

# Schlussbericht



## SET Level

Simulationsbasiertes Entwickeln und Testen von automatisiertem Fahren

*Beitrag des Zuwendungsempfängers:*

**Rheinisch-Westfälische Technische Hochschule Aachen**

Fakultät 4 – Maschinenwesen  
Institut für Kraftfahrzeuge  
Steinbachstr. 7  
52074 Aachen

*Laufzeit:*

01.03.2019 – 31.10.2022

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Wirtschaft und Energie unter dem Förderkennzeichen 19A19004C gefördert.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Version: 1.0  
Stand: 31.12.2022



Bundesministerium  
für Wirtschaft  
und Klimaschutz

## Inhaltsverzeichnis

<b>1</b>	<b>Kurze Darstellung .....</b>	<b>3</b>
1.1	Aufgabenstellung .....	3
1.2	Voraussetzungen für das Vorhaben .....	3
1.3	Planung und Ablauf.....	4
1.4	Stand der Wissenschaft und Technik .....	6
1.5	Zusammenarbeit mit anderen Stellen .....	7
<b>2</b>	<b>Eingehende Darstellung .....</b>	<b>8</b>
2.1	Erzielte Ergebnisse .....	8
2.1.1	Teilprojekt 2: Simulationsbasiertes und virtuelles Entwickeln / Testen.....	8
2.1.2	Teilprojekt 3: Modellspezifikation, -Entwicklung und –Validierung .....	11
2.2	Zahlenmäßiger Nachweis.....	22
2.3	Notwendigkeit der Förderung.....	22
2.4	Nutzen und Verwertbarkeit .....	23
2.5	Bekannt gewordener Fortschritt .....	23
2.6	Veröffentlichungen.....	24
2.6.1	Erfolgte Veröffentlichungen .....	24
<b>3</b>	<b>Abbildungsverzeichnis.....</b>	<b>25</b>
<b>4</b>	<b>Tabellenverzeichnis .....</b>	<b>25</b>
<b>5</b>	<b>Literaturverzeichnis.....</b>	<b>26</b>

# 1 Kurze Darstellung

## 1.1 Aufgabenstellung

Das ika hat in SET Level verschiedene Arbeitspakete in den Teilprojekten 2 und 3 bearbeitet, wobei der Fokus des Personalaufwands auf letzterem lag.

Im Teilprojekt 2 wurden die Themen Systemarchitektur, Szenariendefinition und Modellintegration, -kopplung bearbeitet. Ziel war dabei eine einheitliche Struktur zu gestalten, mit der sich Szenarien skalierbar simulieren lassen. Dazu unterstützte das ika die Entwicklung der Schnittstellenarchitektur des Gesamtsystems. In weiteren Arbeitspaketen wurden Standards und Spezifikationen für die Szenarienbeschreibung weiterentwickelt, sowie die notwendige Beschreibung der Beziehung zwischen Simulationsmodellen und Szenarien entwickelt. Zur Sicherstellung der korrekten und realistischen Funktionsweise der Modelle – einzeln und im Zusammenspiel mit anderen Modellen – waren die Gestaltung und Definition von Anforderungen zur Validierung und Qualifizierung vorgesehen.

Für das Teilprojekt 3 war maßgeblich die gezielte Entwicklung von Simulationsmodellen für das skalierbare Testen von automatisierten Fahrfunktionen vorgesehen. Zwei wichtige Punkte waren dabei die geschlossene Simulation (closed-loop) sowie die automatische Generierung der Strecke für notwendige Variationen. Die notwendigen Modellerweiterungen im Fahrermodell und Streckenmodell waren Forschungsfokus des ika. Die im Projekt benötigten, Modelle wurden spezifiziert und der Aufbau im Sinne der Prozesse und Methoden aus TP2 definiert. Das ika ist insbesondere im Bereich Verkehrsmodellierung und Umfeldmodellierung tätig geworden und hat dabei entsprechende Algorithmen erstellt und kalibriert. Zudem leistete das ika einen Beitrag im Bereich der Modellauswahl und Integration, deren Mechaniken es erlauben, geeignete Modelle für spezifische Use-Cases zu identifizieren und in die Simulation zu integrieren. Eine wichtige Aufgabe für das ika war die automatische Absicherung der Modelle. Aufbauend auf den obigen Aufgaben, wurden Methoden und Spezifikationen des Datenmanagements in Bezug auf die Modelle (Verwaltung der Parametersets, Metadaten) entwickelt.

## 1.2 Voraussetzungen für das Vorhaben

SET Level wurde als Folgeprojekt von PEGASUS konzipiert. Das Projekt PEGASUS (Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen) mit einer Laufzeit von 2016 bis 2019 hatte das Ziel, Basismethoden für die Automatisierungslevel 3 und 4 im Einsatz auf Autobahnen zu entwickeln (siehe Abbildung 1).

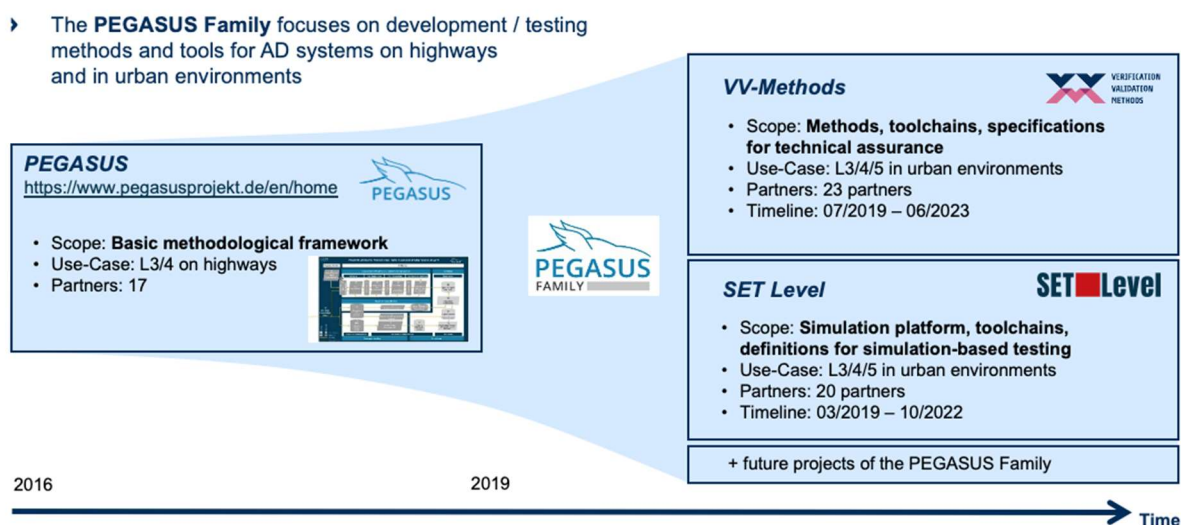


Abbildung 1: Projekte der PEGASUS-Familie

SET Level wurde speziell für den Einsatz von Simulation beim Entwickeln und Testen hoch automatisierter Fahrfunktionen in urbanen Umgebungen konzipiert. Dabei wurden wesentliche Partner aus der deutschen Automobilindustrie, Zulieferer, IT-Vendoren und renommierten Forschungseinrichtungen zusammengeführt, um Kompetenzen, Methoden und Technologien gemeinsam aufzubauen (siehe Abbildung 2).

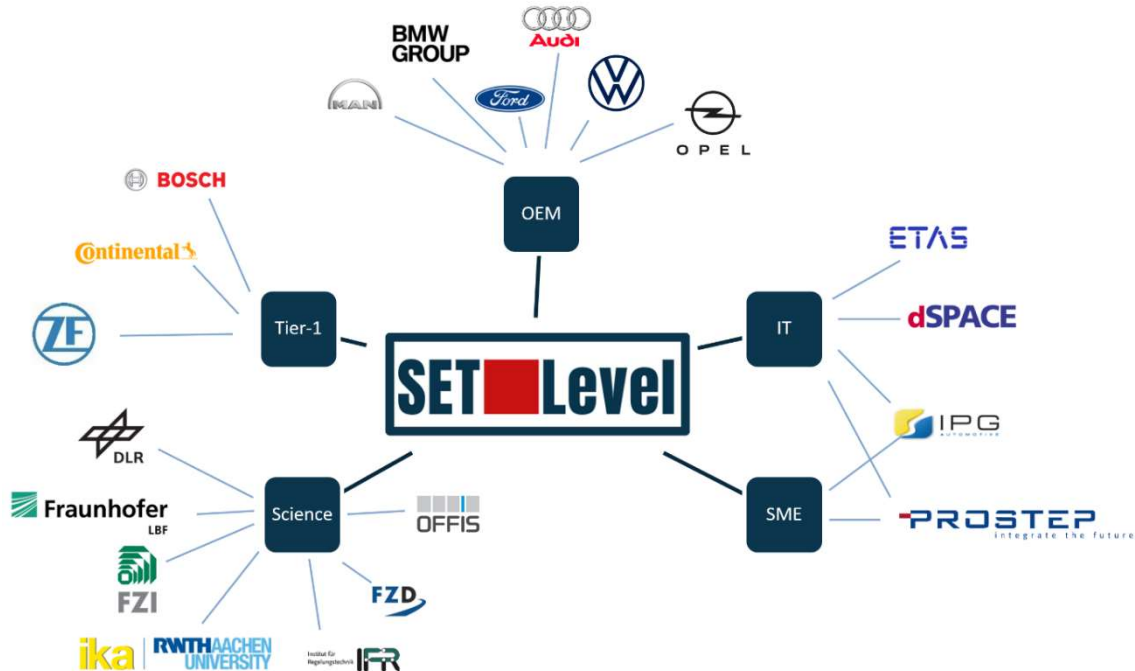


Abbildung 2: Zusammensetzung der Partner von SET Level

### 1.3 Planung und Ablauf

Grundansatz zur Erschließung und Bearbeitung der komplexen Fragestellung bildete ein iteratives Vorgehen unter schrittweiser Komplexitätssteigerung zur gemeinsamen Durchdringung und Entwicklung des Themas (siehe Abbildung 3).

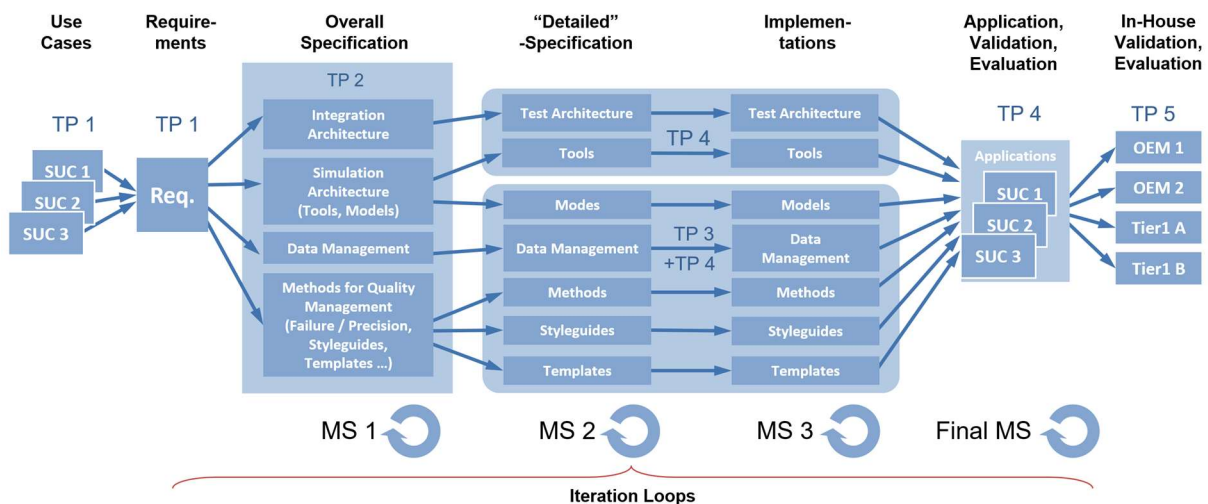


Abbildung 3: Schnelle Iterationen in Projektmeilensteinen

Als Erfolgsmerkmal und -nachweis wurden so genannte Simulation Use Cases (SUCs) konzipiert. Um einen Erfolgsnachweis und eine Basis für die Reflektion der gewählten Ansätze zu schaffen, wurden zu definierten Meilensteinen Demonstratoren implementiert und evaluiert. Dieses Vorgehen war entscheidend für ein schnelles Lernen und für den Nachweis, dass die Grundideen von SET Level, die modularisierte und auf Standards basierenden Architekturen von Simulationstools, auch wirklich funktionieren.

Ausgehend von den Use Cases wurden im Teilprojekt 1 (TP1) Anforderungen (Requirements) an Simulationstools spezifiziert. Inhaltlich fand die Konzeption (Specification) des modularen und standardbasierten Aufbaus von Simulationstools im Teilprojekt 2 (TP2) statt. Die detaillierte Spezifikation notwendiger Prozesse und die Implementierung von Modellen wurde im Teilprojekt 3 (TP3) durchgeführt. Die Ausführung von Simulationen auf unterschiedlichen Simulationstools (von der Implementierung bis zur Evaluation) fand in Teilprojekt 4 (TP4) statt. Die Sicherstellung der industriellen Anwendbarkeit erfolgte in Teilprojekt 5 (TP5) durch regelmäßige Bewertungen des aktuellen Entwicklungsstands sowie entsprechendes Feedback.

BMW und das DLR koordinierten das Projekt als Ganzes. Die Arbeit erfolgte in einer Vielzahl von Teilprojekten (siehe Abbildung 4).

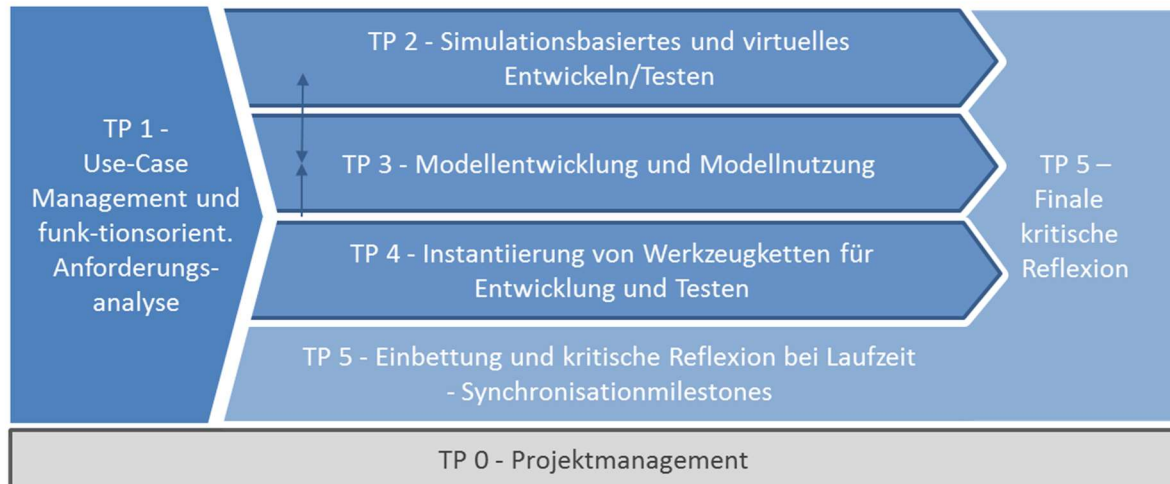


Abbildung 4: Aufbau der Teilprojekte von SET Level

### Zeitplan

In Abbildung 5 ist eine Gesamtübersicht der Projektlaufzeit dargestellt. Die verschiedenen Farben sind qualitative Intensitätsangaben, grob eingeteilt in geringe, mittlere und hohe Kapazitätsbedarfe der jeweiligen Teilprojekte. Wie zu erkennen ist, hatte Teilprojekt 1 als Anforderungs- und Synchronisationsprojekt zu VVMMethoden einen starken Fokus im ersten Jahr und benötigte im weiteren Verlauf immer weniger personelle Ressourcen, um die Kommunikation zu VVMMethoden zu gewährleisten. Die Teilprojekte 2 und 3 hatten ihre Belastungsspitze beginnend mit Q3 über den größten Teil des zweiten Projektjahres. Teilprojekt 4 hatte einen gleichmäßigen Kapazitätsbedarf über die Projektlaufzeit, während der von Teilprojekt 5 über den Projektverlauf immer höher wurde und gegen Ende am höchsten war. In Abbildung 5 sind auch drei Gesamtmeilensteine jeweils zum Ende eines Projektjahres verzeichnet, deren Hauptergebnisse als Prototyp, Vorversion, Endversion der Werkzeugketten verstanden werden können.

	Projektjahr 1				Projektjahr 2				Projektjahr 3				Projektjahr 4		
Arbeitspaket	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
0															
1.1															
1.2															
2.1															
2.2															
2.3															
3.1															
3.2															
3.3															
4.1															
4.2															
4.3															
4.4															
5.1															
5.2															
5.3															

Abbildung 5: Zeitplanung mit Meilenstein- und Abschlussberichten

Dabei war als Erfolgsmerkmal die Konzeption von Simulation Use Cases (SUCs) und deren Implementierung in Demonstratoren entscheidend für schnelles Lernen und für den Nachweis, dass die Grundideen von SET Level, die modularisierte und auf Standards basierenden Architekturen von Simulationstools auch wirklich funktionieren.

Als zielführend hat sich die Organisation sogenannter Quartalsmeetings bewährt, bei denen alle Projektbeteiligten nach vorgegebener Agenda ihre jeweiligen Fortschritte berichtet und dass jeweils weitere Vorgehen mit den übrigen Projektpartnern diskutiert und abgestimmt haben.

Im ersten Projektjahr konnten alle Quartalsmeetings durch physische Treffen vor Ort organisiert werden. Erfolgsfaktor war dabei die Einrichtung eines gemeinsamen Repositories für Dokumente, Modelle und weitere Artefakte auf der Basis von GitLab.

Ab dem zweiten Projektjahr waren Corona-bedingt im Wesentlichen nur noch virtuelle Meetings möglich. Hier hat die Nutzung von Microsoft Teams und die Einführung von Miro als Kommunikationsplattform erheblich geholfen.

#### 1.4 Stand der Wissenschaft und Technik

Das Projekt SET Level knüpfte an den aktuellen Stand der Wissenschaft und Technik betreffend Methoden und Werkzeuge des Testens automatisierter Fahrzeugfunktionen an. Insbesondere lieferten die Dokumentationen aus den Projekten PEGASUS (Verweis <https://www.pegasusprojekt.de/de/>), ENABLE S3 (Verweis <https://enable-s3.eu/>) und SmartSE (<https://www.prostep.org/en/projects/smart-systems-engineering/>) dafür eine gute Ausgangsbasis.

Die Hauptideen des Projektes PEGASUS sind in der Definition einer Gesamtmethode zur Bewertung einer hochautomatisierten Fahrfunktion zusammengeführt worden. Diese Gesamtmethode umfasst die Aspekte Anforderungsdefinition, Testableitung, Testvorgehen und Werkzeugunterstützung. Bereits dort wird ein besonderer Schwerpunkt auf die Rolle der Simulation gelegt, und die Thematik der Formulierung von Simulationaufgaben mithilfe der damaligen Versionen der Formate OpenDRIVE und OpenSCENARIO wird eingehend betrachtet. Insbesondere werden dort auch Defizite der Formate für die zweckgemäße Nutzung festgestellt und Vorschläge zur Behebung der Defizite gemacht. Diese Untersuchungen zur Rolle von Simulation in der Absicherung hochautomatisierter Fahrzeugfunktionen und zu den Realisierungsansätzen für eine praktisch nutzbare Anwendung konnten für SET Level aufgegriffen werden und bildeten einen wesentlichen Teil der Basis, auf der das Projekt systematische Lösungen entwickelte.

ENABLE S3 betrachtete Verifikation und Validierung nicht nur im Automobilbereich, sondern auch in Luft- und Raumfahrt, Bahn, Seefahrt sowie Medizin- und Agrartechnik. Das Projekt strebte nach möglichst einheitlichen, übertragbaren Ansätzen, Methoden und Lösungen. So konnten neben spezifischen automobilbezogenen Ergebnissen (etwa zur Simulationsvalidität)

auch generische Beschreibungen in SET Level herangezogen werden, um darauf aufsetzend die automobilspezifischen Ansätze weiterzuentwickeln.

Für SET Level sehr spezifische Konzepte und Ergebnisse konnten aus dem prostep-ivip-Projekt SmartSE herangezogen werden. Die dort entwickelten und erprobten Ansätze zur Austauschbarkeit von Verhaltensmodellen eingebetteter Systeme unter Verwendung des FMI-Standards (Functional Mockup Interface) wurden direkt im Projekt aufgegriffen. Auch der Credible Simulation Process hat seine Wurzel in SmartSE.

Damit sind die wesentlichen Teile der fachlichen Aufsetzbasis von SET Level skizziert. Um den Stand der Wissenschaft und Technik umfassend zu beschreiben, müsste eine große Zahl an Quellen referenziert werden, da die Absicherung automatisierter Fahrfunktionen von vielerlei Seiten bearbeitet wurde und wird. An dieser Stelle sei auf die Dokumentationen der oben besprochenen drei Projekte verwiesen, die mit ihren Dokumentationen bereits einen großen Teil davon abdecken.

### **1.5 Zusammenarbeit mit anderen Stellen**

Fand nicht statt.



## 2 Eingehende Darstellung

### 2.1 Erzielte Ergebnisse

In der Teilprojektstruktur von SET Level hat das ika Personenmonaten in den Teilprojekten 2 (7 PM) und 3 (47 PM) eingeplant. Die erzielten Ergebnisse werden im Folgenden näher erläutert.

#### 2.1.1 Teilprojekt 2: Simulationsbasiertes und virtuelles Entwickeln / Testen

In diesem Teilprojekt hatte das ika einen kleinen Anteil eingeplant. Die meisten Aufwände haben sich auf Reviewtätigkeiten beschränkt. So fand eine aktive Beteiligung an den Diskussionen für eine Szenarienbeschreibungssprache in Workshops und Telefonkonferenzen statt. Das Konsortium strebte eine Weiterentwicklung von OpenSCENARIO an. Dies sollte, wenn möglich, nah an den Arbeiten der ASAM Standardisierungsprojekte OpenSCENARIO Transfer und Konzept geschehen.

Es konnten jedoch auch produktive Ergebnisse in zwei wesentlichen Themenbereichen erzielt werden:

- Simulationsarchitektur hinsichtlich Verkehrsagenten (UAP 2.1.1).
- Schnittstellenerweiterung OSI und OpenSCENARIO zur Ansteuerung von Verkehrsagenten (UAP 2.1.4 und UAP 2.1.5).

Vor dem Hintergrund, dass das ika im Teilprojekt 3 ein Fahrermodell lieferte und das UAP für die Agentenmodelle leitete war es gerade in Bezug auf die Einbindung dieser erforderlich an den Diskussionen bezüglich Schnittstelle und Kommunikation mit dem Szenario zur Laufzeit mitzuwirken und entsprechende Entwürfe und Meinungen einzubringen. Im Vordergrund stand vor allem die Weiterentwicklung des OSI Sensor Model Packaging (OSMP<sup>1</sup>) Protokolls, sowie die Konzipierung der OSI Schnittstellen für Agenten. Hier wurde vor allem auf die Inhalte der Botschaften geachtet. Im Folgenden wird auf die beiden Themen näher eingegangen.

#### Simulationsarchitektur

Gerade zu Projektbeginn wurden die Diskussionen hinsichtlich einer einheitlichen Simulationsarchitektur sehr intensiv geführt. Im Rahmen von SET Level sollte eine Gesamtarchitektur für eine Simulationsplattform konzipiert werden, um automatisierte Fahrzeuge zu testen und zu entwickeln. Das ika hat maßgeblich an den Diskussionen und Ergebnissen hinsichtlich der Einbindung von Agentenmodellen in diese Plattform mitgewirkt. Die Rahmenbedingungen für die Architektur wurden zu Beginn definiert und bestanden aus den Standards Open Simulation Interface<sup>2</sup> (OSI) auf Signalebene und Functional Mockup Interface<sup>3</sup> (FMI) für das Packen und Austauschen von Modellen. Im Folgenden wird auf die konkreteren Überlegungen und Konzepte für die Anbindung von Verkehrsagenten eingegangen, sowie auf die daraus entstandene Simulationsarchitektur.

Zunächst stand die Frage im Raum, welche Botschaften in OSI fehlen, oder erweitert werden müssen, damit Verkehrsagentenmodelle in die Simulation eingebunden werden können. Ein wichtiger Aspekt ist die Repräsentation der Umwelt, so dass sich Agentenmodelle orientieren und bewegen können. Hier waren grundlegende Fragen zu klären, welche Daten im Agentenmodell und welche in der Simulationsumgebung verarbeitet werden sollen. Das eine Extrem lagert die Interpretation der Simulationskarte komplett in den Agenten aus, so dass nur ein Verweis auf beispielweise eine OpenDRIVE Karte geschickt wird und das Modell diese verarbeitet und sich darin lokalisiert. Diese Lösung kostet viel Rechenzeit und findet zudem in jedem Fall in der Simulationsumgebung statt, was dafür spricht dem Agentenmodell eine interpretierte Version der Karte zu übermitteln. Aus Sicht eines Agentenmodells wäre eine Lö-

---

<sup>1</sup> <https://github.com/OpenSimulationInterface/osi-sensor-model-packaging>

<sup>2</sup> <https://github.com/OpenSimulationInterface/open-simulation-interface>

<sup>3</sup> <https://fmi-standard.org/>



sung zum Beispiel ein virtueller Horizont der relativ für jedes genutzte Agentenmodell von der Simulationsumgebung generiert wird. Doch auch hier würden sich Schwierigkeiten ergeben. So ist es sehr beispielsweise unwahrscheinlich, dass ein virtueller Horizont entworfen werden kann, der generisch für verschiedenste Agentenmodelle nutzbar ist, was sich auch in den Diskussionen schnell gezeigt hat. Fußgängermodelle nutzen andere Informationen als Fahrermodelle und bewegen sich nicht zwingend in strukturierten Fahrbahnen, sondern auf Freiflächen. Aber auch die Konzepte für Fahrermodelle vom ika und dem FZI unterschieden sich zu stark, um einen einheitlichen virtuellen Horizont zu entwerfen. Die resultierende Kompromisslösung war, dass die Simulationsumgebungen befähigt werden, die OSI Ground-Truth Botschaft zuverlässig zu befüllen und die Agentenmodelle diese nutzen, um sich in der virtuellen Umgebung zur orientieren und strukturierte Flächen, wie Fahrstreifen, zu nutzen. So entfällt auf Agentenseite das aufwendige Interpretieren einer OpenDRIVE Karte und die Simulationsumgebungen müssen keine spezifischen Berechnungen machen, um für jeden Agenten einen virtuellen Horizont zu erzeugen. Die OSI GroundTruth beschreibt zudem alle Verkehrsteilnehmer und statische Objekte der Simulation und wird in eine OSI SensorView integriert, wenn sie an das Agentenmodell geschickt wird. Die Modelle haben somit alle Informationen, um sich in der Simulation sinnvoll bewegen zu können. Das allein reicht jedoch für die Durchführung eines Szenarios nicht aus, da Agentenmodelle eine Aufgabe benötigen, die im Szenario definiert ist. Hierfür wurde die Botschaft TrafficCommand konzipiert, mit der es möglich ist einem Agentenmodell Befehle, wie eine Zielposition oder eine Trajektorie, zu geben. Außerdem können Wunschgrößen wie die Geschwindigkeit übergeben werden. Der Umfang und die Gültigkeit der Botschaften orientiert sich stark am OpenSCENARIO 1.x<sup>4</sup> Standard (Vgl. nächster Abschnitt). Der resultierende Rückkanal, um der Simulationsumgebung die neue Position des Agenten mitzuteilen wurde durch die Botschaft TrafficUpdate umgesetzt.

Abbildung 6 zeigt die schematische Darstellung der Gesamtarchitektur für eine beispielhafte Simulation. Eine detaillierte Dokumentation kann im Projektbericht gefunden werden (Teilprojekt 2)

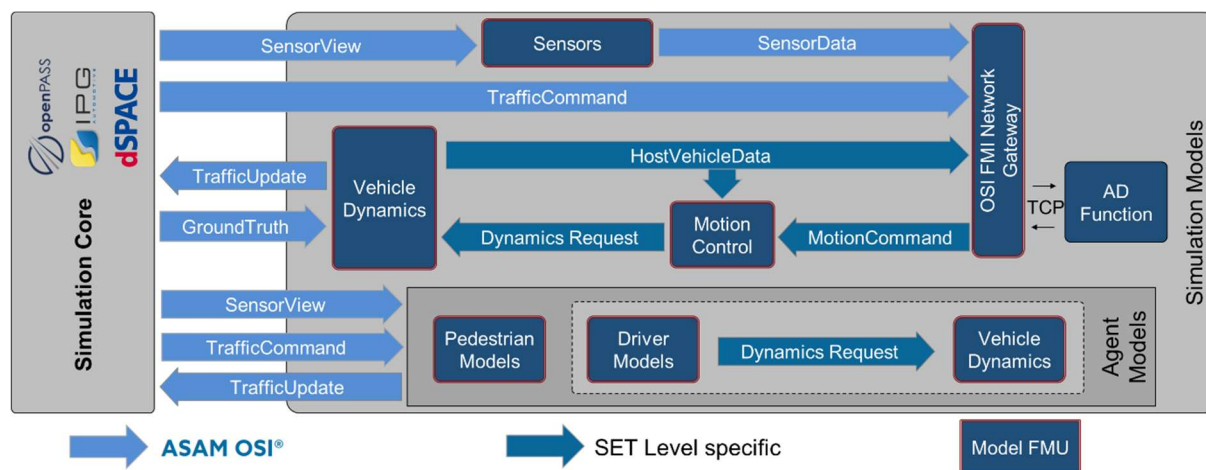


Abbildung 6: Vereinfachte Darstellung der Simulationsarchitektur aus AP2.1 (angewendet für SUC1)

### Schnittstellenerweiterung

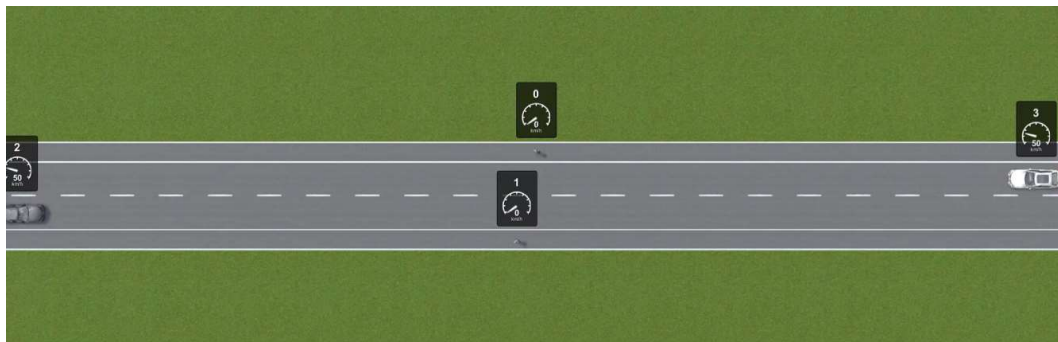
Zur Ansteuerung von Agenten in der konzipierten Simulationsarchitektur wurde die OSI Botschaft TrafficCommand gewählt. Diese bildet einen Großteil der in OpenSCENARIO genutzten PrivateAction, die im Storyboard definiert wird, ab. So kann einem Verkehrsagenten beispielsweise eine Zielposition übergeben werden, oder eine bestimmte Trajektorie, die abgefahren werden soll. Andere Befehle beinhalten die Festlegung einer Wunschgeschwindigkeit,

<sup>4</sup> <https://www.asam.net/standards/detail/openscenario/>

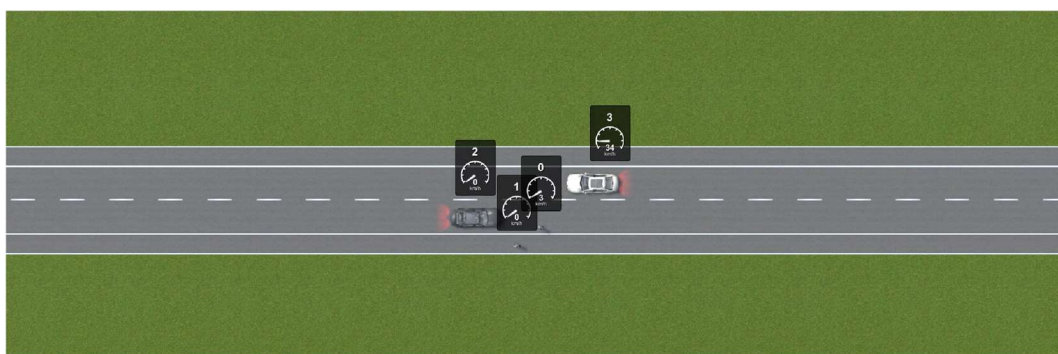
den Abstand zu Zielfahrzeugen, oder den Befehl zum Fahrstreifenwechsel. Eine vollständige Darstellung der nutzbaren Signale ist in der OSI-Spezifikation<sup>5</sup> zu finden.

Eine weitere Möglichkeit zur Beeinflussung von Verkehrsagenten in der Simulation bietet die CustomAction innerhalb des TrafficCommand, bei der ein Freitext im JSON-Format übergeben wird, den das Agentenmodell umsetzen soll. Der Inhalt ist beliebig, in SET Level wurde sich jedoch auf eine Syntax im JSON Format geeinigt, die von der Simulationsplattform erzeugt wird und von den genutzten Verkehrsagenten interpretiert werden kann.

Ein Anwendungsbeispiel dieses Konstrukts wurde zusammen mit Projektpartnern auf dem Halbzeitevent des Projekts gezeigt. Als Simulationsplattform wurde OpenPASS gewählt; die Verkehrsteilnehmer wurden von den Agentenmodellen von BMW (Fußgänger) und dem ika (Fahrer/Fahrzeug) gesteuert. Als Szenario (im OpenSCENARIO Format) wurde eine simple gerade Straße gewählt, auf der in jeder Fahrtrichtung ein Fahrzeug mit 50 km/h fährt. Zwei Fußgänger beabsichtigen die Fahrbahn zu kreuzen. Als Anwendung für den TrafficCommand wurde der Befehl „Ignore\_AllTraffParticipants = true“ gewählt. Die erwartete Beobachtung soll also sein, dass sich einer der Fußgänger nicht regelkonform verhält und auf die Straße läuft, was die Situation für die beiden vom ika Fahrermodell gesteuerten Fahrzeuge potenziell kritisch macht (siehe Abbildung 7 a) für den Initialzustand). Abbildung 7 b) zeigt das Szenario nach ca. 4 Sekunden. Einer der Fußgänger ist, wie erwartet auf die Straße getreten, um diese zu überqueren. Beide Fahrzeuge verzögern, um eine Kollision zu vermeiden. Fahrzeug „2“ muss bis zum Stillstand bremsen, bei Fahrzeug „3“ ist eine Reduktion der Geschwindigkeit ausreichend.



a) Initialisierungszustand des Szenarios mit TrafficCommand. Zwei Fahrzeuge (ID 2, 3) fahren eine gerade Straße entlang und zwei Fußgänger (ID 0, 1) wollen die Straße kreuzen.



b) Kritischer Zustand des getesteten Szenarios nach vier Sekunden. Der Fußgänger, welcher andere Verkehrsteilnehmer ignoriert, läuft auf die Straße. Beide Fahrzeuge verzögern, um eine Kollision zu verhindern. Der zweite Fußgänger verhält sich regelkonform und wartet.

Abbildung 7: Szenario zu Demonstration der CustomAction an zwei Simulationszeitpunkten.

Mit diesem Simulationsaufbau konnte gezeigt werden, dass mit Hilfe der offenen Standards OSI, FMI und OpenSCENARIO Simulationsmodelle verschiedener Herkunft zusammenge-

<sup>5</sup> [https://opensimulationinterface.github.io/open-simulation-interface/structosi3\\_1\\_1TrafficCommand.html](https://opensimulationinterface.github.io/open-simulation-interface/structosi3_1_1TrafficCommand.html)

bracht werden können und gezielt (Fehl-)verhalten erzeugt werden kann, um eine Situation für ein mögliches Testobjekt kritisch zu gestalten.

Als weitere Entwicklung wurde die OSI Botschaft `TrafficCommandUpdate` konzipiert und in den OSI Standard überführt. Diese beschreibt einen Rückkanal vom Verkehrsagenten an die Simulationsumgebung. Verkehrsagentenmodelle können somit mitteilen, wenn sie beispielsweise einen `TrafficCommand` nicht durchführen können. Innerhalb von SET Level wurde diese Botschaft jedoch nicht weiter in einem der Use Cases erprobt.

### **2.1.2 Teilprojekt 3: Modellspezifikation, -Entwicklung und -Validierung**

In diesem Teilprojekt fand der Großteil der geleisteten Projektarbeit statt. Das ika hat im Wesentlichen zwei Modelle konzipiert und implementiert. Zum einen ein Umweltmodell, welches es ermöglicht synthetische Simulationskarten variabel zu erzeugen. Es wurde eine abstrakte logische Streckenbeschreibungssprache entwickelt und ein Tool das daraus Simulationskarten im ASAM OpenDRIVE Format generiert. Auf der anderen Seite wurde ein closed-loop Fahrermodell weiterentwickelt und dazu befähigt in der oben erwähnten Simulationsarchitektur eingesetzt zu werden. Im Folgenden werden beide Modelle näher beschrieben und die grundlegenden Konzepte und Implementierungen dargelegt. Die Arbeitspakete 3.1 bis 3.3 werden jeweils pro Modell gesammelt beschrieben.

#### **2.1.2.1 Umweltmodell**

Die Motivation für ein Umweltmodell, das synthetische Strecken generiert war, ist die Prüfung, welchen Einfluss der Straßenentwurf auf die Leistungsfähigkeit einer automatisierten Fahrfunktion hat. Außerdem wird so ermöglicht eine Vielzahl verschiedener Szenarien in kurzer Zeit zu generieren und automatisiert durchzuführen. Im Rahmen dieser Arbeiten wurden die Veröffentlichungen Nr. 2, 5, 6 und 7 erstellt (Vgl. Abschnitt 2.6.1).

Im weiteren Verlauf ist der Abschnitt wie folgt strukturiert: Zunächst wird ein neues Konzept für die Beschreibung des Straßenverlaufs eingeführt. Dann wird beschrieben, wie diese logische Streckenbeschreibung in einem open-source Tool umgesetzt ist, um eine Vielzahl von Simulationskarten im ASAM OpenDRIVE Standard als Output zu generieren. Abschließend wird auf die Parametrierung und Anwendung des Modells eingegangen.

##### *Konzept*

Ein weitverbreitetes Format für die Darstellung von Straßennetzwerken in der Simulation ist der Standard ASAM OpenDRIVE, welcher die Möglichkeit bietet Straßenverläufe mit mathematisch beschriebenen Geometrien (bspw. Kreisbögen, Spiralen, Polynomen) auszudrücken. Auch in SET Level wurde sich auf ASAM OpenDRIVE als Kartenformat geeinigt. Der Vorteil gegenüber diskreten Punkten besteht darin, dass eine kontinuierliche Beschreibung vorliegt, die dadurch bei geringerer Datenmenge genauer ist. OpenDRIVE wird im XML Format definiert hat jedoch den Nachteil, dass formatbedingt eine sehr redundante Formulierung der Geometrieelemente notwendig ist. Möchte man also beispielsweise die Krümmung einer Straße verändern, zieht das schon bei kleinen Karten Änderungen an hunderten Stellen im Code nach sich.

Damit eine leicht durchzuführende Variation von Straßennetzen umgesetzt werden kann, wurde ein Streckenbeschreibungskonzept entwickelt das modular und weniger redundant als OpenDRIVE ist. Die Umsetzung des Eingangsformats wurde in XML umgesetzt und dessen hierarchische Struktur in Abbildung 8 dargestellt. Auf oberster Kinderelement-Ebene werden Segmente und deren modulare Verknüpfung definiert. Segmente können entweder an deren Enden aneinander geflanscht werden, oder es wird automatisch durch eine mathematische Optimierung eine Verbindungsstraße generiert. Segmente können Kreuzungen, Verbindungsstraßen, oder Kreisverkehre sein. Dieses Konzept ermöglicht den Nutzenden jedes Segment einzeln, ohne Relation zu anderen Elementen, zu erstellen und anschließend zu einem komplexen Netz zusammenzusetzen.

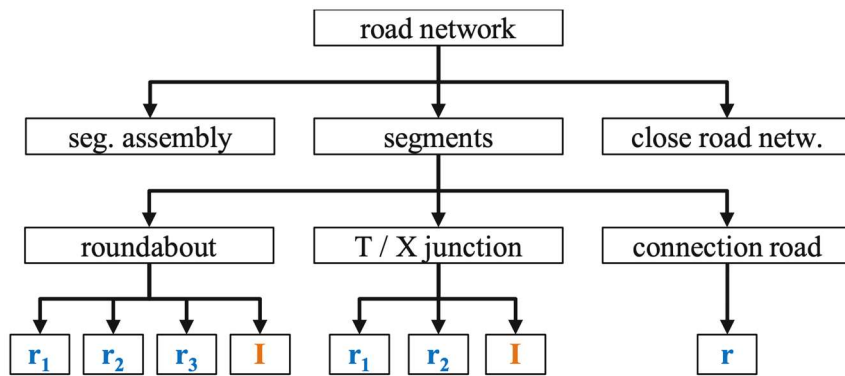


Abbildung 8: Hierarchischer Aufbau des Eingabeformats für die Streckengenerierung.

In Abbildung 9 ist exemplarisch zu sehen, wie eine T-Kreuzung auf Referenzlinieneben angelegt wird. Ohne Angabe der Lage im Raum werden die beiden Straßen der Kreuzung durch Linien, Spiralen und Kreisbögen beschrieben (Abbildung 9 a). Anschließend wird unter Angabe eines Kreuzungswinkels  $\alpha$  und einem Punkt entlang der Referenzlinie der Hauptstraße die T-Kreuzung erzeugt (Abbildung 9 b). In weiteren Schritten werden automatisch Fahrstreifen auf den Zufahrtstraßen generiert, sowie innerhalb der Kreuzung alle nötigen Verbindungsfahrstreifen. Hierfür müssen die ursprünglichen Referenzlinien „aufgeschnitten“ werden. X-Kreuzungen werden analog erzeugt, wobei hier auf beiden kreuzenden Straßen entlang der Referenzlinie Kreuzungspunkte definiert werden.

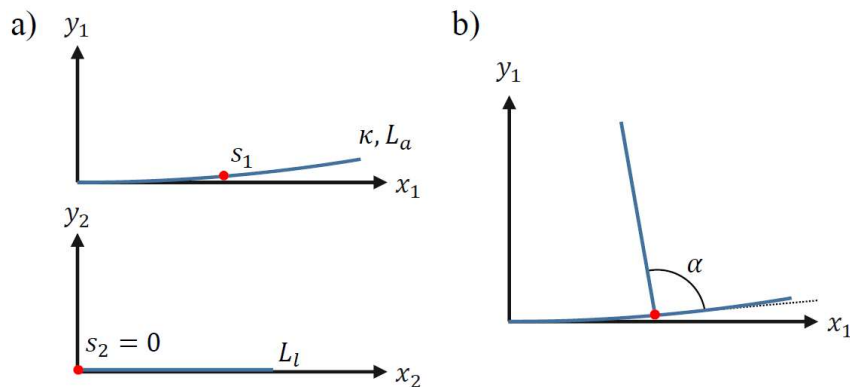


Abbildung 9: Darstellung des Konzepts für die Erstellung einer Kreuzung aus zwei separat definierten Referenzlinien.

Damit aus einzelnen Segmenten ein Straßennetz werden kann, gibt es zwei Möglichkeiten zum Verbinden von Segmenten. Zum einen können Elemente direkt aneinandergesetzt werden. Hier gibt es die Randbedingung, dass kein Krümmungssprung auftreten darf. Diese Einschränkung sorgt einerseits dafür, dass stets ein glatter Verlauf entlang der Referenzlinie gegeben ist und vereinfacht das Verkettens enorm, weil nur die jeweiligen Enden beider Segmente referenziert werden müssen und so keine Freiheitsgrade mehr vorhanden sind (Vgl. Krümmungsband im Straßenbau). Der Verkettungsprozess ist beispielhaft in Abbildung 10 visualisiert. Zunächst wird ein Referenzelement gewählt und im globalen Koordinatensystem platziert und ausgerichtet (Abbildung 10 a). Im weiteren Verlauf kann der eingangs beschriebene Prozess durchgeführt werden und es entsteht ein Straßennetz (Abbildung 10 b).

Auf der anderen Seite bietet das Konzept die Möglichkeit zwischen zwei offenen Enden automatisch eine Verbindungsstraße zu generieren. Dies geschieht mit Hilfe einer numerischen Optimierungsfunktion, die eine Kombination aus Geraden, Spiralen und Kreisbögen errechnet, welche exakt in die definierte Lücke passt.

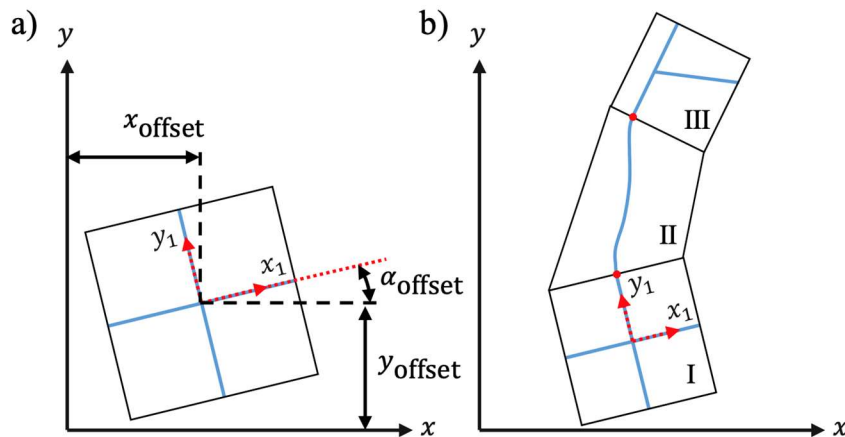


Abbildung 10: Beispielhafte Aneinanderreihung einzelner Segmente.

### Implementierung

Das Konzept wurde im Projektverlauf als C++ Library implementiert, um zu zeigen, dass damit standardisierte OpenDRIVE Karten erzeugt werden können, mit denen man Simulationen zur Absicherung von automatisierten Fahrzeugen durchführen kann. Der Code ist öffentlich verfügbar auf GitHub<sup>6</sup> und stellt eine prototypische Implementierung dar.

Im Wesentlichen werden die im vorangehenden Abschnitt beschriebenen Schritte zur Überführung der vereinfachten Annahmen des Konzepts in eine OpenDRIVE Karte iterativ abgearbeitet. Wenn die Nutzenden keine spezifischen Angaben zu Fahrstreifenbreiten, oder Kreuzungsabmessungen im Eingangsformat machen, werden Standardwerte gewählt, die den Vorgaben des Straßenbaus entsprechen. So ist es einerseits möglich ausgewählte Segmente sehr genau zu spezifizieren und andererseits mit wenig Aufwand Segmente zu spezifizieren, die dann automatisch mit Werten befüllt werden. Innerhalb des Kreuzungsbereichs müssen neue Verbindungsstraßen generiert werden, wofür die ursprünglichen Referenzlinien aufgeschnitten werden. Hierbei wird entweder ein standardisierter Kreuzungsbereich gewählt oder der im Eingangsformat optional definierte Wert für die Aufspanngröße der Kreuzung. Außerdem können automatisch zusätzliche Abbiegefahrstreifen erzeugt werden.

In weiteren Schritten werden in der Implementierung die in Abbildung 10 gezeigten Verkettungen von Segmenten realisiert. Eine vollständige Dokumentation der C++ Bibliothek ist auf GitHub zu finden. Abbildung 11 zeigt ausgewählte Beispiele für generierte Kreuzungen.

### Variations-Erweiterung

Die ursprüngliche Motivation für das entwickelte Streckenbeschreibungskonzept war die Möglichkeit der einfachen Variation von Straßennetzen. Daher wurde in einem weiteren Schritt ein Python Modul entwickelt, welches eben dies ermöglicht. Dazu wurde die Eingangssprache um einen „<vars>“ Tag auf oberster Kind-Ebene erweitert, der es ermöglicht beliebig viele Variablen zu definieren, die dann im weiteren Verlauf der XML Datei anstelle von konkreten Zahlenwerte gesetzt werden können. Folgende Variablentypen können aktuell verwendet werden:

- **Normalverteilung** unter Angabe von Mittelwert und Standardabweichung
- **Gleichverteilung** unter Angabe von Minimal- und Maximalwert
- **Lineare Zusammenhänge** in Abhängigkeit anderer Variablen, z.B.:  $c = 40 * a + b$

<sup>6</sup> <https://github.com/ika-rwth-aachen/RoadGeneration>



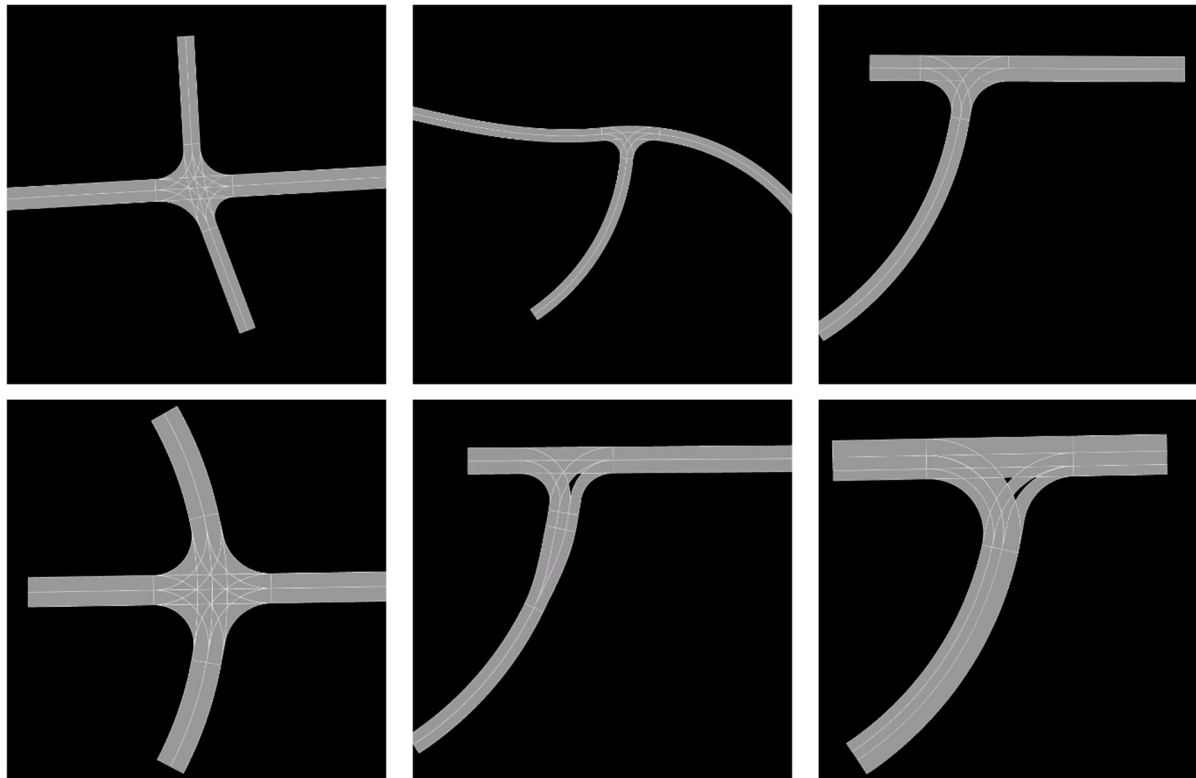


Abbildung 11: Beispiele für generierte Kreuzungen.

Bei der Umsetzung wird zunächst die Eingangsdatei eingelesen, um die Variablen zu erzeugen und entsprechend Ihres Typs zu befüllen. Durch den Benutzer wird festgelegt, wie viele Instanzen der Variablen erzeugt werden sollen. Anschließend werden temporäre Eingangsdateien mit den eingesetzten konkreten Variablensamples, für die im vorherigen Abschnitt beschriebene C++ Bibliothek generiert. Letztere wird in Python eingebunden und so oft aufgerufen, bis die gewünschte Anzahl von Straßennetzen vorliegt.

Auch das Variationstool ist Teil des oben erwähnten GitHub Repository.

### *Parametrierung*

Nach der Implementierung des Konzepts inklusive der Möglichkeit der Variation stellte sich die Frage, welche Parameter und Parameterverteilungen genutzt werden können, um eine automatisierte Fahrfunktion zu testen. Hierfür eignen sich zum einen Vorgaben aus den Richtlinien für die Anlage von Straßen, aus denen eine Vielzahl von zu verbauenden Geometrieparametern extrahiert werden können. Auf der anderen Seite entstand die Idee bestehende Straßennetze systematisch zu analysieren und dadurch reale Parameterverteilungen zu erhalten, die dann für synthetische Kartenerzeugung genutzt werden können. Abbildung 12 zeigt schematisch die konzipierte Verarbeitungskette, welche im folgenden Abschnitt kompakt erläutert wird. Eine detaillierte Beschreibung befindet sich in Veröffentlichung Nr. 7.

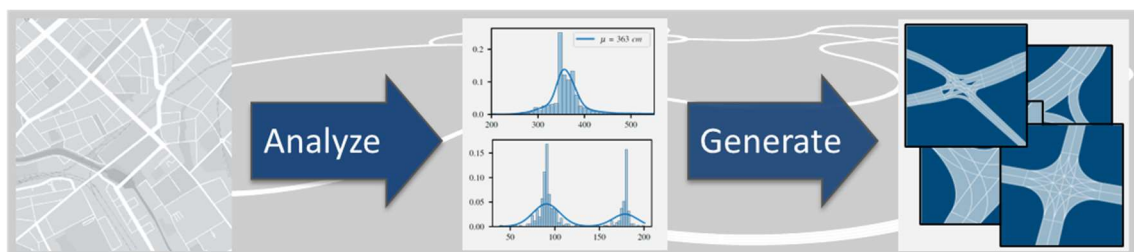


Abbildung 12: Darstellung der Datenverarbeitung für die Streckenerstellung.

Bei der Umsetzung der Analyse werden hochgenaue Karten von HERE<sup>7</sup> benutzt aus denen die Geometrien von realen Kreuzungen extrahiert werden können. HERE Karten sind räumlich in mehrere Kacheln unterteilt und außerdem in einer Ebenenstruktur unterteilt. Die Datenverarbeitung zum Extrahieren von Parametern ist in zwei separate Module gegliedert: Kreuzungsextraktion und Kreuzungsanalyse. Das erste Modul erstellt eine Liste aller Kreuzungen innerhalb vorher ausgewählter Kacheln. Darüber hinaus werden weitere Parameter wie der Abstand zwischen Kreuzungen berechnet. Das zweite Modul führt dann mehrere verarbeitete Kacheln zusammen, kategorisiert die Kreuzungen und generiert neue Parameter, um die Kreuzung genauer zu beschreiben. Ein Beispielparameter hierfür ist die maximale Krümmung der Abbiegespuren. Der Einfachheit halber werden in dieser Arbeit nur Kreuzungen mit drei und vier Armen betrachtet. Das heißt, Autobahnausfahrten oder Kreisverkehre werden in diesem Stadium vernachlässigt. Folglich werden die übrigen Kreuzungen in vier Typen klassifiziert (T-, Y-, X- und unsymmetrische X-Kreuzungen). Ziel ist festzustellen, ob zwei gegenüberliegende Arme einer Kreuzung geometrisch eine durchgehende Straße bilden, was durch einen Winkelbereich definiert wird, der auf 150 bis 210 Grad zwischen zwei Armen festgelegt wird. Mit Hilfe dieses Kriteriums werden die Kreuzungen dann kategorisiert

Anschließend folgt der Prozess der Generalisierung. Ziel ist es, die Kreuzungsarme intern so zu benennen, dass sie für alle Kreuzungen eines bestimmten Typs immer identisch sind. Bei einer T-Kreuzung beispielsweise erhält der Arm den Index 1 und die Glieder, die die Hauptstraße bilden, werden mit dem Index 0 bzw. 2 bezeichnet. Auf diese Weise ist die statistische Analyse reproduzierbar. Nachdem die Topologie der Kreuzungen extrahiert wurde, werden die Fahrspuren der Verbindungen und die Fahrspuren innerhalb der Kreuzung analysiert. Neben Parametern wie Fahrspurbreite und Anzahl der Fahrspuren werden auch die maximale und mittlere Krümmung der Abbiegespuren berechnet, da dieses Maß in HERE Karten nicht enthalten ist. Die Krümmung wird mit Hilfe von diskreten Ableitungen für die Mittellinie jeder Fahrspur berechnet werden, die als Polylinie gespeichert ist.

Abbildung 13 zeigt beispielhaft den Parameter „Kreuzungswinkel“ für T- und X-Kreuzungen. Es ist gut zu sehen, dass bei T-Kreuzungen offenbar Normalverteilungen um 90 und 180 Grad vorliegen, was dem Erwartungsraum entspricht. Die Ergebnisse für X-Kreuzungen sehen ebenfalls plausibel aus mit einem Mittelwert von ca. 90 Grad.

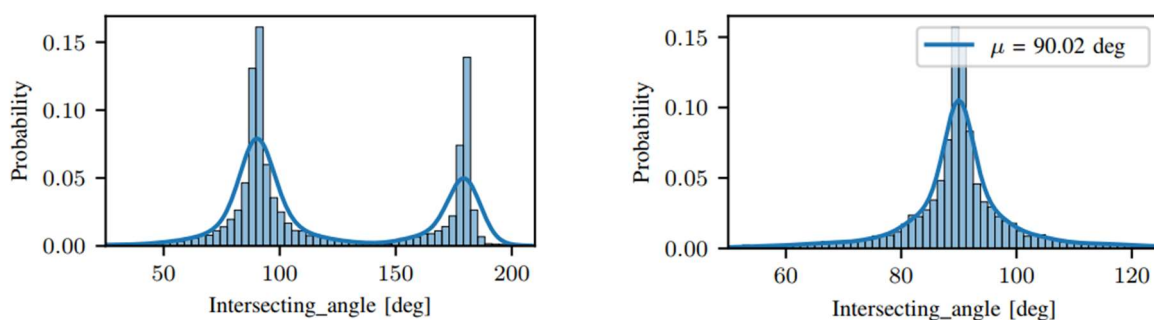


Abbildung 13: Kreuzungswinkel für T- (links) und X-Kreuzungen (rechts) in Berlin.

Die Methodik ist beispielhaft in Berlin auf 49 Kacheln mit einer Fläche von  $183,75 \text{ km}^2$  angewendet worden. Daraus ergeben sich 1379 X-Kreuzungen und 5131 T-Kreuzungen, die extrahiert werden konnten. Die restlichen beiden Typen treten nur vereinzelt auf und sind für die Auswertung vernachlässigt worden.

### Anwendung

Die erarbeitete Werkzeugkette und Parametrisierung wurde im Rahmen von SET Level prototypisch erprobt, um zu zeigen, welchen Einfluss der Straßenentwurf auf die Leistung einer automatisierten Fahrfunktion hat.

<sup>7</sup> <https://www.here.com/platform/HD-live-map>



In Abbildung 14 sind die Verarbeitungs- und Simulationsschritte einer vom ika entwickelten prototypischen Testumgebung zu sehen. Nachdem ein Satz von logischen Streckentemplates erzeugt worden ist, werden diese variiert und in die für die Simulationskomponenten nötigen Kartenformate umgewandelt. In diesem Fall erwartet die prototypische Fahrfunktion des ika eine Lanelet2<sup>8</sup> Karte und die Simulationsumgebung CarMaker benötigt das Format „rd5“ wofür IPG ein Konvertierungstool zur Verfügung stellt. Mit diesen nun vorliegenden Eingangsdaten können CarMaker und die ROS basierte ika Plattform für die AD Funktion initialisiert werden. CarMaker dient hier als Quelle für Umgebungsfahrzeuge und das Bereitstellen der Fahrphysik. Die ika Fahrfunktion verfügt über eine Bewegungsplanung und übergibt deren Ausgang an das CarMaker Fahrzeugmodell. Abschließend werden die durchgeführten Simulationen hinsichtlich acht festgelegter Key-Performance-Indikatoren ausgewertet und die Fahrfunktion bewertet.

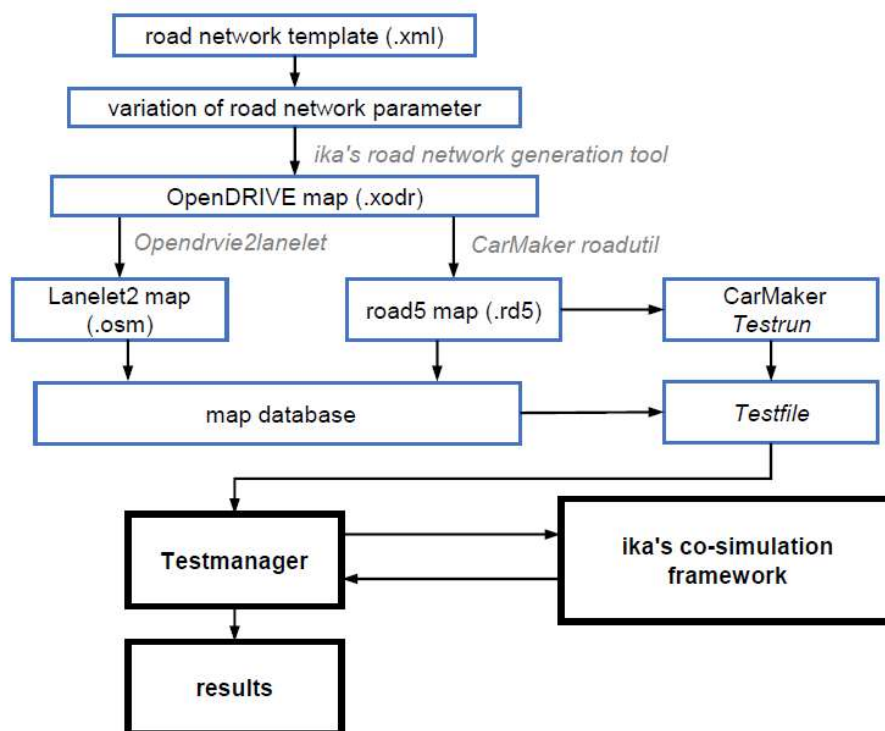


Abbildung 14: Schematischer Aufbau des Testframeworks für die Variation von Straßennetzen in der Simulation

Ziel war hierbei vorrangig eine durchgängige Werkzeugkette aufzubauen, die grundsätzlich in der Lage ist eine Vielzahl von Simulationen durchzuführen und diese zu bewerten. Die Auswertungen des oben beschriebenen Aufbaus haben gezeigt, dass bestimmte Konfigurationen von Parametern zu schlechteren oder besseren Ergebnissen führen. Eine detaillierte Beschreibung und Auswertung wird in Veröffentlichung Nr. 8 zu finden sein.

### 2.1.2.2 Agentenmodell

Agentenmodelle (oder traffic participant models) sind im Kontext von SET Level Modelle von Verkehrsteilnehmern, die in einer szenariobasierten Simulation zusammen mit dem Ego Fahrzeug interagieren. Das ika hat im Verlauf des Projekts ein Agentenmodell entwickelt, das menschliche Fahrer modelliert und closed-loop in der Simulation eingesetzt werden kann, um Verkehr zu erzeugen, oder ein System unter Test (SUT) gezielt herauszufordern. Solche Modelle sind von Vorteil, da das Generieren von Trajektorien für komplexes Verhalten von Agenten in urbanen Szenarien mit mehreren Verkehrsteilnehmern technisch sehr aufwendig ist.

<sup>8</sup> <https://github.com/fzi-forschungszentrum-informatik/Lanelet2>

Modelle, die reaktiv auf verschiedenes Verhalten des SUT reagieren erleichtern die Beschreibung und Durchführung von Szenarien signifikant.

Wie in Kap. 2.1.1 beschrieben muss die Schnittstelle zwischen der Ausführung von Szenarien und der Ausführung von Verkehrsteilnehmermodellen gut konzipiert sein, um sicherzustellen, dass beide für eine bestimmte beabsichtigte Simulationsaufgabe kompatibel bleiben. Daher wurde auf Basis der Schnittstellendefinition von TP2 ein Modell entwickelt, das diesen Anforderungen genügt und in mehrere Simulationsplattformen integriert werden kann.

Im Folgenden wird das Modell auf verschiedenen Ebenen näher beschrieben. Zunächst wird auf die Fähigkeiten und Teile der Implementierung eingegangen. Anschließend wird die Architektur näher beleuchtet, sowie die Modellintegration in verschiedene Simulationsumgebungen erläutert. Abschließend wird erläutert, wie das Modell systematisch getestet wird, um dessen Funktionalität stets sicherzustellen.

### *Modellfähigkeiten*

Aus den Diskussionen und Abstimmungen für den SUC1 haben sich im Laufe des Projekts folgende Anforderungen ergeben, die das ika-Fahrermodell beherrschen muss:

- Von einem Startpunkt zu einem Zielpunkt auf der Karte fahren, unter folgenden Bedingungen:
  - Das Fahrzeug wird durch Simulationsumgebung in die Karte gesetzt.
  - Wahl der zu befahrenden Fahrstreifen bis zum Ziel erfolgt im Agentenmodell.
  - Ziel muss erreichbar sein (ansonsten wird nur bis zum Ende der aktuellen OSI-Lane gefahren).
- Lateral an den Fahrstreifen der Karte orientieren;
- Definierte Geschwindigkeit einregeln und halten;
- Folgen (konstante THW halten);
- Enge Kurvenfahrt;
- Stoppen, halten, wieder anfahren;
- Vorfahrts- und Ampelverhalten modellieren. Hierfür ist eine Prädiktion der Bewegung der anderen Verkehrsteilnehmer sowie die Interpretation von Ampelschaltungen nötig.

Die Modellierung wird in etwa der Fahrermodellierung von (Donges, 1982) nachempfunden, der die Fahraufgabe in Navigations-, Bahnführungs- und Stabilisierungsebene einteilt. Die Navigation findet in einem vorgelagerten Modul statt, das die richtigen Straßenzüge bis zum Ziel auswählt. Falls Fahrstreifenwechsel nötig sind, werden diese in einem Entscheidungsmodul an geeigneter Stelle durchgeführt. Auf der Bahnführungsebene stehen verschiedene Basismanöver zur Verfügung, die durch Führungsgrößen gekennzeichnet sind. Diese Basismanöver können dann kombiniert werden, damit die Fahraufgabe gelöst werden kann. Abbildung 15 stellt die Grundfähigkeiten des Fahrerhaltensmodells in einem Zustandsdiagramm dar. Die Kästen repräsentieren stabile Zustände, in denen eine Führungsgröße in der Closed-Loop Simulation durch das Modell konstant gehalten wird. Das ist im Fall von „Keeping Speed“ die Wunsch-, oder Richtgeschwindigkeit, oder bei „Following“ eine parametrierbare Folgezeitlücke zum Vorderfahrzeug. Beim „Lane Keeping“ wird der Versatz zur Mitte des Fahrkorridors (meist der Fahrstreifen) eingeregelt. Die Pfeile zwischen den Kästen beschreiben die Veränderung einer Führungsgröße, welche im Fall von durchgezogenen Linien durch eine Fähigkeit modelliert ist. Hier sei als Beispiel die Veränderung in der Wunschgeschwindigkeit durch ein Schild am Straßenrand genannt. Es wird ein glatter Geschwindigkeitsübergang realisiert, ohne dass Sprünge in der Beschleunigung auftreten. Gestrichelte Pfeile sind implizite Veränderungen der Führungsgröße, z.B. wenn das Vorderfahrzeug stehen bleibt und das Modell in den Zustand „Standing“ wechselt. Die Fähigkeiten sind in laterale und longitudinale Operationen aufgeteilt. Die Kontrollgrößen sind jeweils die Wunschbeschleunigung, sowie die Wunschkrümmung. Diese beiden Größen werden dann in der Stabilisierungsebene integriert, oder durch Dynamikmodelle kontrolliert.

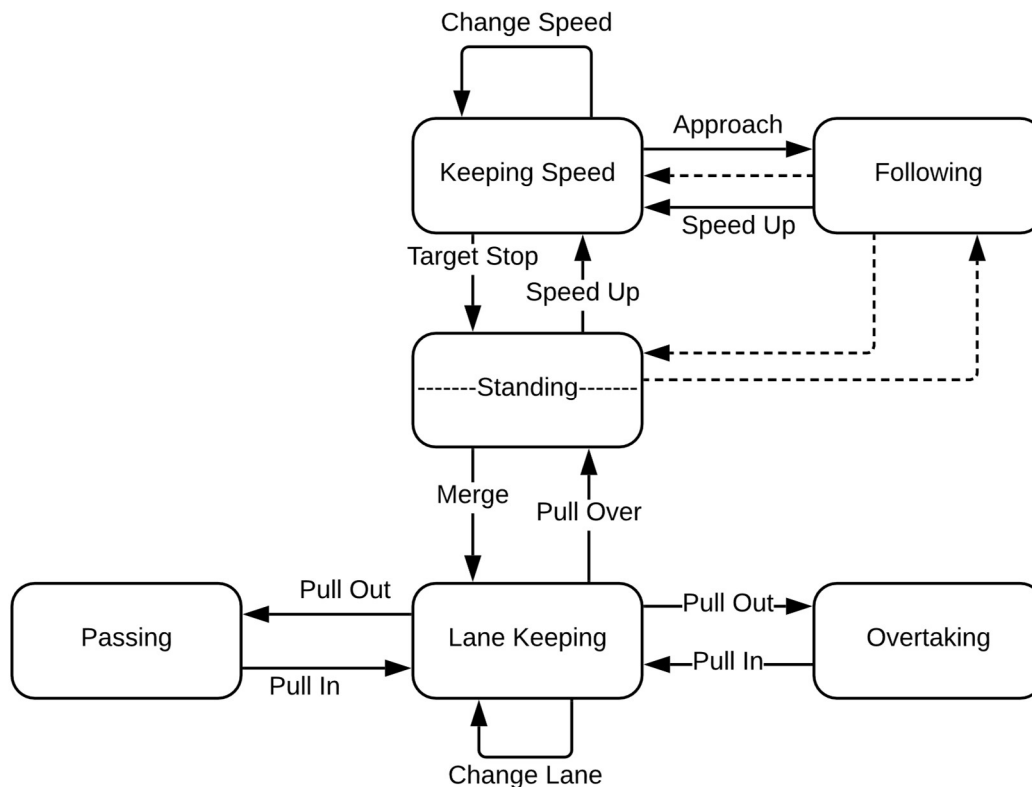


Abbildung 15: Darstellung der Basismanöver des Modells als Zustandsdiagramm.

Um die Architektur zu erläutern, wird das Beispiel eines Fahrstreifenwechsels angeführt: Die Funktionalität erstreckt sich über alle Komponenten des Modells. Sie erfordert interpretierte und konsolidierte Informationen (Interpretation), um eine Entscheidung zum Fahrstreifenwechsel durchzuführen (diskrete Entscheidung). Der Prozess des Fahrstreifenwechsels wird in die Komponenten *bewusste* und *unterbewusste* Führung unterteilt. Die berechneten Antizipations- und Kontrollwerte werden schließlich in der Komponente *Stabilisierung* umgesetzt.

#### Modellarchitektur und -implementierung

Das Agentenmodell steht als FMU zur Verfügung und kann so in die konzipierte SET Level Simulationsarchitektur integriert werden. Die FMU ist in Abbildung 16 als weißer Kasten dargestellt und bekommt als Eingangssignale, die in Kap. 2.1.1 beschrieben, OSI-Botschaften. Es folgt eine Datenverarbeitung der OSI GroundTruth im Modul „OSI-Adapter“, welches eine abstrakte interne Schnittstelle für das Fahrerverhaltensmodell befüllt. Zum einen werden alle umgebenden Objekte in das Fahrzeugkoordinatensystem transformiert, sowie eine Routenplanung durchgeführt. Das Ziel auf der Karte wird über die TrafficCommand Schnittstelle bereitgestellt und eine Suche, die die Relationen zwischen vorhandenen Fahrstreifen analysiert und auf dem Dijkstra Algorithmus basiert, liefert alle abzufahrenden Fahrstreifen. Daraus wird ein Fahrschlauch erzeugt, worin im Verhaltensmodell laterale und longitudinale Bewegung geplant und durchgeführt werden. Weitere Aufgaben des „OSI-Adapters“ sind die Interpretation von Vorfahrtsregeln und Ampelschaltungen entlang der Route, damit das Fahrerverhaltensmodell Entscheidungen hinsichtlich longitudinaler Bewegungsplanung fällen kann. Detaillierte Ausführungen bezüglich der Modellierung im Fahrerverhaltensmodell sind in Veröffentlichung Nr. 4 zu finden.

Den Ausgang des Verhaltensmodells bilden die beiden Wunschgrößen für Längsbeschleunigung und Krümmung. Diese können zum einen mittels der SET Level internen Schnittstelle DynamicsRequest direkt an den Simulationskern übermittelt werden, wo die Umsetzung in eine Bewegung durch ein beliebiges Dynamikmodell geschehen kann. Zum anderen besitzt die FMU des Agentenmodells ein Dynamikmodell, welches die beiden Wunschgrößen in Pedal- und Lenkradwinkelwerte übersetzt, die dann durch ein vereinfachtes Einspurmodell in

eine Bewegung umgesetzt werden. Das Ergebnis ist die neue Pose des Agenten, die mit der Botschaft TrafficUpdate zurück an den Simulationskern gegeben wird.

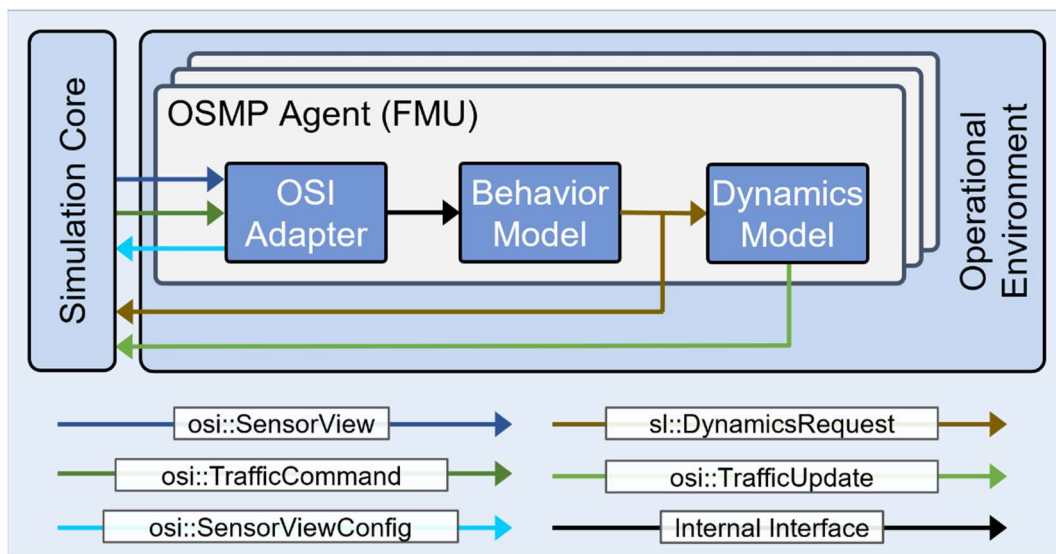


Abbildung 16: Schematischer Aufbau der Agentenmodellbindung.

Die Implementierung wurde in der Programmiersprache C++ durchgeführt. Abbildung 17 zeigt die Codestruktur des Agentenmodells. Auf oberster Ebene liegen Verzeichnisse für Header- und Sourcecode-Dateien, die für das Packen der FMU nötig sind. Die Abhängigkeiten der FMU sind im Ordner „lib“ hinterlegt, wo der OSI-Wrapper als Library gebaut wird. Das gesamte Projekt wird mit CMake konfiguriert, welches ein plattformunabhängiges Programmierwerkzeug für die Entwicklung und Erstellung von Software ist.

- **ika-agent** – Hauptverzeichnis (OSMP Agent Model)
  - **include** – Header-Dateien für OSMP und FMU
  - **lib** – Verzeichnis für den OSI-Adapter, der OSI interpretiert und dem Agentenmodell alle nötigen Informationen liefert
    - **osiWrapper** – Neue Bibliothek (s.o.)
      - **lib** – Abhängigkeiten des OSI-Adapters (Verhaltensmodell, OSI, json-parser)
      - **src** – Quellcode (inkl. Agentenmodell und Regler)
      - **CMakeLists.txt** – Build-Datei für den OSI-Adapter
  - **scripts** – Skripte, um verschiedene Debug-Informationen zu plotten
  - **src** – Quellcode für die FMU „Hülle“ (OSMP)
  - **CMakeLists.txt** – Build-Datei für die ika-agent FMU

Abbildung 17: Vereinfacht dargestellte Verzeichnisstruktur des ika Agentenmodells.

Eines der Ziele von SET Level war die Bereitstellung prototypischer Modelle für die simulative Absicherung von automatisierten Fahrfunktionen. Im Laufe des Projekts wurde daher ein öffentlich zugänglicher GitLab-Bereich geschaffen<sup>9</sup>, wo Projektergebnisse verfügbar gemacht werden sollen. Das ika hat den aktuellen Stand des Agentenmodells zur Verfügung gestellt:

[https://gitlab.setlevel.de/open/models/traffic-participants/driver\\_model](https://gitlab.setlevel.de/open/models/traffic-participants/driver_model)

### Modellintegration und -erprobung

<sup>9</sup> <https://gitlab.setlevel.de/open>

Das ika Agentenmodell, basierend auf der zuvor vorgestellten OSMP Simulationsarchitektur, wurde in drei Simulationswerkzeuge integriert: OpenPASS, CARLA und CarMaker. Im Laufe des Projekts wurden die Tools befähigt OSI und FMI bereitzustellen, um eine unkomplizierte Nutzung der Architektur zu ermöglichen. Die Projektpartner BMW und IPG haben ihre Tools OpenPASS und CarMaker entsprechend erweitert und in eng abgestimmten Iterationsschleifen ist die Modellintegration gelungen. Für das Simulationstool CARLA hat der Projektpartner DLR eine prototypische OSI und FMI Anbindung entwickelt, die nach leichten Anpassungen durch das ika auch für das Agentenmodell genutzt werden konnte. Abbildung 18 zeigt beispielhaft die Integration von mehr als 20 Instanzen des ika Agentenmodells in OpenPASS im Szenario des SUC3 in dessen größter Ausbaustufe. Als weiteres Beispiel ist in Abbildung 19 die Simulationsumgebung CARLA zu sehen, in der ebenfalls alle Fahrzeuge durch das ika Agentenmodell gesteuert sind.

Darüber hinaus bieten alle genannten Werkzeuge Unterstützung für ASAM OpenSCENARIO und ASAM OpenDRIVE, so dass das Modell Szenario-basiert in allen Umgebungen verwendet und getestet werden kann.



Abbildung 18: Komplexe Kreuzung, die für Meilenstein 3 im Simulation Use Case 1 verwendet wird.



Abbildung 19: Kreuzungsszenario im Simulationstool CARLA mit Fahrzeugen gesteuert vom ika Agentenmodell.

Nachdem das Modell in die Simulation integriert werden konnte, wurden simulative Tests durchgeführt. Dazu gehörte das modulare Testen von genau definierten Szenarien mit zuvor definierten Metriken. Vier Beispiele sind in Abbildung 20 zu sehen. Zum einen die drei wichtigsten Fähigkeiten des Agentenmodells: Annähern und Folgen, Geschwindigkeitsbegrenzungen einhalten und taktische Spurwechsel. Außerdem ist in Abbildung 20 d) zu sehen, wie das Modell die Geschwindigkeit prädiktiv anpasst, wenn die Querbeschleunigung einen parametrisierten Wert zu überschreiten droht.

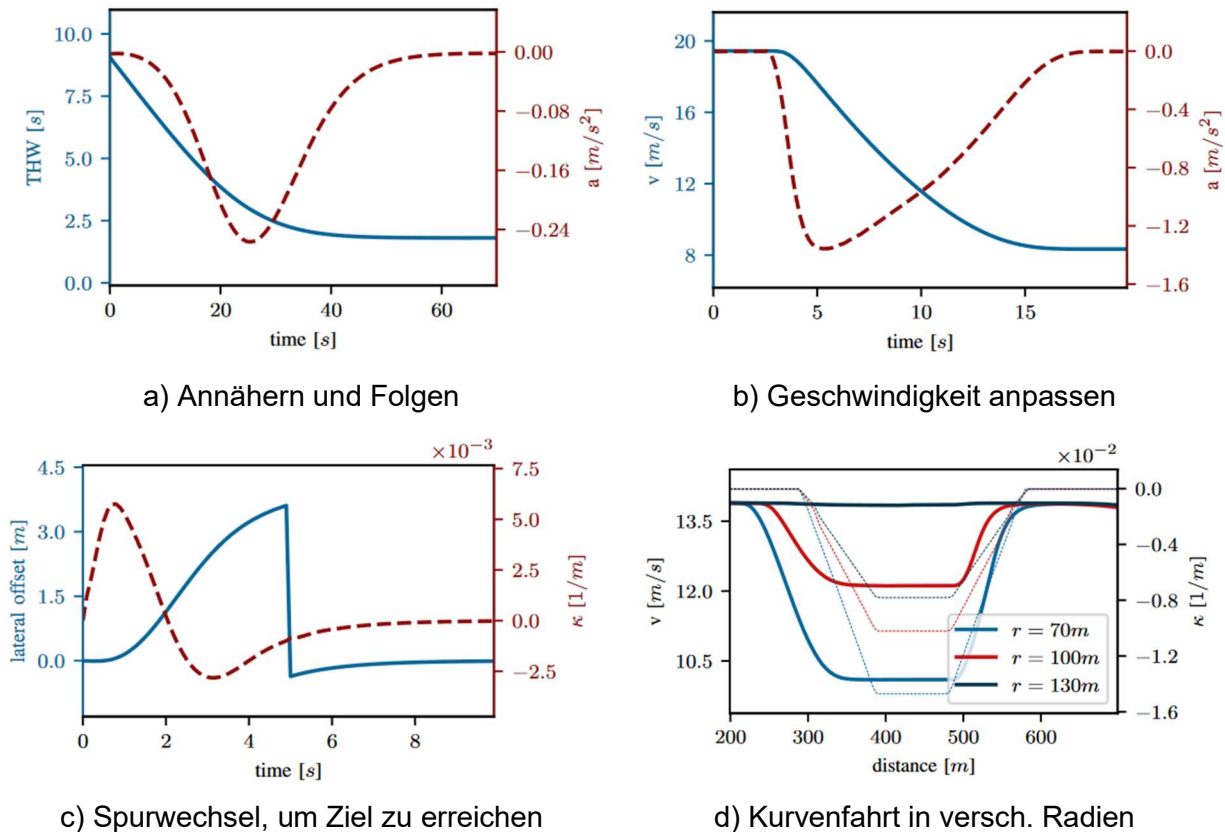


Abbildung 20: Beispieltestauswertungen des closed-loop Agentenmodells.

Mit Hilfe eines End-to-End-Testing-Frameworks und OpenPASS wurde eine automatisierte Testpipeline implementiert, um Tests aus einem Katalog in Form von Continuous-Integration (CI) Jobs durchzuführen. Dabei handelt es sich hauptsächlich um grundlegende Fähigkeitstests für das Agentenmodell. So ermöglichen die CI-Ergebnisse ein direktes Feedback im DevOps-Prozess während der Entwicklung des Agentenmodells und gewährleisten eine konsistente Verifizierung der Kernfähigkeiten des Modells.



## GNSS-Modell

Im Zuge von SET Level wurde vom ika ein Modell eines globalen Navigationssatellitensystem (GNSS) konzipiert und anhand eines vereinfachten GPS Modells prototypisch implementiert. Nach Rücksprache mit den Projektpartnern FZI und Bosch wurde das in Abbildung 21 dargestellte Schema für das Modell konzipiert und GPS als GNSS Technologie gewählt.

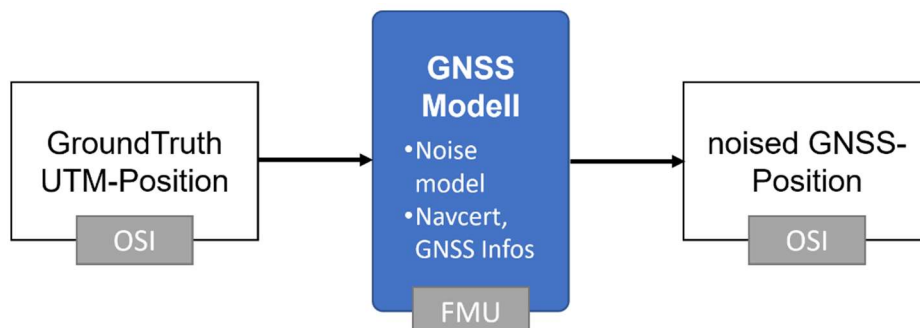


Abbildung 21: Schematische Darstellung des GPS-Modell-Konzepts. Input und Output sollen im OSI Standard vorliegen; Das Modell wird, ähnlich wie Agentenmodelle als FMU mit OSMP gepackt.

Als Eingang dient die GroundTruth (möglicherweise auch in eine SensorView eingebettet), in der die Aufstandspunkte als kartesische Koordinaten hinterlegt sind. Eine als mit OSMP gepackte FMU ermittelt daraus eine GPS-Position, welche in einem weiteren Schritt mit realistischen Werten verrauscht wird. Auf der Ausgangsseite muss eine neue OSI Botschaft konzipiert werden, die eine potenzielle automatisierte Fahrfunktion als Eingang für die Lokalisierung nutzen kann. Es wurde prototypisch die Botschaft `GnnsLocalization` erstellt, die im Projekt konsolidiert werden sollte. Ein weiterer offener Punkt ist zum Beispiel eine bisher fehlende GPS-Referenz in der GroundTruth, von der ausgehend die in kartesischen Koordinaten gegebenen Aufstandspunkte von UTM zurück in GPS-Koordinaten gerechnet werden können. Aktuell muss dieser Wert als Parameter im GPS-Modell hinterlegt werden.

Im Laufe des Projekts und vor allem in den SUCs (Simulation Use Cases) hat sich jedoch keine konkrete Anforderung ergeben das Modell einzusetzen. Daher geschah keine Erprobung des implementierten Modells.

## 2.2 Zahlenmäßiger Nachweis

Der zahlenmäßige Nachweis wurde bereits im Dezember 2022 eingereicht.

## 2.3 Notwendigkeit der Förderung

Die Förderung ermöglichte eine umfassende, technisch tiefe Zusammenarbeit des Konsortiums, in dem alle relevanten Rollen von Akteuren im Sektor vertreten waren (OEM, Zulieferer, Technologieentwickler, Wissenschaft und Forschung). So konnten Fortschritte auf dem Feld der Simulation für den Test und die Absicherung automatisierter Straßenfahrzeuge erzielt werden, die eine breite Akzeptanz zu finden versprechen.

Insbesondere im Bereich der Standardisierung der Formate für Testbeschreibung, die simulationsinternen Schnittstellen und der darüber ausgetauschten Inhalte wurden in Abstimmung mit allen Beteiligten große Fortschritte erzielt. Auch die Nutzbarkeit der Setzungen für die praktische Anwendung konnte in gemeinsam umgesetzten Demonstrationen gezeigt werden. So hat der Ansatz, die akteurübergreifende Zusammenarbeit in einem Verbundprojekt zu fördern, wesentlich dazu beigetragen, im vorwettbewerblichen Bereich Übereinkünfte zu treffen, die zukünftig in wirtschaftliche Umsetzungen einfließen und fortschrittliche Technologien auf dem aktuellen Stand der Wissenschaft und Technik für alle Akteure verfügbar machen.



Durch die Förderung des Projekts SET Level wurde die Qualität der erreichten Ergebnisse insbesondere im Bereich der standardisierten Datenformate und der eingesetzten Tools durch die Abstimmung und Zusammenarbeit im Gegenstandsbereich der beteiligten Firmen und Institutionen in erheblichem Maße gesteigert. Ebenso wurden durch die Zusammenarbeit im Projekt umfangreichere Projektergebnisse erzielt, da die Zuwendung eine umfassendere Betrachtung und erweiterte Evaluation der Ergebnisse ermöglicht hat.

## **2.4 Nutzen und Verwertbarkeit**

Das Agentenmodell ist für das ika weiterhin im Themenfeld der simulativen Absicherung nutzbar. In den EU Förderprojekten SUNRISE und V4Safety besteht die Möglichkeit das entwickelte Modell in Absicherungsframeworks zu nutzen und weiterzuentwickeln. Ebenso ist eine Anbindung an den hochdynamischen Fahrsimulator zur Erzeugung von Umweltverkehr angedacht.

Ein weiterer wesentlicher Aspekt ist das „Nachleben“ von Forschungsprojekten. Die [openMSL](#) (Open Source Model & Simulation Library) des Vereins Automotive Solution Center for Simulation („asc(s)“) verfolgt dieses Ziel. Die Library soll eine sichtbare Plattform werden, auf der Modelle und Methoden aus Forschungsprojekten weiterentwickelt und im Idealfall nutzbar sind. Zum einen besteht hier für das ika die Möglichkeit der Mitgestaltung von Standardanwendungen und -Schnittstellen im Bereich der Agentenmodellierung. Die Ergebnisse und Erkenntnisse aus den Teilprojekten 2 und 3 von SET Level können bei diesem Projekt wertvoll eingesetzt werden. Außerdem können, ausgehend vom Agentenmodell, Fahrerverhaltens- und Fahrerleistungsmodelle implementiert werden. Die Arbeit aus SET Level und der Nutzung, sowie Weiterentwicklung von CARLA als Simulationsumgebung lassen sich ebenfalls einbringen. Es gibt erste Überlegungen, wie beispielsweise das Agentenmodell genutzt werden kann, um dessen Leistung in den ALKS Szenarien zu prüfen. Aktuell findet eine Prüfung statt, inwieweit sich das ika bei den Aktivitäten der openMSL einbringen kann.

Auch die Arbeiten im Bereich der Streckengenerierung sind weiterhin nutzbar und bieten Möglichkeiten für fortführende Arbeiten. Der Code steht, wie oben beschrieben, öffentlich zugänglich zur Verfügung und kann von jedem genutzt werden, um einfach und effektiv OpenDRIVE Karten zu generieren.

## **2.5 Bekannt gewordener Fortschritt**

Während des SET Level-Projektes sind keine Fortschritte anderer Vorhaben auf dem Gebiet von SET Level bekannt geworden, die einen signifikanten Einfluss auf SET Level hatten.

## 2.6 Veröffentlichungen

### 2.6.1 Erfolgte Veröffentlichungen

Im Zuge der Ergebnisverbreitung erfolgte die Präsentation der SET Level-Umfänge, wie in Tabelle 1 aufgelistet.

Tabelle 1: Erfolgte Veröffentlichungen des ika in SET Level (kursiv: geplant)

Nr.	Datum	Veranstaltung, Ort	Titel	Autoren	Anmerkung
1.	19.04.2020	GitHub.com, <i>Repository-Veröffentlichung</i>	A responsive driver model for traffic simulations to create closed-loop microscopic traffic scenarios	Jens Klimke (E.Go Moove GmbH), Daniel Becker (ika)	<a href="https://github.com/ika-rwth-aachen/SimDriver">https://github.com/ika-rwth-aachen/SimDriver</a>
2.	29.07.2020	Automated Vehicles Symposium 2020, virtual event	ika's generic Driver Model for AD Safety Reference	Daniel Becker (ika), Jens Klimke (E.Go Moove GmbH), Lutz Eckstein (ika)	Vortrag
3.	23.09.2020	The 23rd IEEE International Conference on Intelligent Transportation (ITSC), Rhodes, Greece (virtual conference)	Generation of Complex Road Networks Using a Simplified Logical Description for the Validation of Automated Vehicles	Daniel Becker, Fabian Ruß, Christian Geller, Lutz Eckstein (alle ika)	Paper +Vortrag
4.	07.10.2020	29th Aachen Colloquium 2020, Aachen	System Design of an Agent Model for the Closed-Loop Simulation of Relevant Scenarios in the Development of ADS	Jens Klimke (E.Go Moove GmbH), Daniel Becker, Lutz Eckstein (beide ika)	Paper +Vortrag
5.	05.2021	ATZechnik, <i>Printausgabe</i>	Agentenmodell für die Closed-Loop Simulation von Verkehrsszenarien	Daniel Becker (ika), Jens Klimke (Cariad SE), Lutz Eckstein (ika)	Zeitschriftenartikel
6.	17.09.2021	IDEA League summer school 2021, Chalmers University	Parametrization of Human Driver Models with Naturalistic Drone Data	Daniel Becker	Gastvorlesung
7.	14.11.2022	GitHub.com, <i>Repository-Veröffentlichung</i>	Road Generation Tool for Basic OpenDRIVE Road Networks	Daniel Becker (ika), Christian Geller (ika)	<a href="https://github.com/ika-rwth-aachen/RoadGeneration">https://github.com/ika-rwth-aachen/RoadGeneration</a>
8.	17.11.2022	International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICEC-CME), Malediven (virtual conference)	Road Network Variation Based on HD Map Analysis for the Simulative Safety Assurance of Automated Vehicles	Daniel Becker, Christian Geller, Lutz Eckstein (alle ika)	Arbeit in SET Level, Finanzierung über VVMethoden (Projektende), Paper +Vortrag
9.	22.05.2023	15th ITS European Congress, Lisbon, Portugal	Simulative Performance Analysis of an AD Function with Road Network Variation	Daniel Becker, Guido Küppers, Lutz Eckstein (alle ika)	Arbeit in SET Level, Finanzierung über VVMethoden (Projektende)
10.	01.10.2023	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA	Driver Model Integration for an Open-Source based Simulation Architecture using Scenario-Based Testing	Christian Geller, Daniel Becker, Jobst Beckmann, Lutz Eckstein (alle ika)	Review Prozess, Arbeiten in SET Level, Finanzierung über VVMethoden (Projektende)

### 3 Abbildungsverzeichnis

Abbildung 1: Projekte der PEGASUS-Familie.....	3
Abbildung 2: Zusammensetzung der Partner von SET Level.....	4
Abbildung 3: Schnelle Iterationen in Projektmeilensteinen.....	4
Abbildung 4: Aufbau der Teilprojekte von SET Level.....	5
Abbildung 5: Zeitplanung mit Meilenstein- und Abschlussberichten.....	6
Abbildung 6: Vereinfachte Darstellung der Simulationsarchitektur aus AP2.1 (angewendet für SUC1) .....	9
Abbildung 7: Szenario zu Demonstration der CustomAction an zwei Simulationszeitpunkten. ....	10
Abbildung 8: Hierarchischer Aufbau des Eingabeformats für die Streckengenerierung. ....	12
Abbildung 9: Darstellung des Konzepts für die Erstellung einer Kreuzung aus zwei separat definierten Referenzlinien.....	12
Abbildung 10: Beispielhafte Aneinanderreihung einzelner Segmente.....	13
Abbildung 11: Beispiele für generierte Kreuzungen.....	14
Abbildung 12: Darstellung der Datenverarbeitung für die Streckenerstellung. ....	14
Abbildung 13: Kreuzungswinkel für T- (links) und X-Kreuzungen (rechts) in Berlin.....	15
Abbildung 14: Schematischer Aufbau des Testframeworks für die Variation von Straßennetzen in der Simulation .....	16
Abbildung 15: Darstellung der Basismanöver des Modells als Zustandsdiagramm.....	18
Abbildung 16: Schematischer Aufbau Agentenmodellanbindung.....	19
Abbildung 17: Vereinfacht dargestellte Verzeichnisstruktur des ika Agentenmodells.....	19
Abbildung 18: Komplexe Kreuzung, die für Meilenstein 3 im Simulation Use Case 1 verwendet wird.....	20
Abbildung 19: Kreuzungsszenario im Simulationstool CARLA mit Fahrzeugen gesteuert vom ika Agentenmodell.....	20
Abbildung 20: Beispieltetauswertungen des closed-loop Agentenmodells. ....	21
Abbildung 21: Schematische Darstellung des GPS-Modell-Konzepts. Input und Output sollen im OSI Standard vorliegen; Das Modell wird, ähnlich wie Agentenmodelle als FMU mit OSMP gepackt. ....	22

### 4 Tabellenverzeichnis

Tabelle 1: Erfolgte Veröffentlichungen des ika in SET Level (kursiv: geplant).....	24
--	----

## **5 Literaturverzeichnis**

Donges, E. (1982). *Aspekte der Aktiven Sicherheit bei der Führung von Personenkraftwagen*.  
Automobil-Industrie 27, S. 183-190.

## Berichtsblatt

1. ISBN oder ISSN -	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht	
3. Titel Schlussbericht		
4. Autor(en) [Name(n), Vorname(n)] Daniel Becker	5. Abschlussdatum des Vorhabens 31.10.2022	
	6. Veröffentlichungsdatum 30.4.2023	
	7. Form der Publikation Dokument	
8. Durchführende Institution(en) (Name, Adresse) Rheinisch-Westfälische Technische Hochschule Aachen Fakultät 4 – Maschinenwesen Institut für Kraftfahrzeuge Steinbachstr. 7 52074 Aachen	9. Ber. Nr. Durchführende Institution -	
	10. Förderkennzeichen 19A19004C	
	11. Seitenzahl 26	
12. Fördernde Institution (Name, Adresse)  Bundesministerium für Wirtschaft und Klimaschutz (BMWK) 53107 Bonn	13. Literaturangaben 1	
	14. Tabellen 1	
	15. Abbildungen 21	
16. Zusätzliche Angaben		
17. Vorgelegt bei (Titel, Ort, Datum) -		
18. Kurzfassung Schlussbericht der am ika durchgeführten Arbeiten im Förderprojekt SET Level. Die Arbeiten an den Teilprojekten 2 und 3 werden zusammengefasst und dokumentiert. Folgende Aspekte erwarten den Leser: <ul style="list-style-type: none"> <li>- Konzeption einer Simulationarchitektur für das szenarienbasierte, simulative Testen</li> <li>- Agentenmodell für den Umgebungsverkehr in der Simulation</li> <li>- Streckenmodell zur effizienten Erzeugung von Simulationskarten</li> </ul>		
19. Schlagwörter Schlussbericht ika SET Level		
20. Verlag -	21. Preis -	

## Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication)	
3. title Final report		
4. author(s) (family name, first name(s)) Daniel Becker	5. end of project 31.10.2022	
	6. publication date 30.4.2023	
	7. form of publication Document	
8. performing organization(s) (name, address) Rheinisch-Westfälische Technische Hochschule Aachen Fakultät 4 – Maschinenwesen Institut für Kraftfahrzeuge Steinbachstr. 7 52074 Aachen	9. originator's report no.	
	10. reference no. 19A19004C	
	11. no. of pages 26	
12. sponsoring agency (name, address)  Bundesministerium für Wirtschaft und Klimaschutz (BMWK) 53107 Bonn	13. no. of references 1	
	14. no. of tables 1	
	15. no. of figures 21	
16. supplementary notes		
17. presented at (title, place, date) -		
18. abstract Final report of the work carried out at ika in the SET Level funding project. The work on subprojects 2 and 3 is summarized and documented. The following aspects await the reader: - Conception of a simulation architecture for scenario-based simulative testing. - Agent model for ambient traffic in the simulation - Road network model for efficient generation of simulation maps		
19. keywords Final report ika SET Level		
20. publisher --	21. price -	