

Dokumentation
in
Systems Engineering
Autonomes Fahren
Modellierung des Fahrzeugsystems mit SysML

Referent: Prof. Dr. Pascal Laube
Vorgelegt am: 25.01.2024
Vorgelegt von: SETEAM:
Marvin Roll
Robin Jendrusch
Stephan Kloess
Furkan Tasdemir
(Mazlum Ergin)
(Gerhard Friess)
(Dominic Moser)

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Bearbeiter Liste	3
Teammitglieder	3
Block-Definitions-Diagramm zur Fahrzeugarchitektur	4
Requirements-Diagramm - Kollisionsvermeidung	5
Use-Case-Diagramm - Kollisionsvermeidung	6
Use-Case-Diagramm - Trajektorien Planung.....	7
Requirements-Diagramm Trajektorien Planung	8
Requirements-Diagramm – Sensorset	9
Aktivitätsdiagramm	10
Internes Blockdefinitionsdiagramm.....	11
Sequenzdiagramm - Notbremssystem.....	13
Parametrisches Diagramm und Constraint Blocks für das Notbremssystem	14
Fazit	16

Bearbeiter Liste

Datum	Bemerkung / Änderung	Bearbeiter
13.11.2023	Blockdefinitionsdiagramm	Furkan Tasdemir, Mazlum Ergin
03.12.2023	Requirements-Diagramm Trajektorien Planung	Marvin Roll
03.12.2023	Requirements-Diagramm – Sensorset	Robin Jendrusch, Marvin Roll
06.01.2024	Parametrisches Diagramm und Constraint Blocks für das Notbremssystem	Robin Jendrusch, Marvin Roll
16.01.2024	Sequenzdiagramm Notbremssystem	Stephan Kloess, Marvin Roll
03.12.2023	Use-Case Diagramm - Trajektorien Planung	Robin Jendrusch
03.12.2023	Requirements-Diagramm Kollisionsvermeidung	Gerhard Friess, Robin Jendrusch
23.11.2023	Aktivitätsdiagramm	Furkan Tasdemir, Stephan Kloess, Mazlum Ergin
13.12.2023	Internes Blockdefinitionsdiagramm	Furkan Tasdemir, Mazlum Ergin
25.01.2024	Dokumentation und Präsentation	Präsentation präsentiert von Stephan Kloess; Dokumentation, Ausarbeitung und Erstellen von der Präsentation: Alle Teammitglieder

Teammitglieder

Name	Rolle	Zeit der Mitarbeit
Prof. Dr. Pascal Laube	Product Owner	(aktiv)
Stephan Kloess	Scrum Master	aktiv
Furkan Tasdemir	Entwickler	aktiv
Marvin Roll	Entwickler	aktiv
Robin Jendrusch	Entwickler	aktiv
Mazlum Ergin	Entwickler	29.09.23 - 16.01.24
Gerhard Friess	Entwickler	29.09.23 - 12.12.23
Dominic Moser	Entwickler	29.09.23 - 02.11.23

Block-Definitions-Diagramm zur Fahrzeugarchitektur

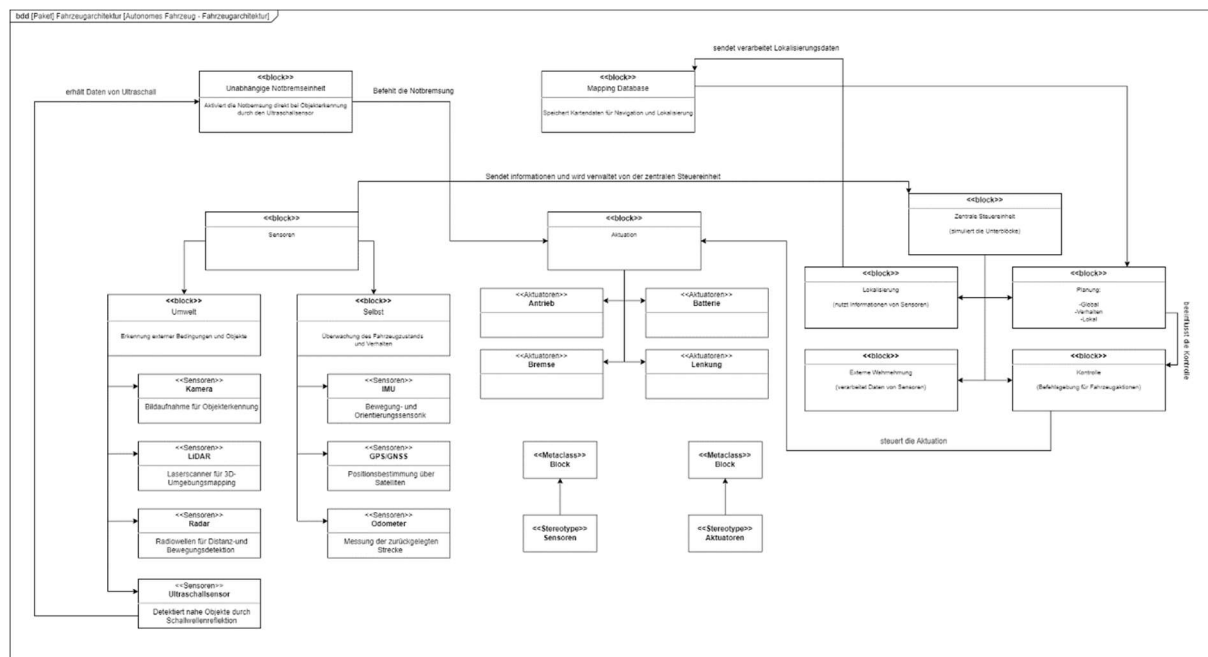


Diagramm auf [GitHub](#)

Hierin sind alle Komponenten aufgelistet, die für das autonome Fahren verwendet werden. Angefangen mit den Sensoren, die in zwei Gruppen unterteilt werden Umwelt und Selbst. Zur Gruppe Umwelt fallen alle Sensoren, die zur Erkennung externer Bedingungen und Objekten angewendet werden. Dies wären zum einen die Kamera, LiDAR, Radar und Ultraschallsensoren. Zur Überwachung des Fahrzeugzustands sind die Sensoren der Gruppe „Selbst“. Dazu gehören IMU-Sensoren zur Bewegungs- und Orientierungssensorik, GPS/GNSS und Odometer zur Messung der zurückgelegten Strecke. Diese Sensoren senden ihre Daten an die zentrale Steuereinheit, welche das autonome Fahren gewährleistet.

Dann gibt es noch die Aktuation. Diese beinhaltet den Antrieb, Batterie, Bremse und die Lenkung. Als weiteren Block gibt es dann noch die zentrale Steuereinheit. Diese beinhaltet vier Hauptfunktion.

1. Lokalisierung:
2. Planung:
3. Externe Wahrnehmung
4. Kontrolle

Es gibt noch zwei unabhängige Systeme, welche bestimmte Aufgaben ausführen. Zum einen das „Mapping Database“. Dieser speichert Kartendaten für die Navigation und Lokalisierung. Dies wird anhand der gesendeten Lokalisierungsdaten aus dem Block Lokalisierung ermöglicht.

Letzteres gibt es eine unabhängige Notbremseinheit, welche die Notbremsung direkt bei Objekterkennung durch den Ultraschallsensor aktiviert. Dieser sendet den Befehl der Notbremsung dann der Aktuation, wodurch die Bremse betätigt wird und eventuell eine Lenkung durchgeführt wird.

Requirements-Diagramm - Kollisionsvermeidung

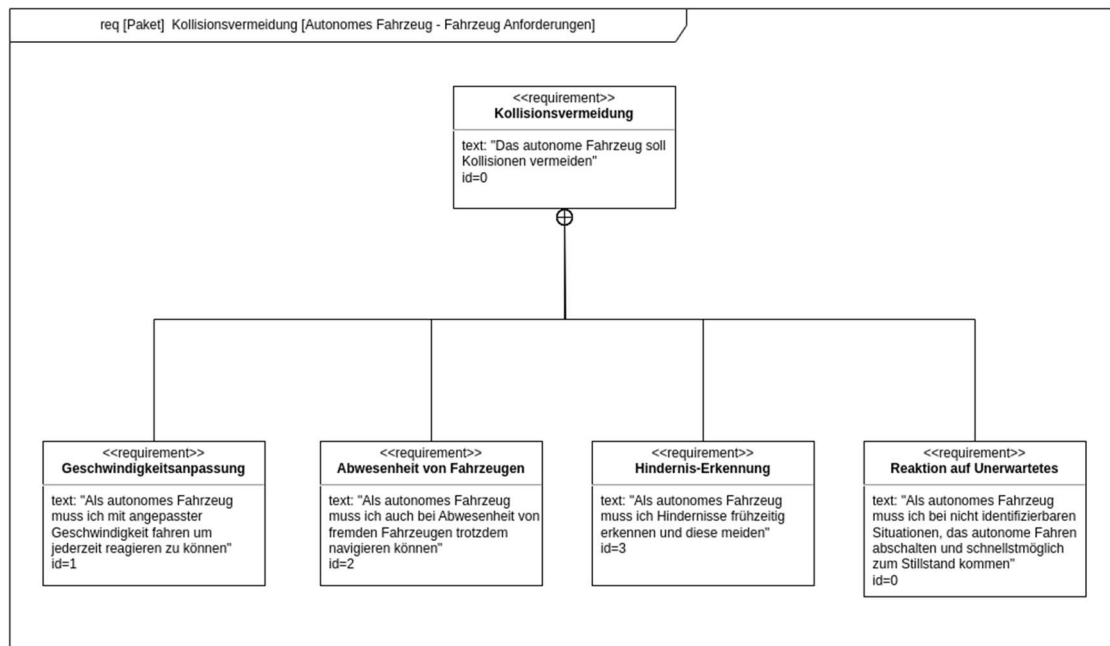


Diagramm auf [GitHub](#)

Das "Über"-Requirement "Kollisionsvermeidung" besteht aus mehreren "Unter"-Requirements:

- Geschwindigkeitsanpassung,
- Abwesenheit von Fahrzeugen,
- Hindernis Erkennung,
- Reaktion auf Unerwartetes

Diese werden mit dem zugehörigen Text genauer beschrieben.

Use-Case-Diagramm - Kollisionsvermeidung

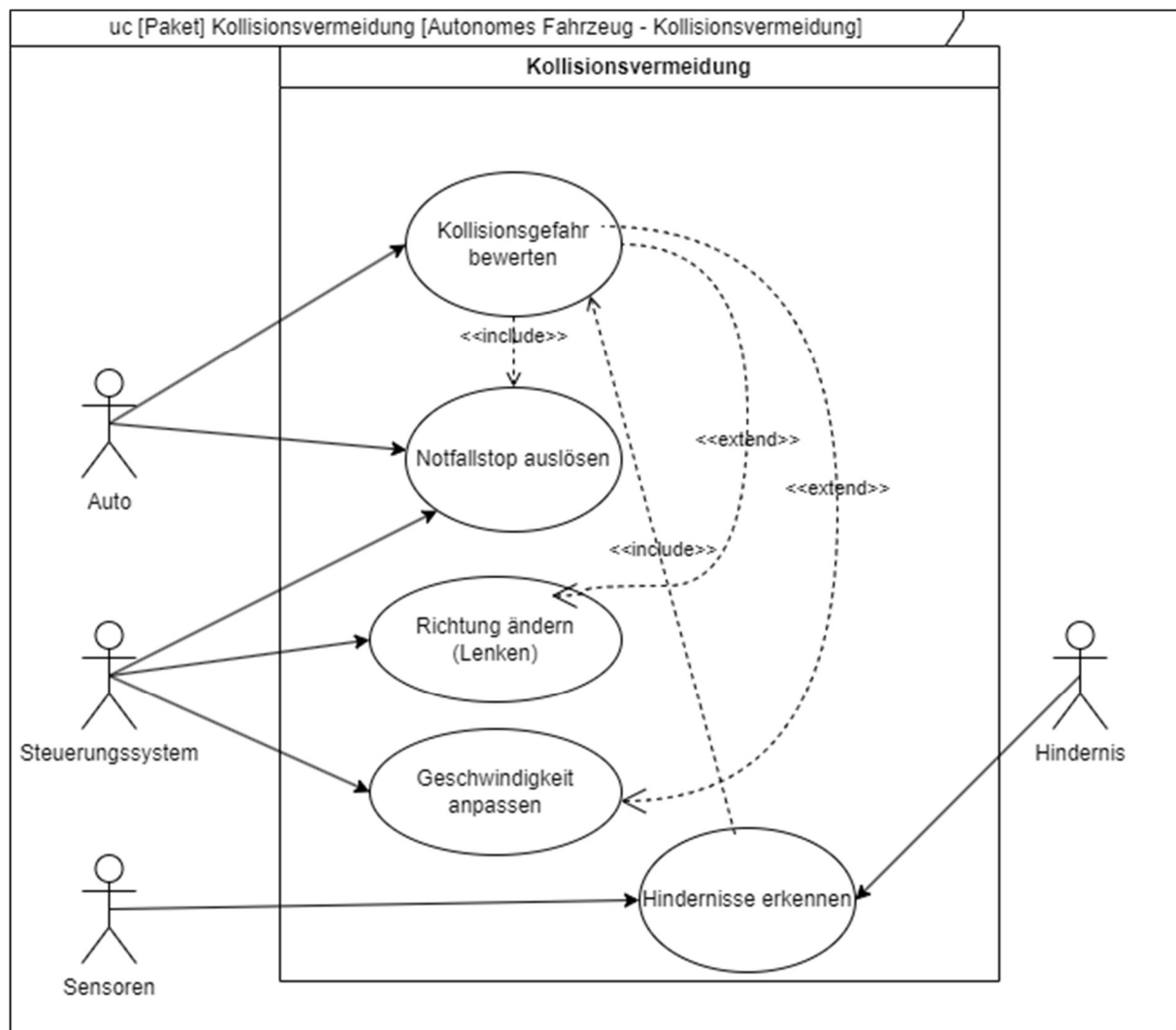


Diagramm auf [GitHub](#)

Es gibt folgende vier Akteure:

- Sensoren
- Steuerungssystem
- Auto
- Hindernis

Die Sensoren erkennen Hindernisse, das Auto bewertet die Kollisionsgefahr und führt bei Bedarf einen Notfallstopp durch. Das Steuerungssystem kann das Hindernis umfahren, oder diesem ausweichen, oder im Notfall ebenfalls einen Notfallstopp ausführen. Auch kann das Steuerungssystem die Geschwindigkeit bei Bedarf anpassen (Beschleunigen, verlangsamen, Bremsen). Zur Kollisionsvermeidung wurde die Funktionen des Ausweichens / Lenkens / Geschwindigkeit anpassen berücksichtigt und diese miteinander verbunden.

Der Akteur "Hindernis" wurde nachträglich hinzugefügt.

Use-Case-Diagramm - Trajektorien Planung

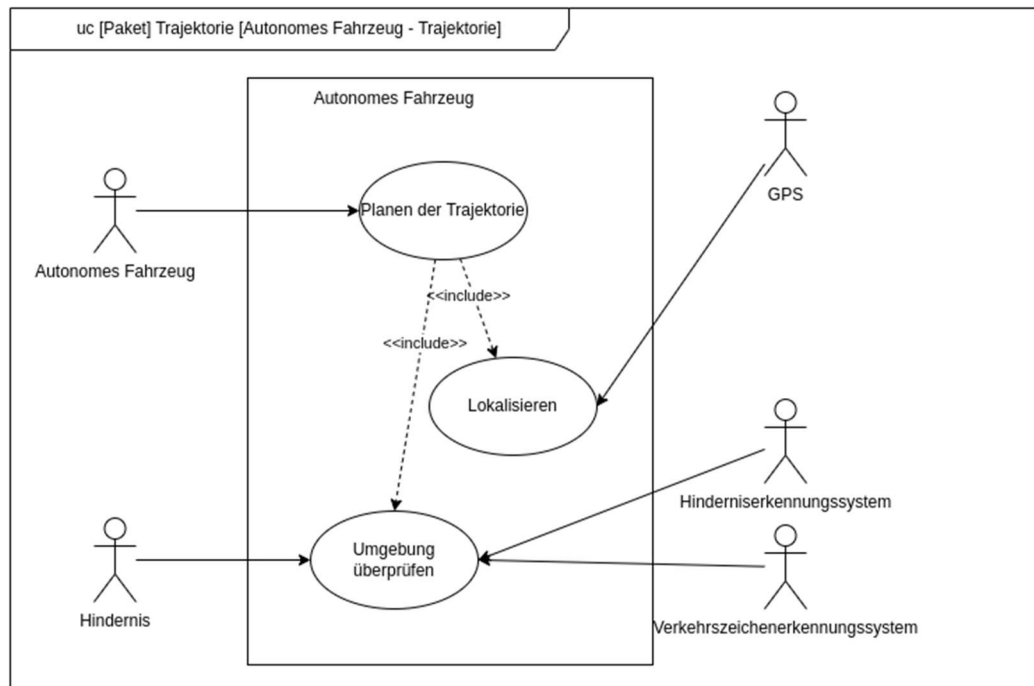


Diagramm auf [GitHub](#)

Akteure:

- Autonomes Fahrzeug
- Hinderniserkennungssystem
- Verkehrszeichenerkennungssystem
- GPS
- Hindernis

Beziehungen:

Hinderniserkennungssystem sendet Informationen an "Planen der Trajektorie" und "Umgebung überprüfen",

"GPS" und "Lokalisieren" unterstützen das "Planen der Trajektorie" und "Umgebung überprüfen",

"Planen der Trajektorie" sowie "Umgebung überprüfen" beeinflussen das "Autonome Fahrzeug".

"Hindernis" beeinflusst direkt das "Autonome Fahrzeug".

Requirements-Diagramm Trajektorien Planung

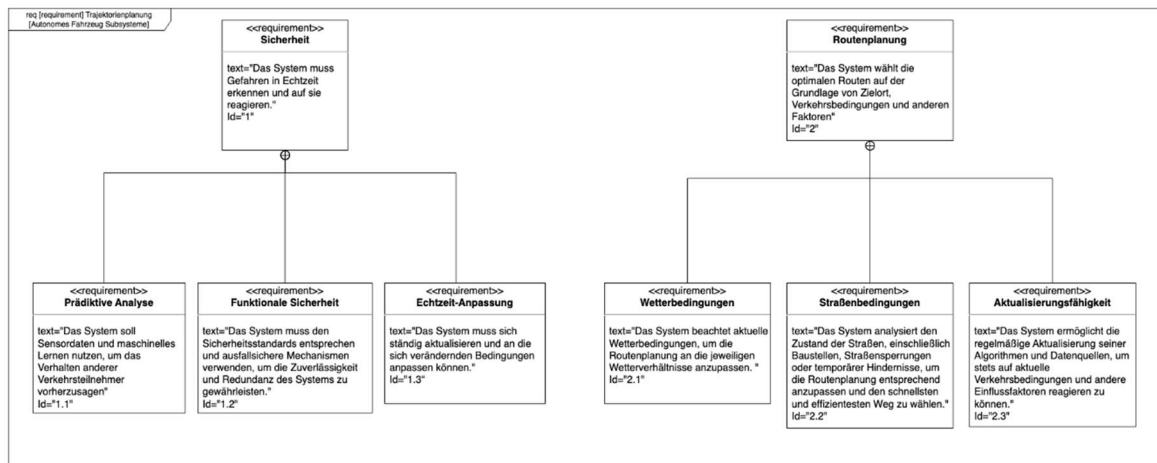


Diagramm auf [GitHub](#)

Das Requirements-Diagramm legt die Anforderungen für die Trajektorienplanung in autonomen Fahrzeugen fest, wobei der Schwerpunkt auf Sicherheit und Routenplanung liegt.

Im Sicherheitsbereich werden detaillierte Sub-Anforderungen definiert, um eine umfassende Sicherheitsstruktur zu gewährleisten. Hierbei liegt besonderes Augenmerk auf der präzisen Analyse von Sensordaten, der Einhaltung strenger Sicherheitsstandards und der Fähigkeit des Systems zur Echtzeit-Reaktion auf unvorhergesehene Ereignisse.

Die Routenplanung wird ebenfalls über das grundlegende Requirement hinaus erweitert und berücksichtigt verschiedene Aspekte, um eine robuste und flexible Planung zu ermöglichen. Dies schließt die Anpassung an sich verändernde Wetterverhältnisse, temporäre und permanente Straßenbedingungen sowie die generelle Aktualisierbarkeit der Routen ein. Die Berücksichtigung von sich dynamisch ändernden Faktoren stellt sicher, dass das autonome Fahrzeug in der Lage ist, sich an die aktuelle Verkehrssituation anzupassen und die sicherste und effizienteste Route zu wählen. Es wird außerdem darauf geachtet, dass die Routenplanung kontinuierlich aktualisiert wird, um auf neue Informationen oder veränderte Bedingungen reagieren zu können.

Requirements-Diagramm – Sensorset

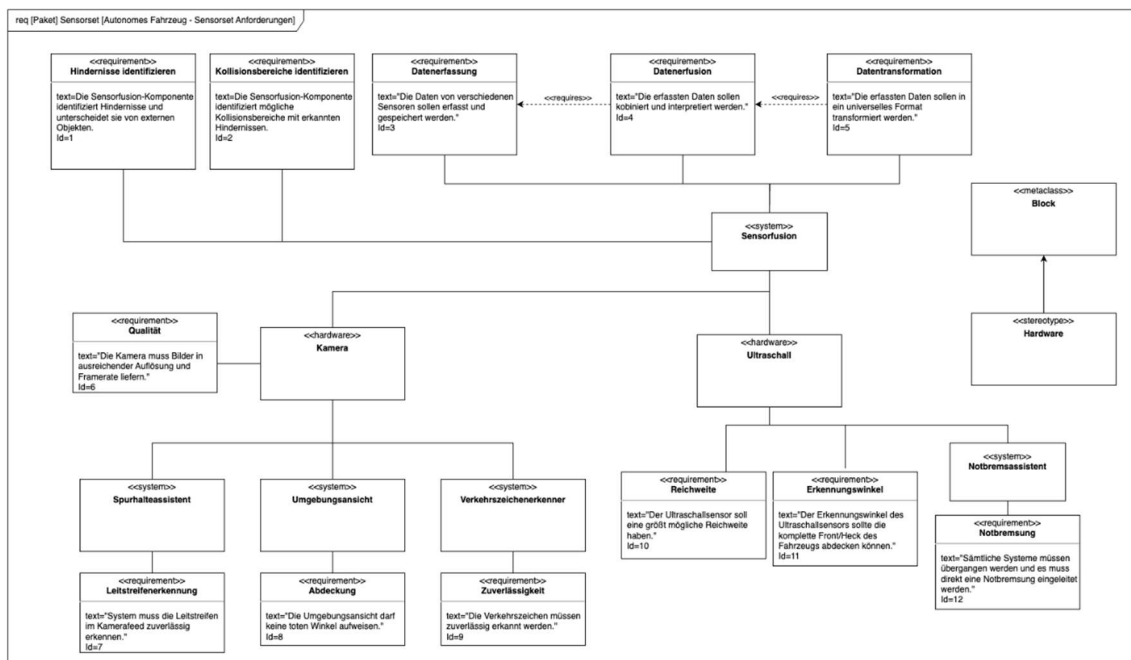


Diagramm auf [GitHub](#)

Das Requirements-Diagramm legt spezifische Anforderungen für verschiedene Systeme und Sensoren fest. Eine direkte Zuordnung von Anforderungen an die Sensorqualität erfolgt dabei unmittelbar zu den entsprechenden Sensoren. Dies gewährleistet, dass die Leistungsanforderungen an die Sensoren klar definiert sind und eine qualitativ hochwertige Datenerfassung ermöglicht wird.

Die Assistenzsysteme sind wiederum jeweils ihren spezifischen Sensoren zugeordnet und erhalten ebenfalls klare Anforderungen, um eine präzise Datenverarbeitung und -interpretation zu gewährleisten.

Ein weiterer Schwerpunkt des Diagramms liegt auf den Anforderungen an die Sensorfusion, die den Prozess der Integration und Zusammenführung der Daten aus verschiedenen Sensoren betrifft. Hierbei werden auch Abhängigkeiten zu anderen Anforderungen deutlich herausgearbeitet, um sicherzustellen, dass die Sensorfusion optimal auf die Gesamtfunktionalität des autonomen Fahrzeugs abgestimmt ist.

Durch diese detaillierte Zuordnung und Festlegung von Anforderungen wird sichergestellt, dass jedes System und jeder Sensor klar definierte Leistungsmerkmale erfüllt und die Gesamtfunktionalität des autonomen Fahrzeugs auf höchstem Standard operiert.

Aktivitätsdiagramm

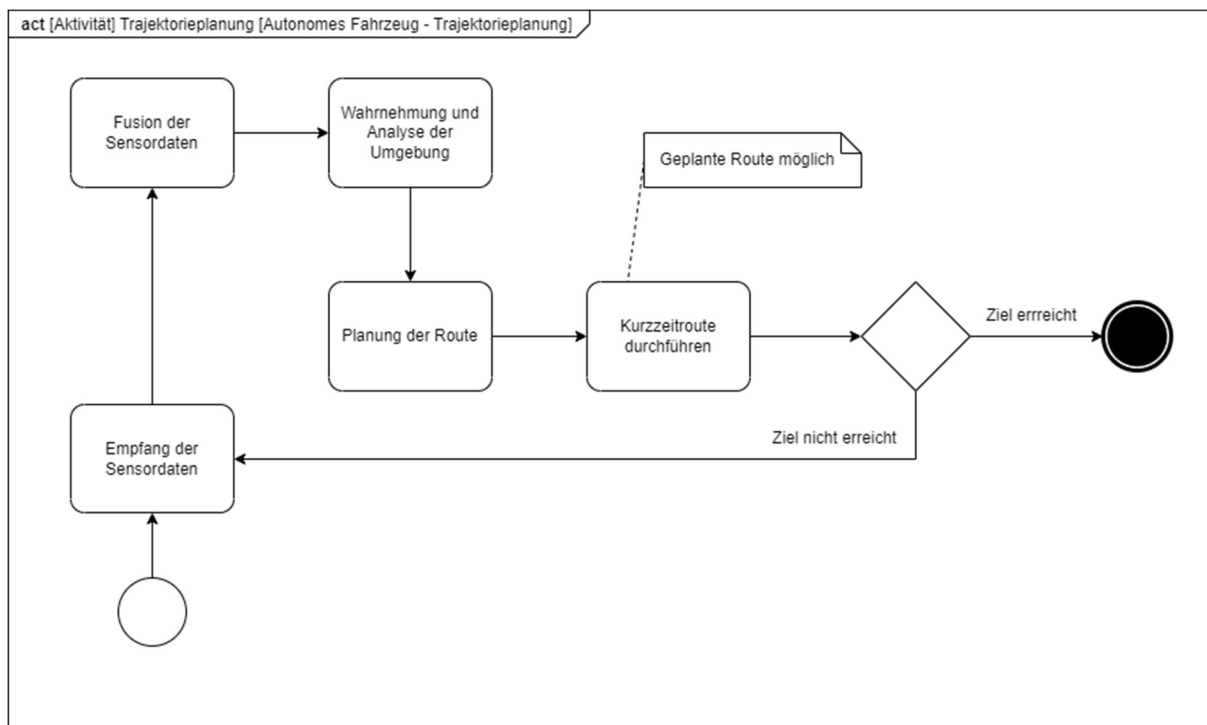


Diagramm auf [GitHub](#)

Die Aktivität wird gestartet, indem die Sensordaten von den verschiedenen Sensoren erfasst und an Bordcomputer übermittelt werden. Dort erfolgt die Fusionierung der Daten, um ein detailliertes Bild der Umgebung zu erstellen. Die fusionierten Daten werden analysiert, um die Umgebung zu interpretieren und Hindernisse zu identifizieren.

Basierend auf dieser umfassenden Wahrnehmung und Analyse der Umgebung plant der Bordcomputer eine Route. Daraufhin wird eine Kurzzeitroute durchgeführt, worin die Länge vordefiniert wird (z.B. soll für 2 Sekunden die Route durchgeführt werden). Nach Durchführung der Route überprüft das System, ob das Ziel erreicht wurde. Ist das Ziel erreicht, endet die Aktivität.

Falls dies nicht der Fall ist, so wird der Durchgang erneut durchgeführt, beginnend mit dem Empfang neuer Sensordaten. Bei der Aktivität „Planung der Route“ wird die Route auch ständig aktualisiert, falls Hindernisse oder ähnliches ergeben.

Internes Blockdefinitionsdiagramm

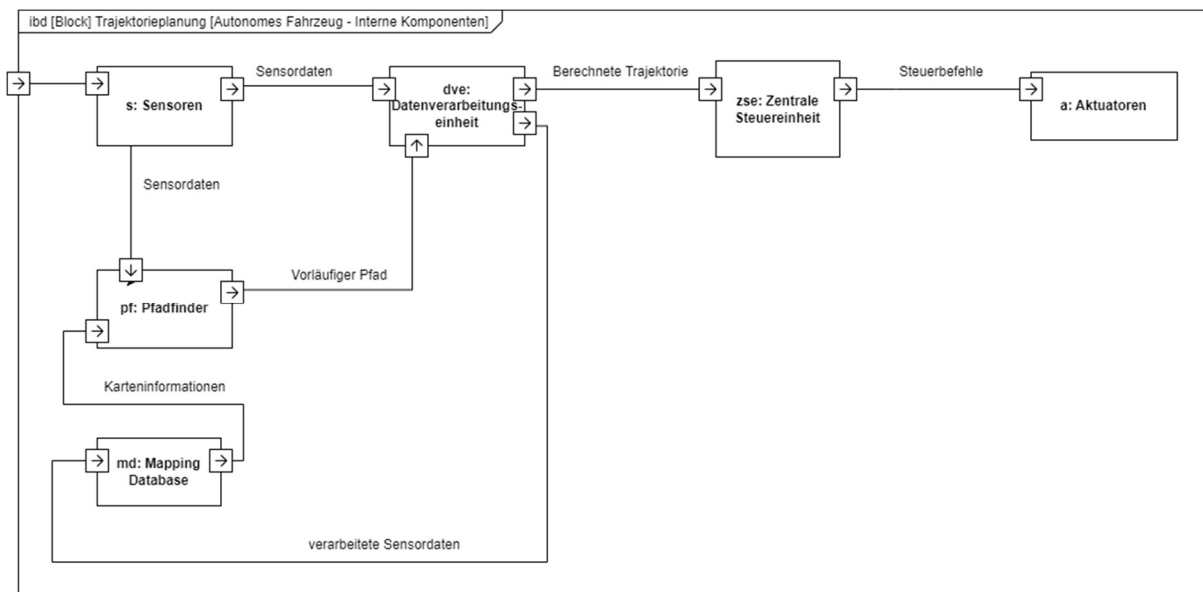


Diagramm auf [GitHub](#)

Sensoren:

- Erfassen kontinuierlich Daten über die Umgebung des Fahrzeugs, einschließlich Hindernisse, Verkehr, Straßenmerkmale etc.
- Senden die gesammelten Daten an den Pfadfinder und die Datenverarbeitungseinheit zur weiteren Analyse.

Pfadfinder (Pathfinder):

- Empfängt Eingangsdaten von den Sensoren und der Mapping Database.
- Nutzt Algorithmen zur Pfadfindung, um auf Basis dieser Daten einen vorläufigen Pfad zu erstellen.
- Sendet den vorläufigen Pfad zur Datenverarbeitungseinheit für die Trajektorien Berechnung

Mapping Database:

- Beinhaltet detaillierte Karteninformationen, die für die Navigation und Pfadfindung erforderlich sind.
- Stellt die Kartendaten dem Pfadfinder zur Verfügung, um die Pfadfindung zu unterstützen.
- Erhält Daten von der Datenverarbeitungseinheit für die Kartenerstellung
- Datenverarbeitungseinheit:
 - Erhält den vorläufigen Pfad vom Pfadfinder und Sensordaten direkt von den Sensoren.
 - Verarbeitet diese Informationen, um eine sichere und effiziente Fahrttrajektorie zu berechnen, die Hindernisse vermeidet und die Fahrzeugdynamik berücksichtigt.
 - Sendet die berechnete Trajektorie an die zentrale Steuereinheit, die darauf basierend Steuerbefehle erstellt.

Zentrale Steuereinheit:

- Empfängt die endgültige Trajektorie von der Datenverarbeitungseinheit.
- Wandelt die Trajektorie in Steuerbefehle um, wie Lenkbewegungen, Beschleunigung und Bremsung.
- Sendet die Steuerbefehle an die Aktuatoren, um die Befehle physisch umzusetzen.
Aktuatoren:
- Empfangen Steuerbefehle von der zentralen Steuereinheit.
- Führen physische Aktionen aus, die zum Fahren des Fahrzeugs erforderlich sind, wie das Betätigen der Lenkung, Beschleunigen, Bremsen etc.

Sequenzdiagramm - Notbremssystem

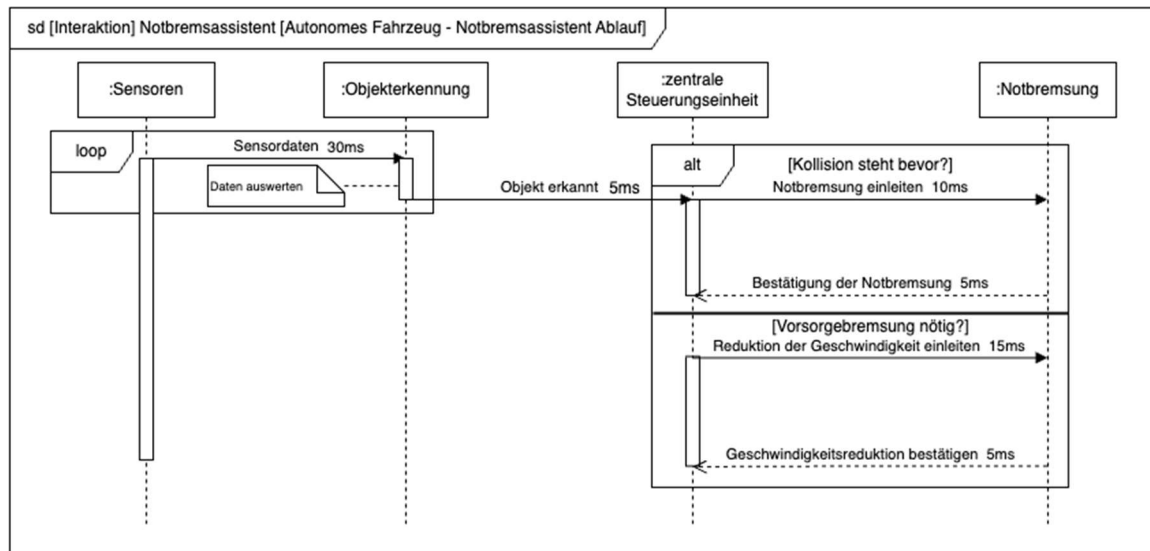


Diagramm auf [GitHub](#)

Die Sensoren des autonomen Fahrzeugs übermitteln kontinuierlich Daten im Intervall von 30 Hertz an die Objekterkennung. Sobald die Objekterkennung ein relevantes Objekt identifiziert, werden die entsprechenden Daten mit einer Latenz von etwa 5 Millisekunden an die zentrale Steuereinheit weitergeleitet.

Die zentrale Steuereinheit analysiert die empfangenen Daten eingehend und trifft anschließend Entscheidungen bezüglich der erforderlichen Maßnahmen. Dies kann die Einleitung einer Notbremsung oder auch die Aktivierung einer präventiven Bremsung sein, je nach der erkannten Gefahrensituation.

Falls keine unmittelbare Aktion erforderlich ist, setzt die zentrale Steuereinheit den Auswertungsprozess fort, um die nächsten empfangenen Daten zu analysieren. Dieser iterative Prozess gewährleistet eine fortlaufende Überwachung der Umgebung des Fahrzeugs und ermöglicht eine schnelle Reaktion auf potenzielle Gefahrensituationen.

Parametrisches Diagramm und Constraint Blocks für das Notbremssystem

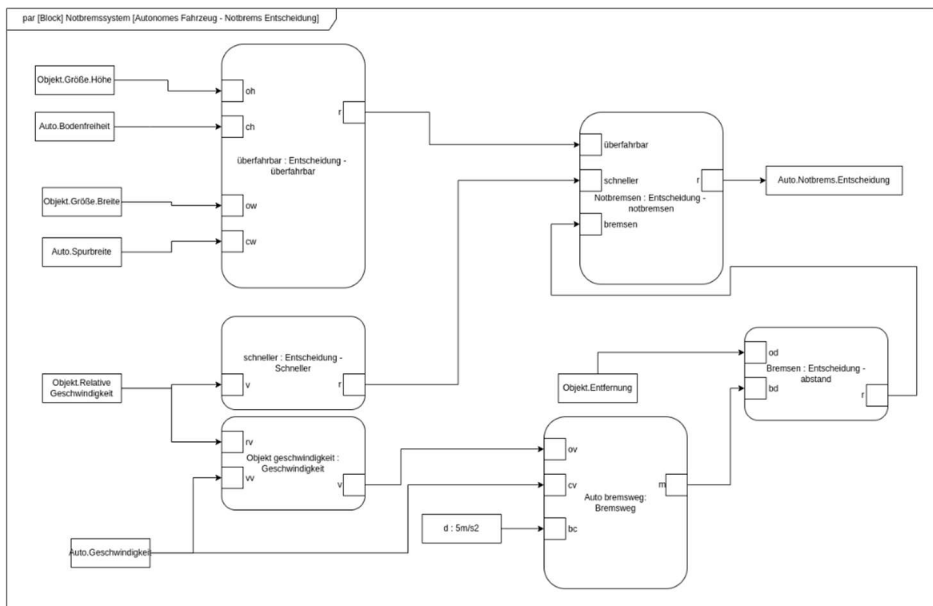


Diagramm auf [GitHub](#)

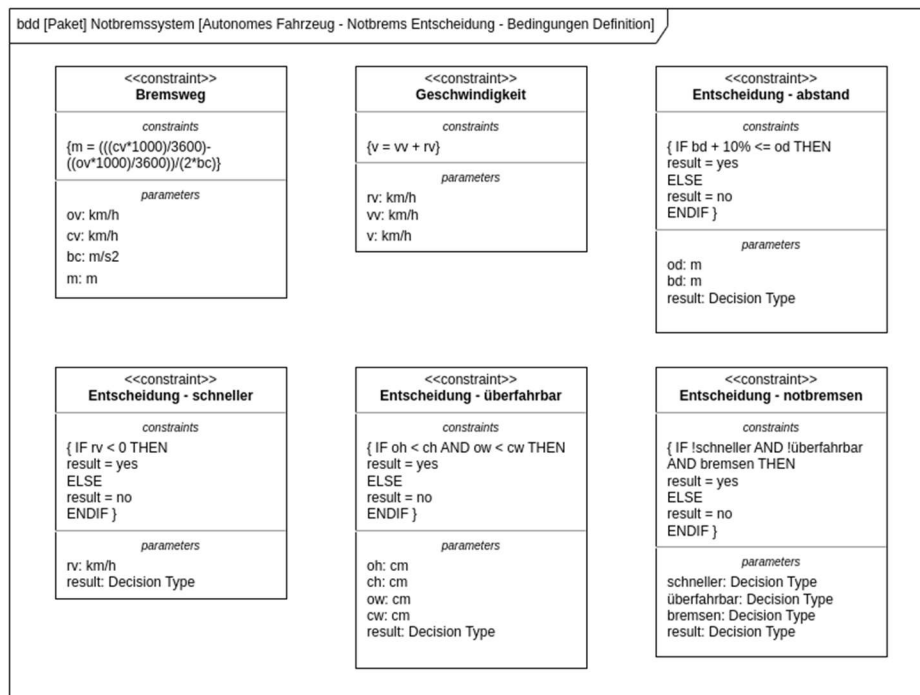


Diagramm [GitHub](#)

Das parametrische Diagramm veranschaulicht den Entscheidungsprozess für das Einleiten einer Notbremsung und berücksichtigt dabei verschiedene Parameter und Kriterien. Die nachfolgenden Entscheidungen und Berechnungen werden dabei durchgeführt:

Überfahrbar:

Überprüfung anhand von Bodenfreiheit und Spurbreite des Fahrzeugs, ob diese größer sind als die Objekthöhe und Objektbreite. Diese Überprüfung bestimmt, ob das Fahrzeug, das erkannte Objekt überfahren kann.

Schneller:

Entscheidung anhand der relativen Geschwindigkeit, ob das erkannte Objekt schneller ist als das Fahrzeug. Dies trägt zur Beurteilung der potenziellen Gefahr bei.

Geschwindigkeit:

Ermittlung der Objektgeschwindigkeit durch Berücksichtigung der Fahrzeuggeschwindigkeit und relativen Geschwindigkeit zwischen Fahrzeug und Objekt.

Bremsweg:

Berechnung des Bremswegs in Metern durch Einbeziehung der Objektgeschwindigkeit und der Fahrzeuggeschwindigkeit. Dieser Parameter gibt Auskunft über den benötigten Bremsweg des Fahrzeugs.

Abstand:

Entscheidung, ob die Entfernung zum erkannten Objekt kleiner ist als der berechnete Bremsweg plus einen Sicherheitsabstand von 10%.

Notbremsen:

Durch die Kombination der Entscheidungen Überfahrbar, Schneller und der Beurteilung des Abstands wird festgelegt, ob eine Notbremsung eingeleitet werden muss. Dieser umfassende Prozess stellt sicher, dass das Fahrzeug in kritischen Situationen angemessen reagiert und rechtzeitig eine Notbremsung durchführt, wenn dies erforderlich ist.

Fazit

Unser Projekt war geprägt von einer insgesamt positiven Erfahrung, jedoch trübten Probleme in der Kommunikation mit zwei Gruppenmitgliedern das Gesamtbild. Diese schlechte Kommunikation führte dazu, dass wir zu viert eine beträchtliche Menge Arbeit bewältigen mussten, insbesondere da wir auch die Aufgaben der ausgeschiedenen Teammitglieder unter Zeitdruck übernehmen und nacharbeiten mussten.

Trotz dieser Herausforderungen waren die uns gestellten Aufgaben ansonsten gut formuliert und leicht verständlich. Es machte sogar Spaß, an ihnen zu arbeiten, und die Überschaubarkeit der Aufgaben trug dazu bei, dass wir effizient vorankommen konnten.

Ein weiterer Aspekt, der unser Projekt beeinflusste, war die Nutzung von Jira. Die Plattform stellte sich als gewöhnungsbedürftig heraus und brachte einige Unannehmlichkeiten mit sich. Möglicherweise sollten wir in Erwägung ziehen, beim nächsten Mal auf alternative Tools wie Open Projects oder ähnliche Plattformen umzusteigen, um einen einfacheren Einstieg in Tools für agiles Arbeiten mit Scrum zu gewährleisten.

Trotz der genannten Herausforderungen können wir unser Projekt insgesamt als "gut/okay" bewerten. Es war eine lehrreiche Erfahrung, und wir konnten die gestellten Aufgaben erfolgreich bewältigen, auch wenn die Kommunikation mit einigen Gruppenmitgliedern verbessert werden könnte.