Operational Mode durch das Fahrsystem erfolgreich durchgeführt. Über das in Kapitel 3.1 erwähnte HMI wird in beiden Szenarios die Umwelt stets über die Fahrzeughandlungen informiert.

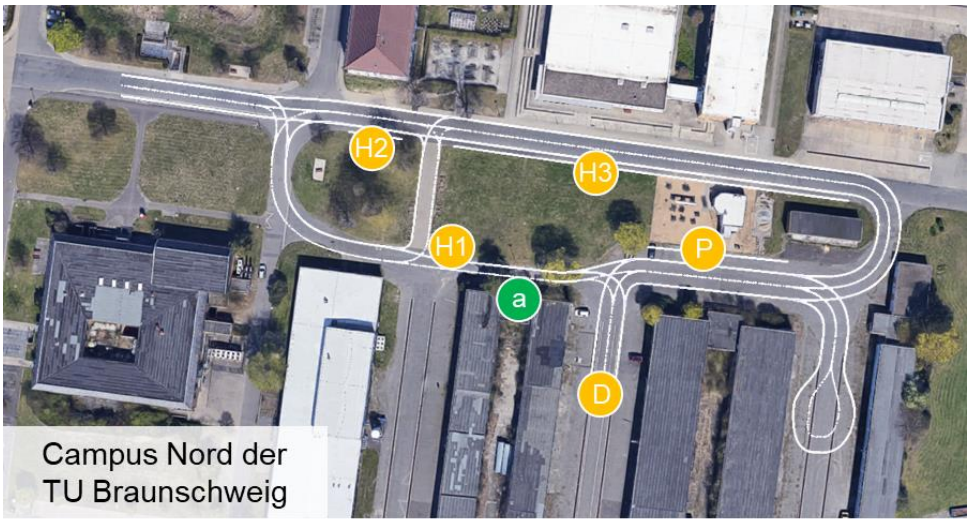

Abbildung 9: Demonstrationsumgebung mit hinterlegter digitalen Karte (weiß) und den für das Demo-Szenario relevanten Haltepunkten (H), dem Depot (D), einem Parkbereich (P) und der Engstelle (a).

5.2 Zusammenfassung

Aus den Ausführungen wird deutlich, dass zur Realisierung automatisierter Fahrsysteme im Allgemeinen und der Automatisierung der Paketzustellung im Projekt „VanAssist“ im speziellen ein umfangreiches Sicherheitskonzept notwendig ist, welches es ermöglicht, kritische und für das Fahrsystem nicht lösbare Aufgaben zu detektieren und durch eine Rekonfiguration oder externe Hilfe zu lösen. Das vorgestellte Sicherheitskonzept zur Lösung dieser Herausforderung besteht aus Fahrsystem, Onboard- und Offboard-Laufzeitüberwachung und konnte erfolgreich vom Konzept über die soft- und hardwareseitige Entwicklung bis hin zur Implementierung und Demonstration in der realen Testumgebung gebracht werden. Durch das Zusammenspiel der verschiedenen Teilsysteme mit der Möglichkeit zur Rekonfigurierbarkeit des Fahrsystems und den Wechsel vom Nominal Mode in den Fail-Operational Mode konnten hier kritische Situationen, wie z. B. Engstellen oder verdeckte Kameras, erfolgreich detektiert und durch die *INs* „Decision“ bzw. „Action“ gelöst werden. Dadurch konnte der Wirkradius des automatisierten Fahrzeuges deutlich vergrößert werden. Durch den Ansatz eines Remote Leitstandes (Offboard-Sicherheitsfahrer) ist das Konzept auf eine Fahrzeugflotte anwendbar.

ACKNOWLEDGEMENTS

Diese Arbeit entstand im Rahmen des Verbundprojekts "VanAssist - Interaktives, intelligentes System für autonome fernüberwachte Kleintransporter in der Paketlogistik" und wurde vom Bundesministerium für Verkehr und digitale Infrastruktur aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

REFERENCES

[1] SAE J3016: Levels of Driving Automation, SAE International, 2016.
[2] A. Aniculaesei et Al.: Towards A Holistic Software Systems Engineering Approach for Dependable Autonomous Systems, 1st International Workshop on Software Engineering for AI in Autonomous Systems, Gothenburg, Schweden, 2018.
[3] A. Aniculaesei et Al.: Graceful Degradation of Decision and Control Responsibility for Autonomous Systems based on Dependability Cages, 5th International Symposium on Future Active Safety Technology toward Zero, Blacksburg, Virginia, USA, 2019.
[4] J. Grieser et Al.: Assuring the Safety of End-to-End Learning-Based Autonomous Driving through Runtime Monitoring, 23rd Euromicro Conference on Digital System Design (DSD), Kranj, Slovenien, 2020.
[5] F. Buschmann et Al.: Pattern-oriented Software Architecture – A Pattern Language for Distributed Computing, Vol. 4, Joh Wiley & Sons, Ltd, England, 2007.
[6] T. Warnecke et Al.: Managing Communication Paradigms with a Dynamic Adaptive Middleware, ADAPTIVE 2018, 10th International Conference on Adaptive and Self-Adaptive Systems and Applications, Barcelona, Spanien, 2018.
[7] A. Asteroth, C. Baier: Theoretische Informatik – Eine Einführung in Berechenbarkeit, Komplexität und formale Sprachen mit 101 Beispielen, ISBN: 3-8273-7033-7, Pearson Studium, 2002.